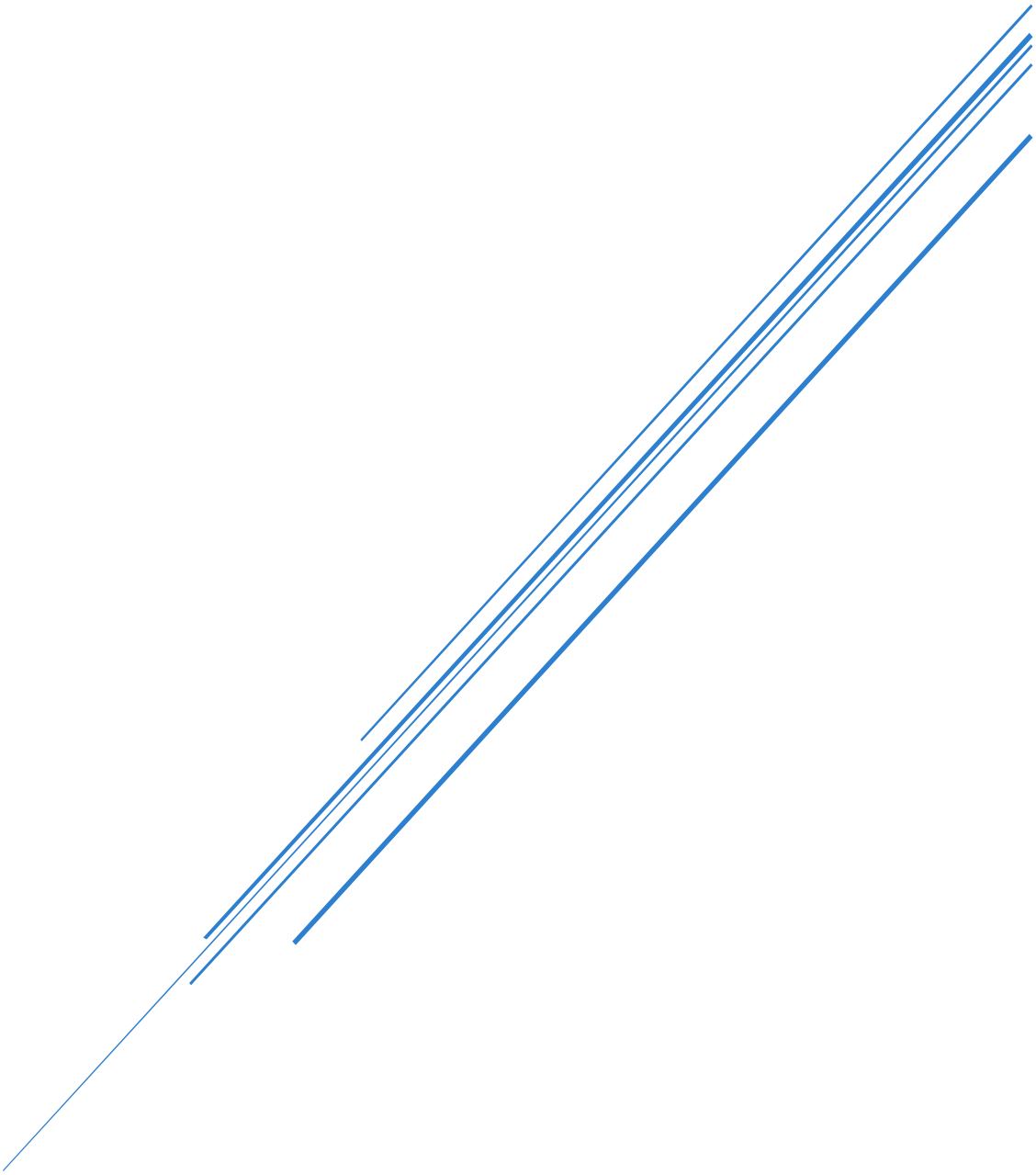
VERSATILE 3D WORLD RECONSTRUCTION USING A DIFFUSION MODEL

Sumer Kaistha - 3035859448



FYP Final Report

Date of Submission: 21st April, 2025

Acknowledgements

I would like to express my sincere gratitude to Professor Victor O.K. Li and Professor Jacqueline C.K. Lam for their invaluable guidance and advice throughout this research. Their insights have greatly helped in developing novel ideas and perspectives which have defined the direction and enhanced the quality of my work significantly.

Abstract

3D reconstruction from 2D images is a critical task across numerous fields, including robotics, gaming, augmented reality, and architecture. However, current methods for 3D reconstruction from sparse 2D views often struggle with issues such as missing details, loss of texture, and low accuracy. This research aims to address these challenges and enhance the 3D reconstruction process as well as enable the use of domain-specific expertise, making it applicable to diverse fields. A novel model architecture that integrates diffusion models and LLMs with Gaussian Splatting was proposed after an extensive literature review. The model will perform 3D reconstruction on an initial set of images using Gaussian Splatting and refine the Gaussian parameters through a pretrained LDM guided by conditioning which include a multimodal CLIP embedding of text and image priors provided by the user as well as utilizing LLM to generate prompts or images to give additional semantic information. The outcome is the development of a robust 3D reconstruction model capable of producing high-fidelity structures with sparse input data while leveraging domain-specific expertise. This work contributes to advancing 3D reconstruction methods, with potential applications in visual technologies and interactive fields. Future research could focus on training and utilizing the model in specialized fields to outline the importance of the user input as conditioning. It could also focus on utilizing more computational resources to further train the system and get more detailed evaluations and comparisons for the system.

