ESE546 Final Project Report

Akash Sundar, Ojas Mandlecha and Nidhi Bali

***Abstract*— This project aims to enhance video calls with dynamic virtual backgrounds. Leveraging diffusion-based scene representation trained on 3D scene datasets, we create high-definition, textured environments from text input. Additionally, our system provides real-time adaptation to camera movements during calls, allowing users to adjust their camera angle or position, seamlessly mirroring these changes in the virtual environment. This endeavor redefines video calls by introducing an immersive, adaptive virtual space that faithfully replicates real-world dynamics.**

I. Introduction

Within the ever-transforming realm of remote communication technologies, our project represents a significant undertaking. Our primary aim is to enrich the virtual communication experience through the seamless integration of dynamic background replacement, leveraging the convergence of NeRFs (Neural Radiance Fields) and diffusion techniques for a virtual 3D environment generation.

To realize this goal, we harness the power of optical flow, an indispensable technique for precise camera tracking that adapts backgrounds in real-time to mirror camera movements. Simultaneously, segmentation techniques come into play, effectively isolating objects of focus while seamlessly transforming backgrounds. This dynamic amalgamation of NeRFs, diffusion, optical flow, and segmentation promises to engender immersive and adaptable communication platforms, profoundly enhancing the video call experience.

II. Related Work

In recent years, the field of text-based image generation has advanced significantly, with diffusion-based scene representation models like DreamFusion [1] and Text2Room [2] leading the way in transforming text into immersive visual landscapes. In particular, papers such as Text2Room, RoomDreamer [3] and Text2NeRF [4] have delved into the domain of text-generated 3D indoor scenes. As a more specialized sub-category, these deal predominantly with generating hyperrealistic indoor 3D scene representations.

The methodology to achieve this result can largely be condensed into 2 categories. RoomDreamer [3], by Apple, uses an approach wherein the model accepts an input of a 3D mesh as well as a text input. The model has been trained on their own dataset (ARKitScenes) to improve the style of the input 3D mesh to match more closely with the text input.

Text2NerF, Text2Room and MVDiffusion [5] all adopt a slightly different approach. They split the task into 2 sections. They first employ a text-to-image generation model such as StableDiffusion [6] to generate a view of the scene given the text prompt. They then build a pipeline to process this view(s) of the scene into a 3D NeRF sequence.

The authors of MVDiffusion implemented their own custom Stable Diffusion class and trained the model on Matterport3D and Scannet datasets to improve their performance on predicting indoor images while the latter two used a pretrained network and focused on making better 3D mesh predictions upon this.

MVDiffusion introduces ”correspondence-aware attention” (CAA) between UNet blocks that facilitates cross-view interactions, ensuring multi-view consistency during image synthesis. Trained separately, CAA blocks are integrated into the Stable Diffusion model. MVDiffusion excels in generating high-resolution, photorealistic 360-degree panoramic images from diverse texts, including outdoor and cartoon styles.

Text2NeRF estimates the depth of the generated view of the scene and utilizes a depth image-based rendering (DIBR) to construct a support set S0. A progressive scene inpainting and updating strategy, incorporating diffusion and depth estimation models to generate new views are used to iteratively complete missing regions. The NeRF is updated with each iteration using rendered images and aligned depths, enhancing scene synthesis based on the input text prompt.

This project uses image segmentation to create a background mask using DeepLab-V3 [7] trained on COCO train2017. DeepLab utilizes ”Atrous Convolution for Dense Feature Extraction” in semantic segmentation, employing cascaded ResNet blocks with atrous convolution for flexible output stride, enhanced long-range information capture, and improved semantic segmentation accuracy in DCNNs. This framework ensures precision in isolating subjects from the background objects. DeepLab-V3 introduces a dedicated mask branch to predict segmentation masks for each region of interest, by preserving exact spatial locations. Additionally, Optical flow estimation is an essential technique for tracking camera movement in real-time.

III. Method

We explored different approaches to the problem found in the literature. We adapted the method from [2] and followed up with elements from [5]. Our method involves using the Stable-Diffusion model to estimate the first view of a 3D scene based on given text prompts and then improving upon that image to construct an entire scene. Recognizing a potential problem with relying only on one starting image, we noticed challenges in creating a detailed depth map. Hence we simplified the objective to develop a detailed panoramic view of the scene. The method has been explained in greater detail below.

