Bringing Static 3D Scenes to Life: Language-Guided Video Diffusion for Dynamic 4D Generation

Chenyang Xu*
New York University
New York, USA
cx2219@nyu.edu

Changyue Su*
New York University
New York, USA
cs7483@nyu.edu

Sihang Li*
New York University
New York, USA
sl10496@nyu.edu

Anbang Yang*
New York University
New York, USA
ay1620@nyu.edu

***Abstract*—'Liveliness' is the key component in creating engaging 3D experiences. With cutting-edge novel view synthesis capability and efficiency, 3D Gaussian Splatting technology has become a crucial technique for reconstructing 3D environments in fields like computer vision, robotics, and virtual reality. Based on such powerful representation, dynamically selecting and moving objects via language instruction and converting static 3D scenes into 4D motion representation, is a natural next step while presenting significant challenges. Current language-driven editing approaches often focus on adding, removing objects, or style transfer, limiting their practical use in dynamic and interactive applications. This paper addresses the problem of dynamic object manipulation in static 3D Gaussian models with language instruction by proposing a novel method that allows for the dynamic movement of objects in static images while preserving the 3D structure of the reconstruction. Our framework combines 3D Gaussian Splatting reconstruction with text-guided video diffusion models to transform static scene images into dynamic, interactive 4D environments. Leveraging advancements in language-guided video diffusion, 3D Gaussian Segmentation, and compositional 4D scene representation to refine depth information and integrate language-embedded content, our method enables flexible, realistic manipulations within our open world. The results demonstrate the potential of our approach for applications such as dynamic scene simulation, robotic navigation, and interactive virtual environments. Our project repository can be found at: https://github.com/Louis-Leee/Animate3DGS**

I. Introduction

The ability to reconstruct 3D environments from visual data has become a cornerstone in modern computer vision and robotics. 3D Gaussian models [1] have gained attention due to their probabilistic nature and novel view synthesis, which allows for a more flexible representation of scenes, accounting for uncertainties in object shapes, textures, and positions.

However, while static 3D reconstructions have been extensively studied, dynamically altering these reconstructed environments remains a challenging and underexplored area. The ability to move objects within a probabilistic reconstruction is vital for several emerging applications. In robotics, for example, a robot may need to interact with a dynamic environment, where objects shift or are repositioned. Similarly, in virtual and augmented reality applications, users often expect realistic and flexible interactions with their environment, requiring objects to move or be altered without breaking the consistency of the scene. Traditional methods for manipulating objects within a reconstructed space are often designed for rigid models that lack probabilistic flexibility. These methods can fail when dealing with Gaussian models, as the relationships between object locations and the probabilistic properties of the scene are more complex. Ensuring that the new location of an object is consistent with the rest of the scene, both in terms of visual and probabilistic accuracy, is a non-trivial task.

Recent innovations in object-centered 3D segmentation and 4D content creation have offered new frameworks for dynamic scene manipulation [2], [3]. By focusing on the object itself as a central unit in the scene, these methods enable more modular and flexible adjustments within complex 3D environments. Furthermore, the rise of large language models (LLMs) and diffusion models [4] has opened doors to new interaction paradigms, where users can guide the movement of objects through natural language, making dynamic content manipulation more intuitive.

In this paper, we build upon these recent advancements to tackle the problem of object manipulation within reconstructed 3D Gaussian models. Specifically, we focus on the challenge of moving objects in static images while preserving the probabilistic integrity of the Gaussian model. Our approach combines 3D Gaussian Splatting reconstruction with text-guided video diffusion models to transform static scene images into dynamic, interactive 4D environments, ensuring that objects can be repositioned without introducing inconsistencies in the reconstruction. By leveraging the strengths of language-guided video diffusion, 3D Gaussian segmentation, and compositional 4D scene representation, our method offers a robust solution for flexible object manipulation in reconstructed 3D Gaussian environments.

Our main contributions are as follows:

- An interactive 3D reconstruction pipeline that converts static images into 4D dynamic Gaussian Splatting representation through video diffusion.
- A language-guided object manipulation method that segments and moves objects in static scenes while maintaining the probabilistic integrity of the Gaussian model.

- A compositional 4D scene that represents the dynamic motion and trajectory of object movements according to user-defined object manipulation tasks.

II. Related Work

A. 3D Gasussian based Editing

Generating and inserting new objects into 3D content is a fundamental task for 3D scenes. MVinpainter [5] designs a multi-view diffusion model to generate scenes and further designs a mask-aware 3D reconstruction technique to refine Gaussian Splatting reconstruction. GaussianEditor [6] is a more generalized tool that can complement object insertion, removal, and object decoration. Compared to these, the target objects edited in our project are not generated but are parts from the original scene, and the interaction of our method is further assisted by LLMs. As introduced before, GSEdit [7] is a pipeline for text-guided 3D object editing based on Gaussian Splatting models; However, its editing is only object-oriented appearance style transferring, instead of object movement. OpenGaussian [8] is another language-assisted method based on 3D Gaussian Splatting capable of 3D point-level open vocabulary understanding. It can complete the tasks of object segmentation and searching in the 3D scene, which might be instrumental to our project. Furthermore, 4D editing requires the reasonable movement trajectories of objects.