Table of Contents

Contents

List of Figures

List of Tables

1. Introduction

Three-dimensional (3D) world reconstruction has emerged as a cornerstone technology for applications demanding precise spatial understanding and environmental interaction, from autonomous robotics to augmented and virtual reality (AR/VR) system. The ability to generate accurate, high-fidelity 3D models or scenes is critical for applications that require spatial awareness or interaction with the environment. For example, AR/VR applications require detailed 3D models of the real-world and virtual environments to create an immersive experience for the user and simulations require well defined models of objects and surroundings. However, reconstructing such models from sparse or occluded 2D imagery remains a formidable challenge in computer vision. Occlusions in 2D views often obscure critical structural details, and limited camera perspectives result in incomplete geometric inference—resulting in abundant 2D visual data but a lower amount of reliable and complete 3D reconstructions. Traditional methods struggle to resolve ambiguities in texture and depth for overlapping objects, necessitating novel approaches that bridge the gap between incomplete 2D inputs and robust 3D outputs. Recent advances in neural rendering and generative priors have reinvigorated this pursuit, yet achieving efficiency, scalability, and fidelity across diverse real-world scenarios remains an open frontier.

1.1 Literature Review and Project Motivation

Most 3D reconstruction systems rely on well-established techniques like stereo vision, where two images from slightly different viewpoints are used to infer depth, or structure-from-motion (SfM), which reconstructs 3D structure from the motion observed in 2D image sequences [1] and producing a point cloud based 3-D model similar to LiDAR. Another method, Simultaneous Localization and Mapping (SLAM), allows a camera system to update the map of an unknown environment while keeping track of the camera system inside the environment [2]. The traditional incremental SFM algorithm is slow, whereas the traditional global SFM algorithm prioritizes speed, compromising the accuracy [3]. A hybrid incremental and global SFM model offers an effective way to combine the advantages of both SFM models [3]. Alternatively, SLAM can significantly reduce computation time without compromising construction quality [4].

Despite the effectiveness of SLAM, it has limitations such as ORB-SLAM3 failing in low texture environments [5], or the issue of scale ambiguity in Monocular SLAM methods [6].

Traditional methods like stereo vision and SfM have been widely used but face challenges in handling sparse data, missing details, and achieving high fidelity. These limitations highlight the need for innovative solutions such as deep learning models and learning-based approaches. These methods offer a novel approach to generating high-quality 3D reconstructions from 2D data, improving efficiency and accessibility. Recent advancements in learning-based techniques like Neural Radiance Fields (NeRF) [7] and Gaussian Splatting [8] have transformed 3D reconstruction methods. NeRF encodes volumetric scene representations, allowing novel view synthesis by optimizing a continuous function of the 3D scene geometry and appearance [7]. PixelNeRF extends NeRF's capabilities by learning a scene-specific prior directly from image pixels, making it particularly effective in settings with limited views [9]. On the other hand, Gaussian Splatting represents scenes with Gaussian primitives instead of regular spare point clouds, offering faster rendering while maintaining high visual fidelity [8]. It is a rasterization-based 3D scene representation method that uses thousands of 3D Gaussians (ellipsoids) to model scenes. Although PixelNeRF performs better with sparse data compared to traditional methods, it struggles with highly sparse data or noisy data, leading to blurry images or missing details in the 3D structures. Gaussian Splatting's performance degrades when dealing with fine-grained details or complex textures, reducing the overall quality of reconstruction. Thus, diffusion models can be integrated to further refine and improve these methods.

In recent years, diffusion models have become powerful tools for generating high-quality images, outperforming traditional methods in some areas. Diffusion models are generative models used to iteratively refine noise into structured data. Diffusion models have demonstrated exceptional performance in 2D image generation and completion tasks and are now being explored for their potential in 3D modelling. DDPMs formalize image synthesis as a gradual denoising process guided by iterative refinement [10], and LDMs address DDPMs' computational inefficiency by operating in a compressed latent space [11]. Diffusion models, in frameworks such as ReconFusion, offer significant promise for overcoming challenges in image reconstruction from limited viewpoints, showing excellent performance even with minimal data [12]. Similarly, SDFusion, developed for 3D shape reconstruction, leverages multimodal data and an encoder-decoder to learn a diffusion model to enhance shape completion tasks [13]. Diffusion has also shown great potential to be integrated with gaussian splatting as GSGEN, a text-to-3D framework leveraging Gaussian Splatting and a two-stage optimization strategy (geometry refinement + appearance densification) [14], and ZERO-1-

to-G, a direct 3D generation method that decomposes Gaussian splats into multi-view attribute images, enabling pretrained 2D diffusion models to synthesize 3D structures via cross-view attention layers [15]. Thus, this project looks at diffusion models over other deep learning methods.

CLIP [16] is vital Recent work has demonstrated the potential of large language models (LLMs) to augment vision-language frameworks like CLIP by refining its textual conditioning. For example, LaCLIP [17] leverage LLMs to automate the generation of semantically diverse and domain-specific prompts, addressing CLIP's reliance on handcrafted, generic descriptors. DiffusionGPT [18] and LMD [19] are shown to be able to create images following complex text prompts

1.2 Project Objectives

This project seeks to build on the previous work with diffusion models to develop a diffusion-based framework for 3D world reconstruction from static images. The primary aim is to provide a flexible and robust method for reconstructing 3D environments that is applicable to various domains, including navigation, robotics, and AR/VR. This aim can be achieved by adding user input with domain-specific knowledge into the diffusion conditioning process which will guide the diffusion process and offer better visual fidelity. A LLM will also be used to assist the user in providing input to condition the diffusion process. There are multiple objectives that the project aims to achieve:

- It will enhance accuracy and detail in 3D object reconstruction and shape completion from a sparse set of input images, where obtaining a large number of images with varying camera perspectives and multiple views of an object or environment may not be feasible. The system will be able to reconstruct detailed 3D scenes from limited 2D input data, such as single images or video frames by inferring occluded sections using a diffusion model.