A. Diffusion Model

Stable Diffusion 2 [8] is a latent diffusion model conditioned on the penultimate text embeddings of a CLIP ViT-H/14 text encoder. It works by:

- Gradually adding noise to an image until it is unrecognizable
- Using a text prompt to guide the model in removing the noise and reconstructing the image
- Repeating this process until the model converges on an image that matches the text prompt
- Converting the latent representation space into a complete image

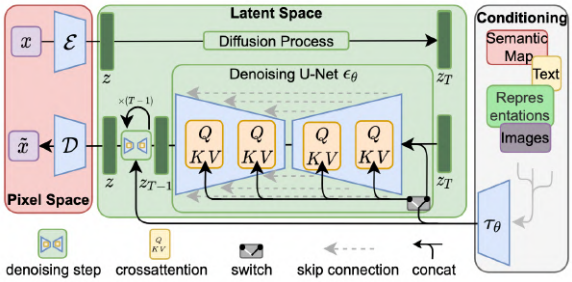


Fig. 1: Stable Diffusion Model Architecture

The main architectural components of Stable Diffusion include a variational autoencoder, forward and reverse diffusion, a noise predictor, and text conditioning.

Variational Autoencoder (VAE): Comprising an encoder and decoder, VAE compresses a 512x512 pixel image into a manipulable 64x64 latent space, later restoring it to a full-size 512x512 pixel image.

Forward Diffusion: Introduces Gaussian noise progressively until the image becomes unidentifiable. Used in training and image-to-image conversion.

Reverse Diffusion: A parameterized process that iteratively reverses forward diffusion, steering towards specified images. In practice, trained on vast image datasets, creating unique images through prompts.

Noise Predictor: Utilizes a Residual Neural Network (ResNet), for denoising. Estimates and subtracts noise from the latent space iteratively based on user-defined steps.

Text Conditioning: Uses text prompts as conditioning, employing a CLIP tokenizer to embed words into a 768-value vector. Up to 75 tokens can be used. These prompts guide the U-Net noise predictor via a text transformer for image generation in latent space.

B. NeRF Model

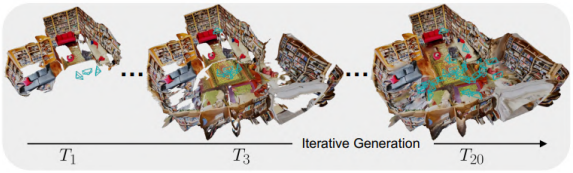


Fig. 2: Generation of 3D image from text.

1) Scene Representation: Consider a dynamic scene represented as a mesh $M = (V, C, S)$, where $V \in \mathbb{R}^{N\times 3}$ denotes vertex coordinates, $C \in \mathbb{R}^{N\times 3}$ represents vertex colors, and $S \in \mathbb{N}^{M\times 3}$ is the set of faces evolving over time.

2) Iterative Scene Generation: Our iterative scene generation methodology follows a sequence of rendering, refining, and repeating steps.

$$I_t, d_t, m_t = \text{generate}(M_t, E_t) \tag{1}$$

Subsequently, we leverage a text-to-image model F_{t2i} to fill in missing pixels based on the textual prompt:

$$\hat{I}_t = F_{t2i}(I_t, m_t, P_t) \tag{2}$$

Unobserved depth values are estimated using a monocular depth predictor F_d within our depth alignment procedure:

$$\hat{d}_t = \text{estimate-and-align}(F_d, I_t, d_t, m_t) \tag{3}$$

Finally, we integrate the novel elements $\{\hat{I}_t, \hat{d}_t, m_t\}$ with the existing mesh using our fusion approach:

$$M_{t+1} = \text{combine}(M_t, \hat{I}_t, \hat{d}_t, m_t, E_t) \tag{4}$$

3) Depth Alignment Strategy: To address the challenge of seamless integration of new and existing content by aligning them accurately in 3D space, we employ a two-stage depth alignment strategy.