B. LLM-Guided Video Diffusion

To generate high-quality videos, several methods have been developed. A recently popular approach is using Diffusion-based models [9]–[11] which generate high-quality videos using conditional 3D U-Nets to denoise sequences of Gaussian noise. Recent foundation models [9], [12], [13] are pre-trained on large image and video datasets [14], enabling powerful generation capabilities. Customizing the pre-trained large foundation models can fit the preferences of users better while maintaining powerful generation knowledge without training from scratch. Methods like Tune-A-Video [15], Text2Video-Zero [16], and AnimateDiff [17] leverage pre-trained text-to-image diffusion models, extending them with temporal layers or cross-frame attention mechanisms to introduce motion consistency across frames. Diffusion-based video generation has become a cornerstone for text-to-video tasks [18], offering scalable and adaptable solutions for generating dynamic language-guided content. Our work incorporates the recent advancements in text-to-video diffusion models into static object motion generation, which gives a dynamic movement trajectory to the reconstructed 3D object in our open scene.

C. 3D Open-Vocabulary Scene Understanding

Recent methods have attempted to incorporate semantics into 3D scene representations for better scene understanding and segmentation tasks. Incorporating accurate semantics into implicit scene representation to build a 3D language field is difficult. Many methods try to embed semantic information into NeRF's [19] implicit MLP-based representation [20]–[25]. In the early stage, Distilled Feature Fields [24] and Neural Feature Fusion Fields [25] were constructed by distilling multi-view features from LSeg [26] and DINO [27] into NeRF. Shen et al. [28] leveraged this feature distillation for language-guided robotic tasks. Semantic NeRF [22] introduced a novel way to jointly encode appearance, geometry, and semantics for semantic view synthesis. LERF [29] was a pioneering approach in embedding CLIP [30] features directly into NeRF, enabling open-vocabulary 3D scene queries.

However, many of these methods face limitations in training and rendering due to the limitations of NeRFs. Methods such as 3D-OVS [23] decompose scenes into dual tensors for geometry and semantics but require pre-defined classes. In contrast, recent advancements like LangSplat [31] and LEGaussians [32] seek to efficiently model open-vocabulary 3D language fields using novel scene representation method 3D Gaussian Splatting [1], addressing both semantic flexibility and computational efficiency challenges inherent in previous NeRF-based approaches.

D. LLM-Guided Compositional 4D Gaussian Generation

In our project, the editing results of the 3D Gaussian scene must adhere to real-world common sense. For instance, overlapping objects, such as a cup partially sinking into a table's surface, are not permissible. A reasonable trajectory design is essential to address this issue. The trajectory can also facilitate the representation of any intermediate dynamic states of the objects, displayed as a 4D Gaussian scene throughout the editing process. Users can select their desired result from these intermediate states. Recently, two concurrent works Comp4D [33] and TC4D [3] also achieved compositional 4D generation (i.e., dynamic 3D generation). However, they only considered the composition between objects, and the trajectory of these methods is either formulated by kinematics-based equations [33] or pre-defined by users [3]. Differently, we explore 3D-aware 4D gaussian generation with integrated 3D scenes and compose various concepts with priors LLM.

III. Methods

In this section, we illustrate the components of our proposed method in detail. We start by introducing language-embedded 3D Gaussians, then two other main components: A powerful driving signal for motion generation and an effective way to distill this motion into the 3D scene while keeping the scene appearance and generating motion realistic. The diffusion model's guidance must remain closely tied to the specified 3D scene and achieve the highest possible multi-view consistency, while the process of translating this signal into the 3D environment should be maximally efficient.

A. Preliminary

3D Gaussian Splatting (3DGS). Given a training dataset $\mathcal{I}$ of multi-view 2D images with camera poses, 3D-GS learns a set of 3D colored Gaussians $\mathcal{G} = \{\mathbf{g}_1, \mathbf{g}_2, ..., \mathbf{g}_N\}$, where N denotes the number of 3D Gaussians in the scene. The mean of a Gaussian represents its position and the covariance indicates its scale. Accordingly, 3D-GS proposes a novel

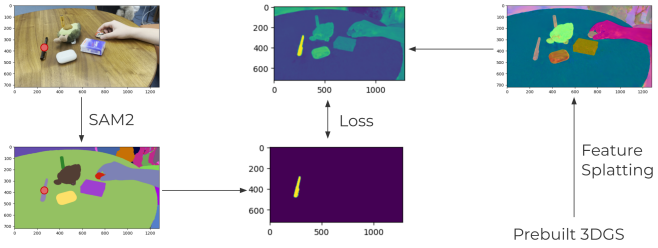

Fig. 1. Overall training pipeline of our LifeGS.