- The system will be versatile enough to be used in various applications, including medicine, robotics, AR/VR, and environmental monitoring.

- The system will incorporate the ability to leverage input from domain experts, allowing them to provide valuable insights and feedback that can significantly enhance the reconstruction process. By integrating expert knowledge, the system will

be able to refine and optimize 3D scene generation, ensuring that the results align more closely with real-world expectations and complexities.

The paper is structured as follows. To begin with, Section 2 details the project methodology providing an overview of the datasets and data collection process, preprocessing techniques, and the diffusion model architecture and workflow. Section 3 covers the results of the project as well as discusses limitations and challenges faced. Finally, Section 4 provides the concluding remarks for this project report.

2. Project Methodology

The proposed methodology integrates ideas and techniques from a multitude of recent papers which use Diffusion Models with NeRF or Gaussian Splatting, such as ReconFusion, SDFusion, and GSGEN [12-15] and expands on them, to achieve the objectives outlined in the section before. The first part includes the data collection and preprocessing, followed by the second part which includes the model architecture and implementation.

2.1 Data Collection and Preprocessing

This subsection provides an overview of the data collection process and covers the public datasets that will be used. It also details a variety of preprocessing techniques employed to prepare the data for the model.

Pretrained models are used wherever possible to lower the computational resources and time needed to train the model. Thus, the data collection is strictly for finetuning the models. The diffusion model, VAE, and text and image encoders are all pretrained. Stable Diffusion 2.1 is used for the diffusion model along with the VAE trained for Stable Diffusion and a CLIP model is used for the text and image embedding for conditioning. Thus, the collected data from the datasets will be used in finetuning a text-to-image diffusion model which utilizes CLIP embeddings to allow for the 3D reconstruction task.

For the image-based 3D reconstruction process, this research will utilize publicly available datasets such as ShapeNet and Tanks and Temples. These datasets are specifically selected for their rich variety of 2D images or 3D models which can be used to generate synthetic multi-view 2D images, providing a comprehensive resource for training and evaluating the model. ShapeNet contains a vast variety of object categories with detailed 3D models [20].

To ensure the quality of the input data, several preprocessing techniques will be applied. The images will be normalized to improve the contrast and clarity of the images and image and standardized to ensure uniformity and make the training process more efficient. Furthermore, in addition to these standard preprocessing techniques, data augmentation will be utilized to further enhance the model's ability to generalize across various scenarios. Augmentation strategies will be applied only to the images used with the diffusion model to simulate a broader range of viewing conditions and input variations. Fine-tuning Stable Diffusion aligns the model’s priors with the target domain (ShapeNet), improving its ability to guide 3D reconstruction with SDS. The pipeline would benefit greatly when fine-tuning on highly

domain-specific images (medical scans, environmental monitoring imagery) as it aligns CLIP embeddings to the target domain and allows the Diffusion Model to associate domain-specific textures and structures. Thus, it is highly recommended to fine-tune the model on domain-specific images before using it for the specialized use case.

2.2 Model Architecture and Pipeline

Both NeRF and Gaussian Splatting were considered as options to create the 3D models before the Diffusion Model refines it. Eventually, Gaussian Splatting was chosen due to faster rendering and less computational complexity. The Gaussian Splatting was used as a base to render the 3D models, and the diffusion model was integrated to refine the Gaussian Splats and allow for multimodal conditioning from user inputs. The user can use multimodal input such as text prompts or images to condition and guide the diffusion process. This works by passing the 2D ground-truth images along with prompts and additional images into the CLIP model to get embeddings with semantic information used to refine the Gaussian parameters and therefore use their domain-specific expertise to assist the creation of the 3D model and renders. An integrated LLM module can be optionally used by the user to help in generating the conditioning prompts and images. The Gaussian Splatting reconstruction loss was integrated with Score Distillation Sampling (SDS) loss from the diffusion model to refine the Gaussian parameters according to the semantic information in the CLIP embeddings. SDS leverages the pretrained Diffusion Model's score function to guide the Gaussian Splatting without backpropagating through the Diffusion U-Net [21]. At each optimization step, noise is sampled and injected into the rendered 2D image, and then the diffusion model is queried to predict the noise and remove it. The difference between the image after the diffusion model predicts noise to denoise the image and the original render gives the SDS loss which provides pixel-wise gradient signals, guiding appearance of the renders towards what the diffusion model considers realistic. SDS integrates well with the pipeline as it is efficient (no backpropagation through frozen U-Net yet uses diffusion priors) and has low variance as compared to other loss functions such as Standard Denoising MSE loss. The pretrained Diffusion U-Net was completely frozen after finetuning, and the loss was used only to improve the gaussian splatting process. The full pipeline can be seen in Figure 1 below.