4) Mesh Fusion Step: The mesh fusion step involves pixel triangulation, face filtering based on surface normals and edge lengths, and merging the remaining faces with the existing geometry.

a) Generation Stage: During the generation stage, we render predefined trajectories from optimal viewpoints, ensuring an optimal observation distance for each pose.

b) Completion Stage: In the completion stage, we sample additional poses a-posteriori by voxelize the scene into dense uniform cells, discarding those too close to existing geometry.

C. Panoramic Image Generation:

The creation of a panorama involves the generation of multiple perspective views, each possessing a constant horizontal field of view with a degree overlap. We achieved this through a set of eight 515×512 images which consisted of a 90 degree FOV with 45 degrees of overlap.

1) Generation Module: The proposed module generates the images through a simultaneous denoising process. Each noisy latent is fed into a shared UNet architecture, referred to as the multi-branch UNet, predicting noises concurrently. To ensure multi-view consistency, a Correspondence-aware Attention (CAA) block is introduced following each UNet block. The CAA block operates on N feature maps concurrently, performing cross-attention to ensure correspondence among the multi-view features.

2) Correspondence-aware Attention (CAA): The CAA block performs cross-attention on N feature maps concurrently. For each source feature map, denoted as F, it computes a message based on corresponding pixels $\{t_l\}$ in the target feature maps $\{F_l\}$, considering a $K \times K$ neighborhood for each target pixel. The message calculation follows the standard attention mechanism, incorporating position encoding γ based on the 2D displacement between the target and source images. The displacement is frequency-encoded, and the target feature is obtained through bilinear interpolation.

$$M = \sum_{l} \sum_{t_l^* \in N(t_l)} \text{SM}\left(W_Q F\gamma(s) \cdot W_K F_l\gamma(t_l^*) \cdot W_V F_l\gamma(t_l^*)\right)$$

$$F\gamma(s) = F(s) + \gamma(0), \quad F_l\gamma(t_l^*) = F_l(t_l^*) + \gamma(s_l^* - s)$$

3) Panorama Extrapolation: The objective is to generate full 360-degree panoramic views based on a single perspective image and per-view text prompts. SD's inpainting model serves as the base model, and CAA blocks with zero initializations are inserted into the UNet and trained on the datasets. During generation, latents of both target and condition images are reinitialized with noises from standard Gaussians. The UNet branch for the condition image includes a mask of ones, preserving the image content. Conversely, the UNet branch for a target image includes a black image and a mask of zeros, prompting the inpainting model to generate a new image based on text conditions and correspondences with the condition image.

4) Training: CAA blocks are inserted into the pretrained stable diffusion UNet or stable diffusion inpainting UNet to ensure multi-view consistency. The pretrained network is frozen, and the following loss is employed to train the CAA block:

$$\mathcal{L} = \mathbb{E}\left\{\sum_{i=1}^{N}\sum_{t=1}^{T} E(x_i)\right\} + \sum_{i=1}^{N} \|\epsilon_i - \epsilon_i^{\theta}(\mathcal{Z}_{i,t}, t, \tau\theta(y))\|_2^2$$

where $\epsilon_i \sim \mathcal{N}(0, I)$ represents the noise, $E(x_i)$ is the energy term, and $\mathcal{Z}_{i,t}$ denotes the latent variable. The loss ensures consistency between the generated samples and the latent variables.

IV. Experiment and Results

Our experimentation primarily focused on two domains. Initially, we selected a pre-trained diffusion model and worked to improve the NeRF pipeline. Then we explored modifications to the diffusion model architecture for consistent image generation.

We selected the Stable Diffusion model trained on the LAOIN-5B dataset as a good text-to-image model that we further leveraged in this project. Figure 3 shows a sample output of the model when supplied with a text prompt.