differentiable rasterization technology for efficient training and rendering. Given a specific camera pose, 3D-GS projects the 3D Gaussians to 2D and computes the color $\mathbf{C}(\mathbf{p})$ of a pixel $\mathbf{p}$ by blending a set of ordered Gaussians $\mathcal{G}_\mathbf{p}$ overlapping the pixel. Let $\mathbf{g}_i^\mathbf{p}$ denote the i-th Gaussian in $\mathcal{G}_\mathbf{p}$, this process is formulated as:

$$\mathbf{C}(\mathbf{p}) = \sum_{i=1}^{|\mathcal{G}_\mathbf{p}|} \mathbf{c}_{\mathbf{g}_i^\mathbf{p}} \alpha_{\mathbf{g}_i^\mathbf{p}} \prod_{j=1}^{i-1} (1 - \alpha_{\mathbf{g}_j^\mathbf{p}}), \tag{1}$$

where $\mathbf{c}_{\mathbf{g}_i^\mathbf{p}}$ is the color of $\mathbf{g}_i^\mathbf{p}$ and $\alpha_{\mathbf{g}_i^\mathbf{p}}$ is given by evaluating the corresponding 2D Gaussian with covariance Σ multiplied with a learned per-Gaussian opacity.

B. Language-embedded 3DGS

Given a pre-trained 3D-GS model $\mathcal{G}$, LifeGS attaches a Gaussian affinity feature $\mathbf{f_g} \in \mathbb{R}^D$ for each 3D Gaussian $\mathbf{g}$ in $\mathcal{G}$. D denotes the feature dimension. To handle the inherent multi-granularity ambiguity of 3D promptable segmentation,

To train the affinity features, LifeGS extracts a set of multi-granularity masks $\mathcal{M}_\mathbf{I} = \{\mathbf{M}_\mathbf{I}^i \in \{0,1\}^{HW} | i = 1, ..., N_\mathbf{I}\}$ for each image $\mathbf{I}$ in the training set $\mathcal{I}$ with SAM. H, W denotes the height and width of $\mathbf{I}$ respectively. $N_\mathbf{I}$ denotes the number of extracted masks. For each mask $\mathbf{M}_\mathbf{I}^i$, its corresponding 3D physical scale $s_{\mathbf{M}_\mathbf{I}^i}$ is calculated using the depth predicted by $\mathcal{G}$ with the camera pose. Subsequently, LifeGS employs a contrastive learning strategy to distill the multi-granularity segmentation ability embedded in multi-view 2D masks into the affinity features. After training, the affinity feature similarities between two Gaussians indicate whether they belong to the same 3D target. During inference, given a specific viewpoint, LifeGS takes 2D visual prompts as input and converts them to corresponding 3D query features, which are then used to segment the 3D target by evaluating feature similarities with the 3D affinity features. Additionally, with the well-trained affinity features, 3D scene decomposition can be achieved through straightforward clustering. Furthermore, by integrating with CLIP, LifeGS can perform open-vocabulary segmentation without the need for additional language fields.

To achieve open-vocabulary segmentation in 3D, traditional methods like LERF and LangSplat align 3D language feature fields with visual features extracted by CLIP, enabling segmentation based on textual prompts. Training these feature fields involves fusing information from multiple views, resembling a "voting" process. Inspired by this idea, we propose a strategy for open-vocabulary segmentation using LifeGS's Gaussian affinity features, requiring minimal modification. To identify 3D objects, we need to group 2D masks across different views. We introduce **global features** using segmented Gaussians, which remain consistent. Here's how it works:

1) **Anchor Gaussians:** We sample anchor Gaussians ($\mathcal{A}$) as reference points.
2) **Feature Extraction:** For a 2D mask ($\mathbf{M}$), we calculate its affinity feature by averaging pixel features within the mask:
$$\mathbf{f_M} = \frac{1}{\delta(\mathbf{M})} \sum_{\mathbf{p} \in \delta(\mathbf{M})} \mathbf{F}^{s_\mathbf{M}}(\mathbf{p}).$$
3) **Similarity Matching:** We compare this feature with the anchor Gaussians to determine which Gaussians are most similar, resulting in a segmentation result for each mask:
$$\mathcal{A}_\mathbf{M} = \{\mathbf{g} \mid \langle \mathbf{f}_\mathbf{g}^{s_\mathbf{M}}, \mathbf{f_M} \rangle > \tau, \mathbf{g} \in \mathcal{A}\}.$$
4) **Distance Calculation:** The distance between two masks is defined based on how much their segmentation results overlap (intersection over union):
$$D(\mathbf{M}_1, \mathbf{M}_2) = \frac{|\mathcal{A}_{\mathbf{M}_1} \cap \mathcal{A}_{\mathbf{M}_2}|}{|\mathcal{A}_{\mathbf{M}_1} \cup \mathcal{A}_{\mathbf{M}_2}|}.$$
5) **Clustering:** Using this distance measure, we cluster the 2D masks into groups representing potential 3D targets.