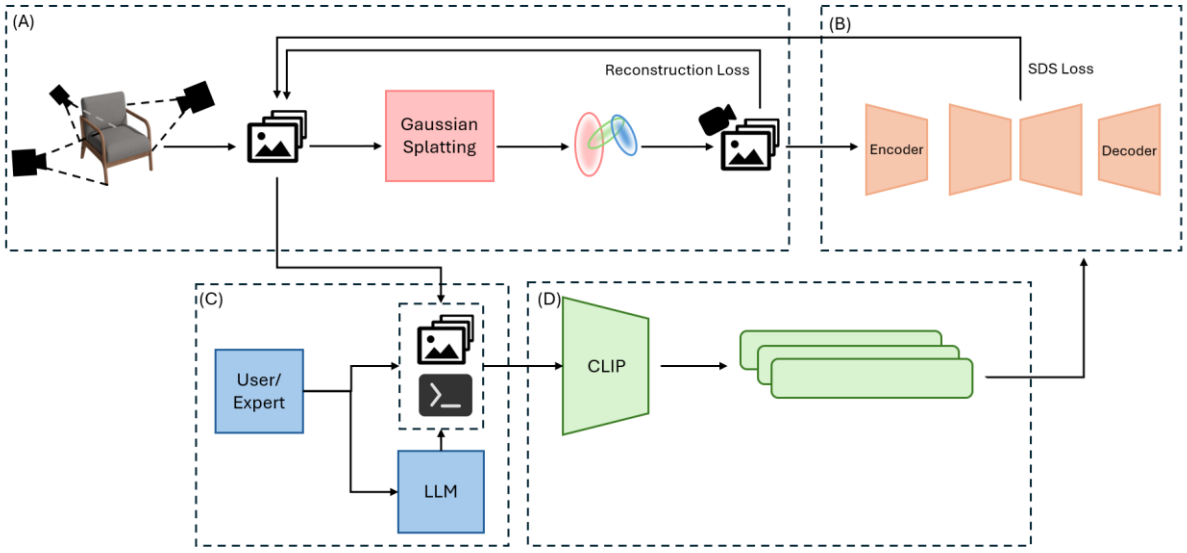


Figure 1: The model architecture is broken down into 4 modules: (A) The Gaussian Splatting module that creates the 3D Gaussians and minimizes the L1 and D-SSIM loss between the rendered views and the ground truth views. (B) The Stable Diffusion LDM will take the input rendered views and conditioning information to calculate SDS loss to refine and guide the Gaussian parameter optimization. (C) The User input and LLM module that will be passed into CLIP. This consists of the original ground truth views, any user prompt/LLM-generated prompt, and additional images/LLM-generated images. (D) The conditioning module uses the input images, user and LLM text prompts, and user provided or LLM-generated images to create CLIP embeddings which will be used in the conditioning for the Stable Diffusion U-Net.

This architecture and pipeline were chosen as SDS along with Gaussian Splatting and a pretrained Diffusion Model such as Stable Diffusion unifies the 2D diffusion priors and explicit 3D representation. It delivers fast guidance without needing to train or backpropagate through a heavy U-Net and enables the integration of a Diffusion Model and its priors without massive data and computation burdens of training models from scratch which demands large 3D datasets, extensive GPU resources, and can suffer from overfitting as a custom Latent Diffusion Model is only as rich as its training data. In contrast, fine-tuning the Stable Diffusion pipeline suffices to bridge highly specific domain gaps and leveraging pretrained models provides robust generalization and rich semantics which can be hard to match from ground-up training.

2.2.1 Gaussian Splatting

The first component is 3D Gaussian Splatting (3DGS), which is the backbone of the entire process used to generate the 3D gaussian primitives which create the 3D object or scene and can be rendered from various camera angles. Each 3D Gaussian is parametrized by position (XYZ coordinates), covariance parameters (scale and rotation), opacity (α) and colour which

is represented using RGB values or spherical harmonics coefficients for view-dependent effects [8]. It starts with using SfM to create a sparse point cloud and then processes the points into isotropic gaussian spheres or "splats". Figure 2 shows the Gaussian Splatting pipeline.

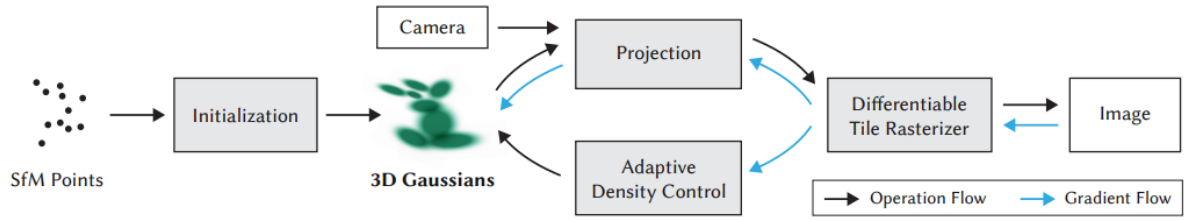


Figure 2: The Gaussian Splatting pipeline, adapted from [8]. The process begins with using SfM to create a point cloud followed by transforming the point cloud into Gaussians and then optimizing the Gaussian parameters by taking the L1 and D-SSIM loss between the projection (rendered image) and the ground truth image as well as adaptive density control.

As can be seen in the figure, each splat's parameters are trained and optimized through Stochastic Gradient Descent (SGD) by minimizing the L1 (Mean Absolute Error) and D-SSIM loss between the rendered scene from the gaussians with the same viewpoint as the ground-truth views and the original 2D image ground-truth views. It goes through adaptive densification which adds gaussians in under-reconstructed regions or removes gaussians that are essentially transparent (where the opacity α is less than a threshold) during the optimization and the rendering is done through projection of the 3D Gaussians into 2D using camera matrices and using a differentiable rasterizer for gaussians to sort the gaussians and blend them tile-by-tile (16x16 pixels). [8].

Gaussian Splatting is ideal to integrate with the pretrained Stable Diffusion model as the rasterization is fully differentiable which allows the gradients from SDS loss to flow back to the Gaussian parameters. Furthermore, each Gaussians is an explicit geometric primitive with trainable parameters, unlike NeRF's implicit radiance fields. This allows for the manipulation of individual Gaussians during the optimization process which aligns well with the Diffusion Model guided by the CLIP embeddings as multimodal conditioning. The gaussians can adjust colours, positions, or other parameters based on the CLIP embeddings for semantic alignment with the user's input. Finally, Gaussian Splatting renders at much higher speeds (60+ FPS) and is more memory efficient compared to NeRF, which can render at <1 FPS, which makes it more practical for iterative SDS optimization.