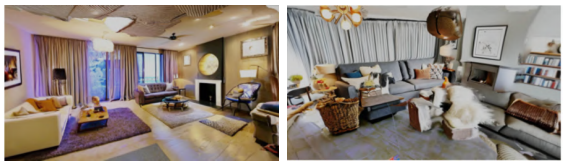


Fig. 3: Text Prompt - *"a well-lit living room with large windows"*

In our pursuit of a good NeRF pipeline, we first adapted a method based on the Text2Room [2] pipeline. The pipeline generated a single 2D image from a standard viewpoint and tried to expand it in all directions to form a complete scene. In order to maintain consistency across the scene, it always retained some part of the original image and used inpainting techniques to fill up the missing information. This approach, although complete, was very slow (several dozens of hours). It also proved hard to reproject the 2D images in space as the apparent depth of the successive images were disoriented by the inpainting techniques. As visible in Figure 4, the sheer number of images generated and their relative inconsistencies with respect to each other, proved to be a bane rather than a boon when stitching the images together.

Working towards a solution, we identified that one possible problem was limiting ourselves to a single starting image. We also realised that trying to generate a comprehensive depth map with fast inference time required devices with more compute power. Hence, in an attempt to retain our primary goal of generating a dynamic virtual background, we simplified the scope of the problem to generate the elements of the scene given a text prompt. In this approach, we understood that a panoramic image of the scene would be the best representation that met our needs. Hence equipped with these two pieces of

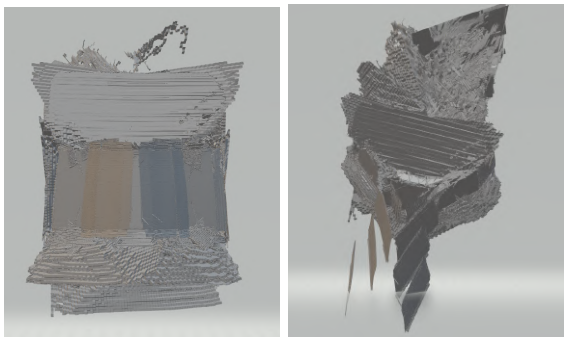


Fig. 4: Output of NeRF sequence for a living toom from inpainting methods

information, we worked on a pipeline that generated multiple views of the scene and tried to stitch them together in the form of a panorama.

The most straightforward implementation of the above technique yielded the following results (Figure 5). Due to the simplicity of the model, we selected this as our baseline to implement and test, we coined this as our baseline model.

Fig. 5: Output of Baseline method

It is visible that the images generated by the ”baseline” model lack consistency across generations and hence the final stitched image is a very poor approximation of the room.

In an attempt to bring about this consistency, we experimented with the architecture of the very model. We adapted from the MVDiffusion [5] pipeline to introduce a cross-patch attention mechanism (CPAttn) that utilizes transformer modules. CPAttn captures scene positional information through a combination of self-attention and positional embeddings.

We froze the pre-trained weights and trained the attention layers on the Matterport3D dataset to train it using MSE Loss (Figure 6) to provide consistent indoor sequences. The results of the model can be seen in Figure 7.

We evaluated individual views generated by our model on the following metrics: Clip Score (CS), Inception Score (IS), FID score and a user-assigned

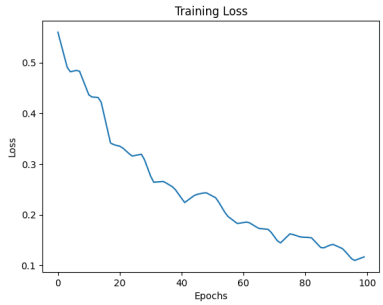


Fig. 6: Training Plot for 100 epochs

score on the panoramic image for photo realism between 1 and 10 (US). From Table I we can see that our model slightly outperforms the individual view generation of the baseline model. The CLIP score is almost the same for both cases as they are both built on the same Text2Image model. The IS shows a slight drop as our attention layers constrain our images to be more consistent and repeatable.

Name	FID	IS	CS	US
our model	22.3	6.57	28.42	8
baseline model	25.59	7.09	28.63	3

TABLE I: Model Performance

V. Analysis of Model

Each element of the model consists of its own pros and cons. Our model heavily depends on the versatility of pre-trained text-to-image models. We have used StableDiffusionv2 in our current implementation. We experimented with StableDiffusionXL [9] to conclude that the SDXL model provided a slight improvement in the quality of generated images but it came at a cost of increased inference time. One implementation of SDXLTurbo (not yet open source) was capable of tackling this problem but we felt it better to use a more tried and test model as a base.