Once we've clustered the masks, we create a vote graph where each cluster corresponds to a 3D object or part. For each 2D mask, we extract visual features using CLIP and calculate a relevancy score based on the similarity between textual prompts and the mask's visual features. For each cluster (3D target), the scores of its associated masks are averaged to determine its final relevancy score:

$$r_\mathbf{T} = \frac{1}{|\mathcal{V}_\mathbf{T}|} \sum_{\mathbf{M} \in \mathcal{V}_\mathbf{T}} r_\mathbf{M},$$

where $\mathcal{V}_\mathbf{T}$ is the set containing all 2D masks corresponding to the cluster centroid $\mathbf{T}$. The cluster is labeled with the category that has the highest score, enabling semantic segmentation.

C. Motion Video Generation for Trajectory Design

From the previous steps, we obtained the segmented 3D static Gaussian Splatting object from the original scene. The next step of the pipeline is to design the trajectory of the motion of the object which can guide the generation of the 4D Gaussian Splatting scene.

There are some desired properties for the trajectory generation. Firstly, it need to be realistic and physics constrained, for example, it must avoid any colliding, overlapping or walking across other objects. The trajectory can not be out of the coordinates of the scene. Since the 4D scene might be used for generating training data for robotics, a certain diversity of trajectory is desired so that there can be more data generated from a limited amount of given input data. Lastly, we want to use the language prompts to guide the trajectory generation.

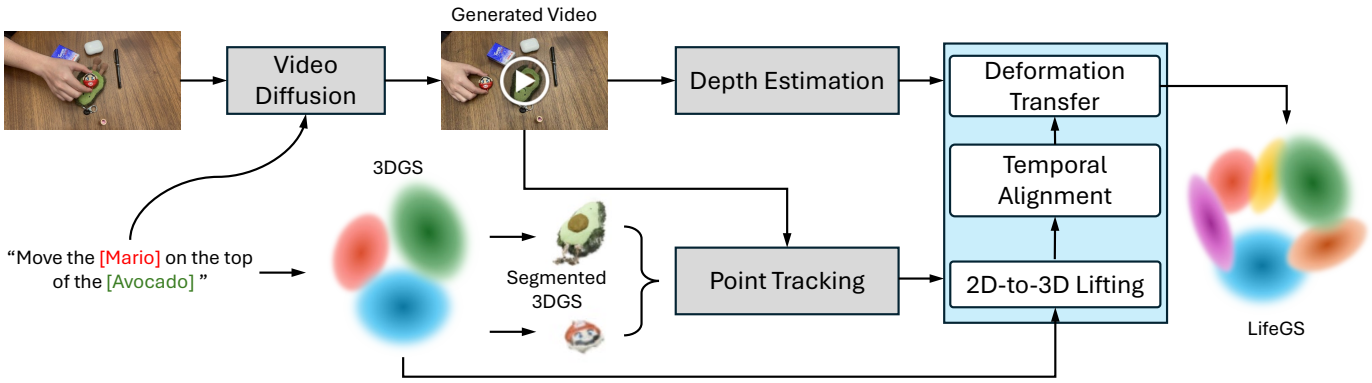

Fig. 2. Our overall pipeline, which combines the static LifeGS and generated video of the motion, prompted by language instruction.

In conclusion, this is a Text-Image-to-Video (TI2V) task, which takes the scene image (s) and the user-designed trajectory prompt to generate realistic motion video of objects [34]. There are many existing methods for TI2V tasks and we explored three methods that are optimized for generating moving objects video specifically. They are make-it-move [34], SuSIE [35] and Seer [36]. Among these methods, make-it-move provides the videos of a high quality with the clearest trajectory. However, all of them have the same issue, weak capability of doing zero-shot tasks. Eventually, we decided to use the commercial diffusional model based video generation model, Runway AI[1] and KLing AI[2]. We implemented some techniques to optimize them specifically for our tasks, using first and last frame of the motion as input, tuning prompt to help model focus on generating the motion only, and adding hands or robot arms to help lifeless object move more realistic. In the end, these two diffusional model completed the motion specific TI2V sub-task significantly well.

Below is a potential subsection to be added after the "Motion Video Generation for Trajectory Design" subsection, describing the depth estimation process on the generated motion video.

D. Depth Estimation on Generated Motion Video

After obtaining the motion video from our Text-Image-to-Video (TI2V) approach, we aim to achieve a more comprehensive 4D representation by incorporating depth information into each frame. Accurate per-frame depth estimation enables a deeper geometric understanding of the scene beyond simple 2D motion cues, providing valuable insights into object positions, spatial relationships, and scene structure. This, in turn, facilitates more advanced downstream tasks such as robotic navigation, 3D reconstruction, and interaction planning within the generated environment. Additionally, these depth maps serve as a crucial verification tool, allowing us to refine and validate the moving object's trajectory against the original static environment, ensuring physically plausible and collision-free motion paths.