2.2.2 Diffusion

The second component is the diffusion model which is the core of the project. Stable Diffusion 2.1, a pretrained latent diffusion model (LDM), is implemented to refine the 2D renders of the gaussian splats by denoising according to the conditioning information provided. As seen in Figure 3, the diffusion process consists of two processes – a forward and a reverse process. The forward process adds noise into the image over several timesteps moving from X_0 to X_T. The transition between each timestep, from X_{t-1} to X_t is modelled as a Gaussian distribution shown by $q(X_t \mid X_{t-1})$. The reverse process learns to reverse the noise, starting from X_T to X_0. This is done through learning a parameterized distribution p_θ $(X_{t-1} \mid X_t)$. [10]

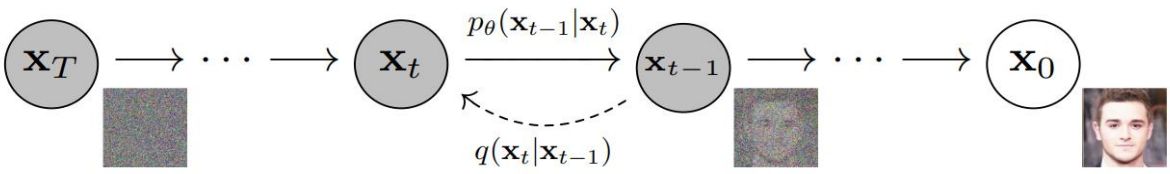


Figure 3: The components of the Diffusion Probabilistic Model (DDPM). The forward process uses a Gaussian distribution to add noise to the image. The reverse process learns parameters to denoise the image through a parameterized distribution. The figure is adapted from [10].

LDMs push the diffusion process into a compact latent space which is learnt by a Variational Autoencoder (VAE) rather than operating on the high dimensions of the image pixel space. As seen in Figure 4, The image is encoded from the pixel space to a lower dimension latent space where the diffusion process occurs and eventually the output is decoded back into the pixel space. This allows for improvements in efficiency, memory, and sampling speed while preserving image quality. LDMs support classifier-free guidance through cross-attention conditioning. LDMs embed conditioning (such as images, text, style or any other semantic information) via cross-attention layers in the latent U-Net allowing for rich multimodal control without extra classifiers [11]. For the proposed pipeline, an LDM like Stable Diffusion offers seamless integration of conditioning information and fast, scalable training making it more practical than a pixel-space DDPM.

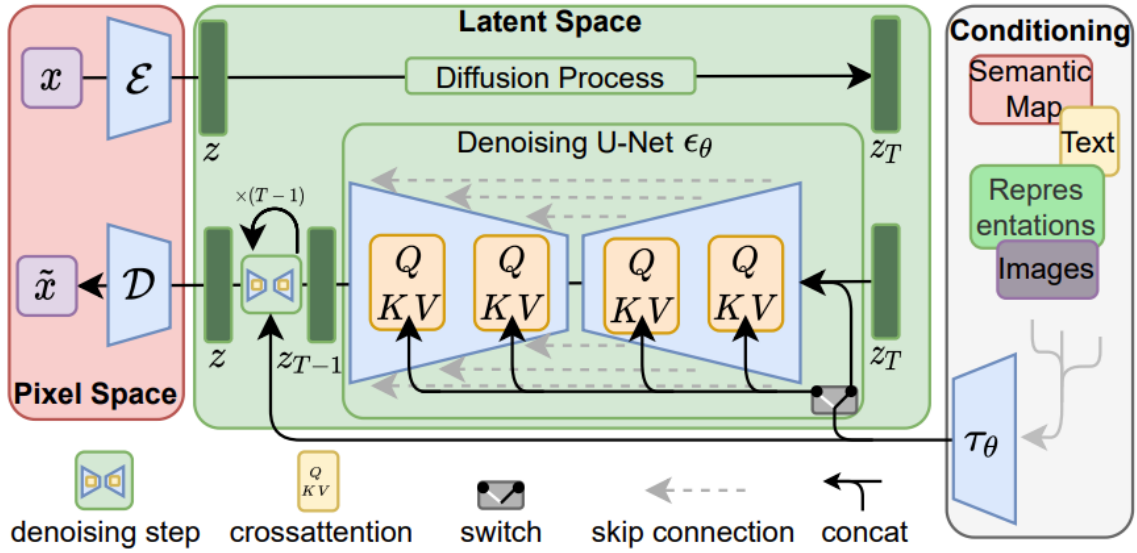


Figure 4: The Latent Diffusion Model Pipeline, adapted from [11]. The image is encoded from the pixel space into a latent space before the diffusion process occurs with cross-attention conditioning. The output is decoded back into the pixel space.

As Gaussian Splatting may leave holes or ambiguous surfaces in unobserved regions, LDMs powerful, pretrained semantic priors can fill these gaps by creating realistic content consistent with the other 2D renders and the conditioning provided. The LDM model will act as a denoising tool, correcting textures and generating plausible surfaces according to the conditioning where data is incomplete or ambiguous. To guide this process, conditioning is vital, and thus both text and image conditioning can be used to guide the diffusion process. Stable Diffusion 2.1 [22] was selected as it is uniquely well-suited to a Gaussian Splatting pipeline as its open licensing, robust documentation and tools, efficient latent architecture, and high-quality priors stand out among other pretrained models. Version 2.1's refinements in face realism, text fidelity, fine detailing, and depth-to-image module make it ideal for adding novel-view priors or filling residual holes in Gaussian splats [23].