The SDv2 model has its limitations that further propagate upstream to affect the final workings of our model. Figure 8 has a few such examples of possible inconsitencies that could affect the performance of our model.

The text prompt must provide a good description of the scene and must yet not be too restrictive. Text prompts (a) and (b) represent the problem of underparametrization, i.e., the prompt is not descriptive enough. (c) is an example of a good description of a lecture hall, however, it is too restrictive to the model performance. This results in a scene resulting from a poor adaptation of various substrings within the prompt. This proves detrimental to the model performance. (d) and (e) are cases that describe a limitation of creativity of the model. The scene described is so far apart from reality that the model training set does not contain any relatable prompt.

(a) Text Prompt - *"A cozy living room retreat, seamlessly fusing contemporary comfort and timeless elegance. A plush sectional sofa, adorned with neutral throw pillows, anchors the space. To the left, a minimalist fireplace emanates warmth, while to the right, expansive windows reveal a tranquil view."*

(b) Text Prompt - *"A immersive arcade haven, seamlessly blending vintage allure and contemporary zest. To the left, rows of retro arcade cabinets beckon with colorful screens and flashing lights. On the right, modern gaming consoles and sleek multiplayer setups invite friendly competition. The room pulses with the energetic hum of electronic excitement."*

(c) Text Prompt - *"A underwater hotel room with a water bed. Fishes are visible through the glass walls and ceiling and the light blue ocean glistens in the morning sun."*

(d) Text Prompt - *"An airport lounge oasis, seamlessly blending modern convenience and comfort. Plush seating surrounds sleek tables, creating inviting clusters. To the left, a stylish bar beckons with an array of beverages. On the right, panoramic windows offer views of departing planes, and overhead, contemporary lighting bathes the space in a warm, sophisticated glow."*

(e) Text Prompt - *"An office space. Picture a set of cubicles as far as the eye can see and people crouched on their desks and scratching away at pieces of paper or typing in their computers. To the left and right are numerous glass doors leading to multiple conference rooms."*

(f) Text Prompt - *"Futuristic human civilzation. Tall futuristic building to the left and right with 3D bill boards. Flying cars on the road ahead."*

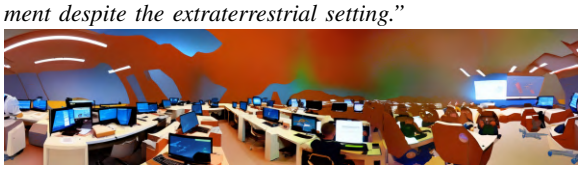
Fig. 7: Outputs of our model

(a) Text Prompt - *"A dozen horses racing on the moon."*

(b) Text Prompt - *"The table of the Last Supper"*

(c) Text Prompt - *"A lecture hall setting. Picture a vast amphitheater-style space, rows of tiered seating facing a central stage, filled with a diverse group of college students. At the centre of the stage is a large screen and a presentation dias. Behind the rows of a seats and on all sides are carpeted noise proof walls with exit signs littered at sparse intervals."*

(d) Text Prompt - *"A Martian classroom, a futuristic blend of technology and adaptability. Students sit in ergonomic, low-gravity chairs, facing holographic screens that display lessons and simulations. To the left, a Martian landscape is visible through large windows, and on the right, advanced life-support systems ensure a comfortable learning environment despite the extraterrestrial setting."*



(e) Text Prompt - *"An open- air Martian classroom, a futuristic blend of technology and adaptability. Students wearing astronaut suits sit facing a screen monitor in front of them. The class has no walls and is situated on the red sands of Mars. Mountains are visible to the distance in the left. To the right in the distance is a futuristic city built on Mars."*

Fig. 8: Analysis of our model

This results in an approximation of the individual words of the prompt but they are not well put together in the final generated image. Well-balanced and informative prompts are crucial for achieving optimal model performance.