[1]https://runwayml.com/
[2]https://klingai.com/

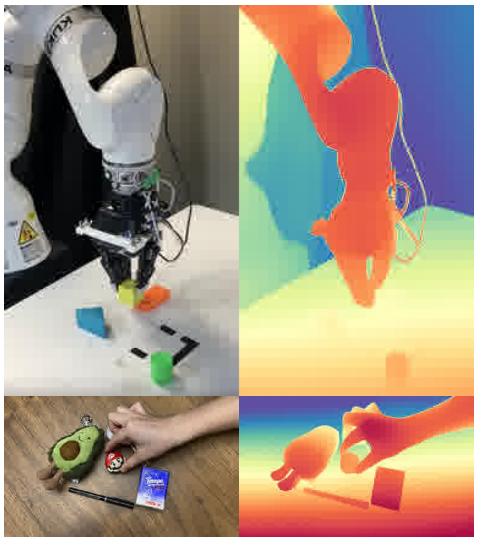

Fig. 3. Comparison between the generated motion video (top) and the corresponding estimated depth maps (bottom). The depth estimation provides valuable geometric context that refines the object trajectories and ensures more plausible, physically consistent 4D scene representations.

To generate these depth maps, we leverage state-of-the-art monocular depth estimation models, such as MiDaS [37], DPT [38], or Depth Any Video [39]. These models have demonstrated robust performance across a wide range of image conditions, including those involving synthetic or diffusion-stylized content, benefiting from extensive training on large-scale and diverse datasets. As a result, the predicted depth maps exhibit strong generalization capabilities and remain reliable even under challenging visual conditions. Moreover, we can further refine the consistency of these estimated depth maps temporally by applying smoothing techniques, optical flow constraints, or by incorporating prior knowledge from the initial 3D Gaussian splatting representation. Doing so helps maintain stable depth predictions over the duration of the video, minimizing visual artifacts such as flickering and ensuring that the reconstructed scene exhibits coherent geometry at all time steps.

Overall, integrating depth estimation into the pipeline en-

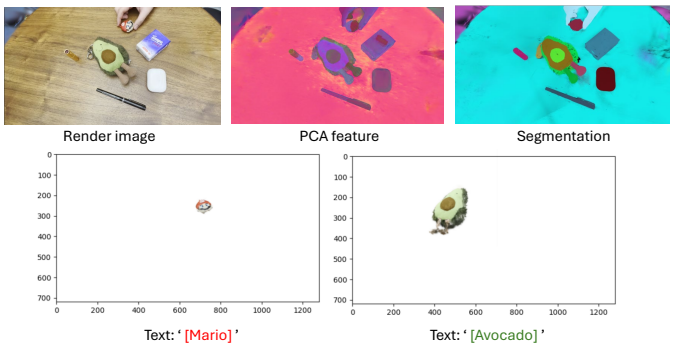

Fig. 4. From our language-embedded 3DGS, we can easily segment the target gaussians based on the language prompt. Meanwhile, the render image quality if satisfactory.

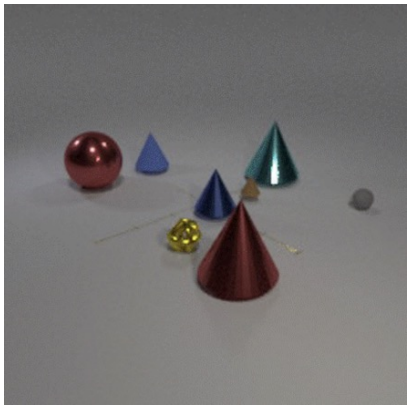

Fig. 5. An example image from the data set CATER

Fig. 6. The screen shots of the video that changes the camera position generated by Seer

riches the generated motion video with critical geometric insights. By complementing the semantic and spatial cues from the TI2V models, depth estimation leads to a more robust and veridical 4D Gaussian splatting representation that is both visually compelling and physically grounded.

IV. Experiments

A. Compare TI2V motion video generation Methods

We explored some existing methods to complete the TI2V task specifically designed for generating motion video to extract trajectory. Optimally, the desired inputs should be the image of the initial scene status and a user prompt of the object motion description. Each of these methods can complete the task partially but all failed eventually, mainly because of the low zero-shot task capability. We will discuss the pros and cons of the three methods that we tried:

- Make-It-Move uses transformer to generate a Motion Anchor (MA) and use it to assist a transformer auto-regressive video generator [16]. MA is trained by a attention model on both images and texts prompt and this process is capable of understanding the position instructions, ranging from clear coordination information to vague orientation words. Due to the motion anchor, the generative model can also focus on generating the motion of object and the result shows very smooth and realistic motion videos that can extract trajectory easily.
  However, our experiments show that it has almost no capability of doing zero-shot tasks. The models are only trained specifically to one 3D scene dataset, CATER, which only includes the motion video of simple geometric objects as shown in the figure 5. The models performed all the ideal properties described above on this dataset but failed to expand to zero-shot tasks.
- Seer is a latest open-source diffusional model based video synthesis method. It inflates the retrained text-to-image stable diffusion models along the temporal axis and uses auto-regressive spatial-temporal attention and frame sequential text decomposer to generate prior knowledge [17].
  It shows a great result on the robot arm scene, which might be due to robot arm video augmented to the training dataset. However, when it comes to other scene, there are many issues: the camera posture oscillates frequently, which must hinder the trajectory extraction. A example can be viewed in the figure 6. The prompt flexibility of this method is weak and the complicated guidance cannot be understood. The inputs needs two successive initial frames which are relatively hard to obtain. What's more, the commercial diffusional model video synthesis products provides way better video generation quality and the codes of this methods also lack of maintenance. Therefore, this method was not implemented.
- SuSIE is a method to generates sequential images with a short intervals instead of a video, which aims to assist a robotic manipulation system [35]. Given an image and a language command, SuSIE executes the command by "editing" the image into a meaningful subgoal and then making the robot arm to achieve that subgoal using a low-level goal-reaching policy. It alternates generating sub-goals using an image-editing diffusion model. Although it is not a TI2V task, the time intervals are short enough to extract a trajectory so suitable for our goal. Failure examples can be found at the appendix of the SuSIE paper [35].
  Our experiments show that the SuSIE has certain capability of zero-short tasks. However, it still has a issue of being unaware of the low-level policy and often generate a failed intermediate image, which leads to the failure of