2.2.3 Contrastive Language-Image Pre-Training Integration

The third component is the pretrained Contrastive Language-Image Pre-Training (CLIP) model which is used to generate the text and image embeddings to use as conditioning for the diffusion process. CLIP is a neural network that connects images and text which is trained to ensure that corresponding image-text pairs are close together in the latent space and resulting embeddings [16]. It learns a shared embedding space by jointly training an image encoder and text encoder with contrastive objects on 400 million image-text pairs [16]. During training, matching image-text pairs pulled together and mismatched pairs are pushed apart.

This makes CLIP embeddings ideal to use as multimodal (text and image) conditioning for the Diffusion Model. Stable Diffusion conditions its U-Net on CLIP text embeddings and thus its native integration with Stable Diffusion makes it the primary choice for conditioning the Gaussian splatting with Stable Diffusion pipeline. The embeddings are obtained from the 2D image ground-truth views as well as user input text prompts or additional images to provide richer semantic information. The text prompts and images go through the CLIP model to generate the CLIP embeddings. Experts will be able to input specific prompts and images and create CLIP embeddings which will be added to the conditioning.

2.2.4 Large Language Model Integration

Furthermore, an optional component was implemented which leverages a Large Language Model (LLM) and its vast and diverse training data to help create the conditioning embeddings. Leveraging its embedded knowledge of language, context, and real-world concepts to generate detailed and diverse text prompts or synthetic images enriches the CLIP conditioning in the pipeline by embedding richer semantic information. LLM-driven prompt augmentation such as in LaCLIP [17] and CuPL [24] have been shown to significantly boost CLIP's zero-shot classification accuracy by creating detailed and varied descriptions. The LLM will allow users to provide insights through detailed LLM created text prompts and LLM generated images which should add additional semantic context to the CLIP embeddings used for conditioning and enable the use of domain experts' insights in the refinement process. By integrating the LLM in the conditioning, the diffusion process will be able to better reconstruct and refine the gaussian splats and subsequently, the rendered 3D structure by better inferring key features and textures.

2.3 Implementation Details

The original Gaussian Splatting Repository [8] was cloned and then configured to integrate with the Stable Diffusion Model and CLIP model to obtain embeddings for conditioning the diffusion process. SDS Loss was weighted with a default weight set to 0.2, balancing its influence against gaussian splatting geometric reconstruction. This value was chosen based on previous literature works like GSGEN [14]. The weight is adjustable via a command-line argument by using –lambda_sds (e.g., --lambda_sds 0.3) to prioritize flexibility in the influence given to the user input conditioning and diffusion process. To address computational constraints, a Docker container was built with nvidia/cuda:11.8.0-devel-ubuntu22.04 which provides CUDA SDK 11.8 as it is required by the gaussian splatting

repository. Stable Diffusion and CLIP were integrated through HuggingFace diffusers and transformers libraries. The corresponding Docker image was uploaded to AWS EC2 g6 instance with Nvidia L4 Tensor Core GPUs (24GB VRAM). The model was trained on 7000 iterations due to the computational and time complexity to train for larger number of iterations. The Stable Diffusion Model was fine-tuned on ShapeNet data. Once the Diffusion model was frozen, the pipeline was trained on Tanks and Temples Dataset and Deep Blending like the original Gaussian Splatting to make comparisons.

2.4 Evaluation Metrics

The system will be evaluated using three metrics to ensure high visual fidelity in the sampled novel views and ensure precision. Peak Signal-to-Noise Ratio (PSNR) will be used to measure the pixel-wise differences for assessing reconstruction quality, Structural Similarity Index Measure (SSIM) will be used to evaluate structural similarity for perceptual accuracy, and Learned Perceptual Image Patch Similarity (LPIPS) will be used to quantify perceptual differences, ensuring high-fidelity outputs. The model will also be evaluated by comparing it to the original gaussian splatting model as a baseline.

3 Discussion

3.1 Results

The results can be visualized through the Interactive SIBR viewer from the official Gaussian Splatting repository, enabling dynamic exploration of the 3D scenes from arbitrary viewpoints. The figure below (Figure 5) showcases one such rendered output of the system for visualization purposes.

Figure 5: Top: Shows the rendered outputs of the system with the gaussian splats set to 1 (full size). Bottom: shows the rendered outputs of the system with the gaussian splats set to minimum size to show the point cloud.

The system was evaluated using the same datasets employed by Gaussian Splatting, including Tanks and Temples and Deep Blending, ensuring comparability with the Gaussian Splattering evaluation. Key image quality metrics (SSIM, PSNR, and LPIPS) were computed to evaluate reconstruction fidelity and assess performance.

Model	SSIM	PSNR	LPIPS
Gaussian Splatting-30k	**0.83**	**26.91**	0.213
Gaussian Splatting-7k	0.78	25.2	0.284
Ours-7k	0.80	26.1	**0.163**

Table 1: Comparison of key metrics between the outlined system (Diffusion + Gaussian Splatting) and Gaussian Splatting.

At 7000 iterations, the model achieved an average SSIM of 0.80 (higher is better), surpassing the Gaussian Splatting baseline of 0.78 at 7000 iterations. This indicates that the system produces structurally more consistent and perceptually coherent reconstructions, even with fewer optimization steps. Notably, while Gaussian Splatting achieves a marginally higher SSIM of 0.83 at 30,000 iterations, the proposed method's accelerated convergence suggests comparable quality could be attained with significantly reduced computational effort.

The proposed system attained an average PSNR of 26.1 (higher is better) at 7000 iterations, compared to 25.2 for Gaussian Splatting at the same point. This improvement underscores the system's superior ability to suppress noise and produce high-fidelity image outputs from limited input views. While Gaussian Splatting once again reaches 26.91 at 30000 iterations, the proposed pipeline's superior early-stage performance could suggest further improvement if the model is extended to a larger number of iterations.

The higher SSIM/PSNR at 7k iterations suggests the SDS loss leverages 2D diffusion priors to fill in missing geometric details from sparse inputs, addressing shape completion challenges.