It was also understood that despite the large repository of training information available to many of these pre-trained models, individual models have their own salient features. For instance, the above prompts that produced infeasible outputs with SDv2 produced well put and complete outputs using DAL ·E 2 [10]

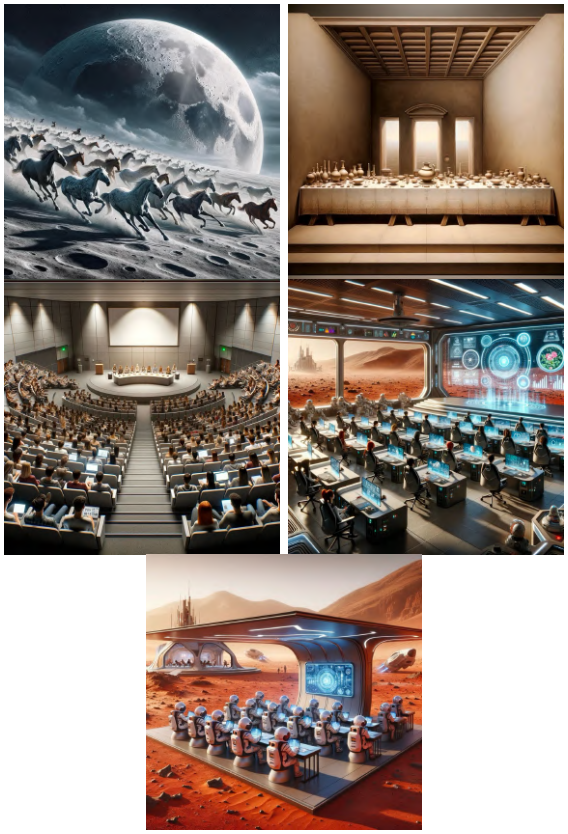


Fig. 9: Output of DALL ·E 2

(Figure 9). This is in line with the expectations that an artistic model is trained to produce more diverse creative solutions to the text provided. However, integrating such licensed models into our pipeline is still a work in progress as they cannot be easily leveraged to generate images from multiple views.

VI. Conclusion

We successfully implemented a Optical Flow pipeline in combination with a Diffusion-based scene generation model to create a dynamic virtual background. In pursuit of this objective, we successfully deployed an Optical Flow pipeline. This innovative system utilizes a text prompt to generate a panoramic image, which is then combined with a user-input video. The panoramic image serves as the foundation for dynamically updating the user's background in response to on-screen movements, detected through feature tracking. For a visual representation of the system's outcomes, please refer to Figure 10.

VII. Challenges

Our model showed the ability to generate consistent images and hence produce decent approximations of the scene. However, the greatest problem lies in the pursuit of a model capable of generating a comprehensive 3D mesh. The advantages the mesh produces are numerous and not limited to the field of computer vision. It can aid robot simulations, it can

(a) Text Prompt - *"This kitchen is a charming blend of rustic and modern, featuring a large reclaimed wood island with marble countertop, a sink surrounded by cabinets. To the left of the island, a stainless-steel refrigerator stands tall. To the right of the sink, built-in wooden cabinets painted in a muted."*

Fig. 10: Sample Results of Overall Pipeline

aid development of AR tools to simulate a new virtual space etc. An even bigger challenge is in developing such a model that can run in real time with limited compute resources that many video call applications can run on such as Android phones etc. One possible workaround would be use text prompts to generate a 3D environment in advance and just leverage such an environment in real-time for any said application.

References

[1] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion, 2022.

[2] Lukas Höllein, Ang Cao, Andrew Owens, Justin Johnson, and Matthias Nießner. Text2room: Extracting textured 3d meshes from 2d text-to-image models, 2023.

[3] Liangchen Song, Liangliang Cao, Hongyu Xu, Kai Kang, Feng Tang, Junsong Yuan, and Yang Zhao. Roomdreamer: Text-driven 3d indoor scene synthesis with coherent geometry and texture, 2023.

[4] Jingbo Zhang, Xiaoyu Li, Ziyu Wan, Can Wang, and Jing Liao. Text2nerf: Text-driven 3d scene generation with neural radiance fields, 2023.

[5] Shitao Tang, Fuyang Zhang, Jiacheng Chen, Peng Wang, and Yasutaka Furukawa. Mvdiffusion: Enabling holistic multi-view image generation with correspondence-aware diffusion, 2023.

[6] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022.

[7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, 2017.

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

[9] 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.

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