the whole sequence generation.

B. Commercial Video Synthesis Methods Implementation Details

In order to solve the zero-shot tasks issue, use flexible prompts and achieve a high-resolution motion video, we decided to use commercial video synthesis methods. Runway AI and KLing AI are the methods that we used. Both of them are able to achieve exactly equivalently well results. The only difference is that Runway AI is significantly faster than KLing AI for the same task, usually 10 seconds vs 10 minutes. So Runway AI is preferred. Because these methods are for the generic video synthesis, there are several issues that we implemented some application tricks to fix. I will go through them:

- The first issue is the excess diversity of diffusional model. The form of inputs we tried at first is only one initial frame and prompt text. The result video not only generate moving object but also generates moving camera posture, new scenes, new objects. All of these objects are not desired since they will hindered the extraction of trajectory. For example, the moving camera posture requires additional calibration for trajectory coordinates calculation in the 3D scene.
  Our solution is to uses two images as inputs instead of only one images, using the images of the initial position and the ending position as the first and last frames. More conditions succeeded to reduce the diversity of the video generation and force it to focus on motion generation only. This is a compromise method as using the initial and ending frame will increases the difficulty of data collection. One image input is still desired. The future work can be designing other ways to add conditions, e.g. prompt, or predict the last frame given the first frame and prompt. A comparison between the results before and after changing the input images number can be viewed in the figure 7
- The second issue is that the condition we provided must follow the rules in the physics worlds. There is a popular augments "Can a human image something that does not exist?". For diffusional model, at least, the thing it "images" would better be included in the training data set. For the lifeless object e.g. a pencil case, if we prompt to make it move while not showing any hands or other kinds of handlers to assist the moving, the diffusional model will very likely introduce a handler or just makes some unrealistic moving, e.g. moving across other objects, doing random moving. Two examples of these trials can be view in the figure 8. These are not stable and undesired for trajectory extraction. Therefore, we always show a hand in our trials.
- There are some other minor rules to follow for generating the object motion video. For example, the prompt format should follow "object+action" instead of "action+subject". "A bottle moves from right to left" make a better emphasis on the object "bottle" than the phrasing "moving a bottle from right to left". For the latter phrasing, the moving object can be either bottle or handlers due to the ambiguity.

V. CONCLUSION

In this paper, we have explored the challenges and solutions for manipulating objects within reconstructed 3D Gaussian models, particularly focusing on moving objects from one location to another while maintaining the integrity of the scene. Traditional approaches often struggle to handle such manipulations without introducing inconsistencies in the probabilistic structure of the model. Our proposed method addresses this limitation by combining spatial transformations with updates to the Gaussian parameters, ensuring that objects can be repositioned accurately within the scene.

Building on advancements in language-embedded 3D Gaussian, compositional 4D scene representation, and LLM-guided video generation for trajectory design, our approach provides a flexible and robust framework for dynamic content manipulation in probabilistic 3D environments. The ability to alter object positions without sacrificing scene coherence opens up new possibilities in a range of applications, from robotics and autonomous systems to virtual and augmented reality. Future work will explore scaling our approach to handle more complex, large-scale scenes, as well as investigating further integration with semantic understanding techniques to enhance the interaction between objects and their environment. We believe this work represents an important step toward more dynamic and interactive 3D scene manipulation, with significant implications for both research and practical applications.

REFERENCES

[1] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," 2023. [Online]. Available: https://arxiv.org/abs/2308.04079

[2] J. Cen, J. Fang, C. Yang, L. Xie, X. Zhang, W. Shen, and Q. Tian, "Segment any 3d gaussians," 2024. [Online]. Available: https://arxiv.org/abs/2312.00860

[3] S. Bahmani, X. Liu, Y. Wang, I. Skorokhodov, V. Rong, Z. Liu, X. Liu, J. J. Park, S. Tulyakov, G. Wetzstein *et al.*, "Tc4d: Trajectory-conditioned text-to-4d generation," *arXiv preprint arXiv:2403.17920*, 2024.