The proposed system achieves an average LPIPS score of 0.163 (lower is better), significantly outperforming Gaussian Splatting's 0.284 at 7k iterations and even surpassing its 0.213 at 30k iterations. This demonstrates superior perceptual alignment with ground-truth imagery, attributable to Stable Diffusion's priors enhancing fine details such as textures, edge sharpness, and subtle lighting variations. While traditional metrics like SSIM and PSNR focus on structural or pixel-level accuracy, LPIPS captures semantic fidelity which is critical for human-centric applications.

Overall, the results illustrate that the integrated Diffusion and Gaussian Splatting pipeline outperforms Gaussian Splatting at early optimization stages, both in terms of perceptual quality and reconstruction fidelity. The combination of semantic diffusion priors and image inputs leads to more data-efficient reconstructions and better shape and detail inference at 7000 iterations. The higher SSIM/PSNR at 7k iterations validates the pipeline's ability to reconstruct accurate 3D scenes from limited views. The Diffusion Model acts as a "detail amplifier," using 2D diffusion priors to infer missing regions. Training on diverse datasets ensures robustness to varied scenes, from indoor objects to outdoor environments. The Diffusion Model also allows for cross-attention conditioning to be used along with CLIP models which enables experts to input text prompts and images to help guide the process and infer missing or sparse regions.

3.2 Ablation Study

To understand the importance of the user input as conditioning and the LLM support, an ablation study was conducted where no prompt, only user prompt, and LLM assisted prompt were considered.

Conditioning	SSIM	PSNR	LPIPS
No Prompts	0.79	25.8	0.173
User Prompts	0.79	25.1	0.172
LLM-Generated Prompts	**0.80**	**26.1**	**0.163**

Table 2: Ablation study comparing the importance of user prompts and LLM-generated prompts as conditioning input.

LLMs' ability to generate rich, context-aware descriptions provides more granular guidance for SDS loss, improving the metrics slightly. In generic datasets, simple prompts from the

user lack detail to meaningfully guide reconstruction, resulting in performance similar to no prompts. However, the user prompts and LLM prompts as conditioning should be more valuable and give more improvements when working with data from highly specialized domains (e.g., "CT scan of a femur with osteoporotic trabeculae" for a medical image). The lower PSNR score with the user prompt as compared to no prompt could be due to the user prompt being misaligned with the reconstruction of the image. It is important to use semantically important and aligned prompts to improve performance.

3.3 Challenges

The development process encountered significant technical hurdles, particularly in environment setup and computational resource limitations. Initial attempts to build the pipeline resulted in dependency conflicts and build issues. To address this, a Docker container was configured with Ubuntu 22.04, encapsulating all dependencies to ensure reproducibility and isolate the environment from host-system inconsistencies. Computational complexity of the model training resulted in another challenge. The Diffusion-guided Gaussian Splatting Model required more VRAM than I had access to (6GB VRAM was exceeded) and long runtimes due to my lacking GPU resources. To avoid too much time and resources spent training, pretrained models were implemented where possible and the training was performed on an AWS EC2 GPU instance.

3.4 Limitations

The use of a pretrained text-to-image LDM on the 2D renders could potentially limit fine structural accuracy as the geometry is informed by 2D semantics. This could be overcome by using large-scale labelled 3D datasets and training an LDM from scratch that directly trains on the Gaussians at the cost of using higher computational resources. Gaussian Splatting renders quickly but creates thousands of gaussians which may make it hard to interact with objects rendered by the splats directly. Using NeRF as an alternative with diffusion or implementing hybrid renderers that fuse splats with mesh proxies could help improve interactivity. Finally, there may be limited quality under extreme novel viewpoints even with diffusion enhancements resulting in rendering artifacts. This could be overcome by fine-tuning VAE decoder or the U-Net using LoRA to further boost performance under sparse views.

3.5 Future Work

Building on this project, there are several directions to advance the capabilities of diffusion-guided 3D reconstruction pipelines. First, training the system with highly specific domain data, such as medical imaging, could better frame the potential of the LLM-generated prompts and domain expert feedback in the conditioning process which will be left for future work. Furthermore, extending the training to 30000 iterations paired with more detailed and granular evaluation and ablation studies could better showcase the impact of this framework and comparisons against state-of-the-art models could help in benchmarking the performance. Finally, curating multi-view 3D datasets with text annotations would enable supervised fine-tuning of diffusion models directly on 3D-consistent data, reducing reliance on 2D distillation. Adapting Latent Diffusion Models (LDMs) to operate natively on 3D gaussians could further bridge the 2D-3D domain gap. Together, these steps would solidify the pipeline's robustness in data-scarce scenarios while unlocking new applications in precision medicine and embodied AI.

4. Conclusion

This research presents a novel framework that synergizes diffusion models, Gaussian splatting, and large language models (LLMs) to address the critical challenge of high-fidelity 3D scene reconstruction from sparse 2D inputs. By integrating Stable Diffusion's generative priors via Score Distillation Sampling (SDS) loss with Gaussian splatting's explicit geometric representation, the pipeline achieves enhanced visual fidelity, structural accuracy, and fine-grained detail preservation compared to standalone Gaussian splatting, as evidenced by superior SSIM, PSNR, and LPIPS scores at 7,000 iterations. The inclusion of LLM-generated conditioning further refines reconstructions through semantic alignment with domain-specific knowledge, enabling tailored outputs for applications ranging from AR/VR environments to medical imaging.