[4] X. Li, W. Chu, Y. Wu, W. Yuan, F. Liu, Q. Zhang, F. Li, H. Feng, E. Ding, and J. Wang, "Videogen: A reference-guided latent diffusion approach for high definition text-to-video generation," 2023. [Online]. Available: https://arxiv.org/abs/2309.00398

[5] H. Zhong, C. Wang, J. Zhang, and J. Liao, "Generative object insertion in gaussian splatting with a multi-view diffusion model," 2024. [Online]. Available: https://arxiv.org/abs/2409.16938

[6] Y. Chen, Z. Chen, C. Zhang, F. Wang, X. Yang, Y. Wang, Z. Cai, L. Yang, H. Liu, and G. Lin, "Gaussianeditor: Swift and controllable 3d editing with gaussian splatting," 2023. [Online]. Available: https://arxiv.org/abs/2311.14521

[7] F. Palandra, A. Sanchietti, D. Baieri, and E. Rodolà, "Gsedit: Efficient text-guided editing of 3d objects via gaussian splatting," 2024. [Online]. Available: https://arxiv.org/abs/2403.05154

[8] Y. Wu, J. Meng, H. Li, C. Wu, Y. Shi, X. Cheng, C. Zhao, H. Feng, E. Ding, J. Wang, and J. Zhang, "Opengaussian: Towards point-level 3d gaussian-based open vocabulary understanding," 2024. [Online]. Available: https://arxiv.org/abs/2406.02058

[9] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet, "Video diffusion models," 2022. [Online]. Available: https://arxiv.org/abs/2204.03458

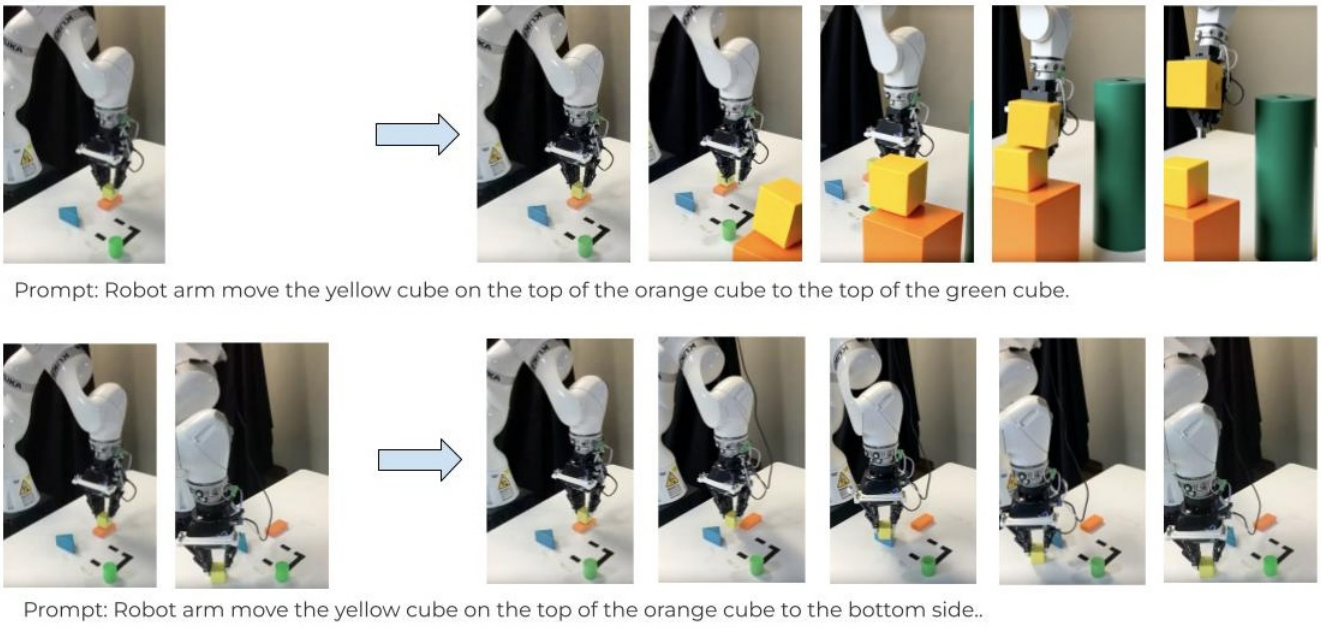

Fig. 7. Runway AI motion video synthesis result comparison, using single frame as input and both initial and ending frames as input. Videos of 5 seconds