The significance of this work lies in its scalable integration of Gaussian splatting for efficient 3D rendering, pretrained diffusion models for detail synthesis and guiding the optimization of the gaussian parameters, and LLMs for semantic guidance—which collectively reduce reliance on dense multi-view data while maintaining computational efficiency. Future work will focus on validating the pipeline in specialized domains to quantify LLM-expert collaboration, extending training to 30,000 iterations to exploring further evaluations, and curating multi-view 3D datasets with text annotations to train diffusion models directly on 3D-consistent data, bridging the 2D-3D domain gap. These advancements aim to solidify the framework's role as a versatile, user-guided solution for next-generation 3D reconstruction, with transformative potential across robotics, telemedicine, and immersive technologies.

References

[1] M. Kholil, I. Ismanto, and M. N. Fu'ad, "3D reconstruction using Structure From Motion (SFM) algorithm and Multi View Stereo (MVS) based on computer vision," *IOP Conference Series: Materials Science and Engineering*, vol. 1073, no. 1, p. 012066, Feb. 2021, doi: https://doi.org/10.1088/1757-899x/1073/1/012066.

[2] A. Macario Barros, M. Michel, Y. Moline, G. Corre, and F. Carrel, "A Comprehensive Survey of Visual SLAM Algorithms," *Robotics*, vol. 11, no. 1, p. 24, Feb. 2022, doi: https://doi.org/10.3390/robotics11010024.

[3] L. Gao, Y. Zhao, J. Han, and H. Liu, "Research on Multi-View 3D Reconstruction Technology Based on SFM," *Sensors*, vol. 22, no. 12, pp. 4366–4366, Jun. 2022, doi: https://doi.org/10.3390/s22124366.

[4] B. Yang, S. Rosa, A. Markham, N. Trigoni, and H. Wen, "Dense 3D Object Reconstruction from a Single Depth View," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 12, pp. 2820–2834, doi: https://doi.org/10.1109/TPAMI.2018.2868195.

[5] C. Campos, R. Elvira, J. J. G. Rodriguez, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1–17, 2021, doi: https://doi.org/10.1109/tro.2021.3075644.

[6] T. Taketomi, H. Uchiyama, and S. Ikeda, "Visual SLAM algorithms: a survey from 2010 to 2016," *IPSJ Transactions on Computer Vision and Applications*, vol. 9, no. 1, p. 16, 2017, doi: https://doi.org/10.1186/s4107401700272.

[7] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis," *arXiv.org*, 2020. https://arxiv.org/abs/2003.08934v2

[8] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3D Gaussian Splatting for Real-Time Radiance Field Rendering," *arXiv.org*, 2023. https://arxiv.org/abs/2308.04079v1

[9] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, "pixelNeRF: Neural Radiance Fields from One or Few Images," *arXiv.org*, 2020. https://arxiv.org/abs/2012.02190v3

[10] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, Eds., Curran Associates, Inc., 2020, pp. 6840–6851.

[11] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-Resolution Image Synthesis with Latent Diffusion Models,” Apr. 2022. Available: https://arxiv.org/pdf/2112.10752

[12] R. Wu et al., “ReconFusion: 3D Reconstruction with Diffusion Priors,” arXiv.org, 2023. https://arxiv.org/abs/2312.02981v1

[13] Y. C. Cheng, H. Y. Lee, S. Tulyakov, A. Schwing, and L. Gui, “SDFusion: Multimodal 3D Shape Completion, Reconstruction, and Generation,” in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4456–4465. doi: https://doi.org/10.1109/CVPR52729.2023.00433.

[14] Z. Chen, F. Wang, and H. Liu, “Text-to-3D using Gaussian Splatting,” arXiv.org, Oct. 30, 2023. https://arxiv.org/abs/2309.16585

[15] X. Meng, C. Wang, J. Lei, K. Daniilidis, J. Gu, and L. Liu, “Zero-1-to-G: Taming Pretrained 2D Diffusion Model for Direct 3D Generation,” Arxiv.org, 2020. https://arxiv.org/html/2501.05427 (accessed Apr. 21, 2025).

[16] A. Radford et al., “Learning Transferable Visual Models from Natural Language Supervision,” arXiv.org, Feb. 26, 2021. http://arxiv.org/abs/2103.00020

[17] L. Fan, D. Krishnan, P. Isola, D. Katabi, and Y. Tian, “Improving CLIP Training with Language Rewrites,” arXiv.org, 2023. https://arxiv.org/abs/2305.20088 (accessed Apr. 21, 2025).

[18] J. Qin *et al.*, “DiffusionGPT: LLM-Driven Text-to-Image Generation System,” *arXiv.org*, Jan. 18, 2024. https://arxiv.org/abs/2401.10061

[19] L. Lian, B. Li, A. Yala, and T. Darrell, “LLM-grounded Diffusion: Enhancing Prompt Understanding of Text-to-Image Diffusion Models with Large Language Models,” *arXiv.org*, 2023. https://arxiv.org/abs/2305.13655v3

[20] “ShapeNet,” *shapenet.org*. https://shapenet.org/

https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf

[21] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, “DreamFusion: Text-to-3D using 2D Diffusion,” arXiv:2209.14988 [cs, stat], Sep. 2022, Available: https://arxiv.org/abs/2209.14988

[22] “stabilityai/stable-diffusion-2-1 · Hugging Face,” huggingface.co. https://huggingface.co/stabilityai/stable-diffusion-2-1

[23] J. Stokes, “Stable Diffusion 2.0 & 2.1: An Overview,” Jonstokes.com, Dec. 29, 2022. https://www.jonstokes.com/p/stable-diffusion-20-and-21-an-overview (accessed Apr. 21, 2025).

[24] A. Kravets, “A Simple Way of Improving Zero-Shot CLIP Performance,” Medium, Nov. 03, 2023. https://medium.com/data-science/simple-way-of-improving-zero-shot-clip-performance-4eae474cb447 (accessed Apr. 14, 2025).