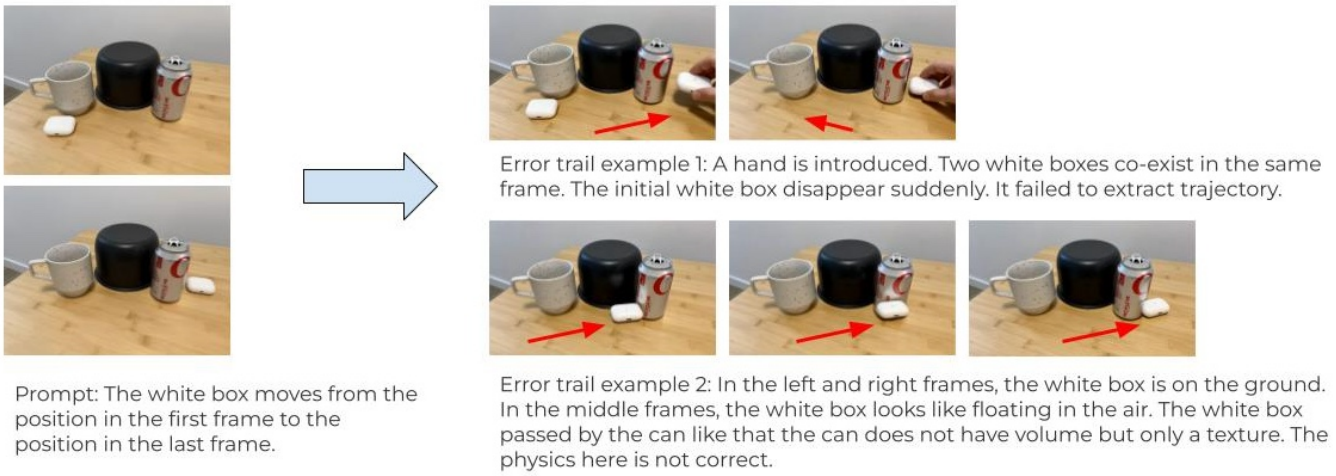

Fig. 8. Runway AI motion video synthesis error trials examples, on condition that there are not a handler given in the images to assist the motion of lifeless objects

[10] H. Ni, C. Shi, K. Li, S. X. Huang, and M. R. Min, "Conditional image-to-video generation with latent flow diffusion models," 2023. [Online]. Available: https://arxiv.org/abs/2303.13744

[11] Y. He, T. Yang, Y. Zhang, Y. Shan, and Q. Chen, "Latent video diffusion models for high-fidelity long video generation," 2023. [Online]. Available: https://arxiv.org/abs/2211.13221

[12] D. J. Zhang, J. Z. Wu, J.-W. Liu, R. Zhao, L. Ran, Y. Gu, D. Gao, and M. Z. Shou, "Show-1: Marrying pixel and latent diffusion models for text-to-video generation," 2023. [Online]. Available: https://arxiv.org/abs/2309.15818

[13] A. Blattmann, R. Rombach, H. Ling, T. Dockhorn, S. W. Kim, S. Fidler, and K. Kreis, "Align your latents: High-resolution video synthesis with latent diffusion models," 2023. [Online]. Available: https://arxiv.org/abs/2304.08818

[14] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, P. Schramowski, S. Kundurthy, K. Crowson, L. Schmidt, R. Kaczmarczyk, and J. Jitsev, "Laion-5b: An open large-scale dataset for training next generation image-text models," 2022. [Online]. Available: https://arxiv.org/abs/2210.08402

[15] J. Z. Wu, Y. Ge, X. Wang, W. Lei, Y. Gu, Y. Shi, W. Hsu, Y. Shan, X. Qie, and M. Z. Shou, "Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation," 2023. [Online]. Available: https://arxiv.org/abs/2212.11565

[16] L. Khachatryan, A. Movsisyan, V. Tadevosyan, R. Henschel, Z. Wang, S. Navasardyan, and H. Shi, "Text2video-zero: Text-to-image diffusion

models are zero-shot video generators," 2023. [Online]. Available: https://arxiv.org/abs/2303.13439
[17] Y. Guo, C. Yang, A. Rao, Z. Liang, Y. Wang, Y. Qiao, M. Agrawala, D. Lin, and B. Dai, "Animatediff: Animate your personalized text-to-image diffusion models without specific tuning," 2024. [Online]. Available: https://arxiv.org/abs/2307.04725
[18] R. Sun, Y. Zhang, T. Shah, J. Sun, S. Zhang, W. Li, H. Duan, B. Wei, and R. Ranjan, "From sora what we can see: A survey of text-to-video generation," 2024. [Online]. Available: https://arxiv.org/abs/2405.10674
[19] 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," 2020. [Online]. Available: https://arxiv.org/abs/2003.08934
[20] F. Engelmann, F. Manhardt, M. Niemeyer, K. Tateno, M. Pollefeys, and F. Tombari, "Opennerf: Open set 3d neural scene segmentation with pixel-wise features and rendered novel views," 2024. [Online]. Available: https://arxiv.org/abs/2404.03650
[21] Y. Siddiqui, L. Porzi, S. R. Buló, N. Müller, M. Nießner, A. Dai, and P. Kontschieder, "Panoptic lifting for 3d scene understanding with neural fields," 2022. [Online]. Available: https://arxiv.org/abs/2212.09802
[22] S. Zhi, T. Laidlow, S. Leutenegger, and A. J. Davison, "In-place scene labelling and understanding with implicit scene representation," 2021. [Online]. Available: https://arxiv.org/abs/2103.15875
[23] K. Liu, F. Zhan, J. Zhang, M. Xu, Y. Yu, A. E. Saddik, C. Theobalt, E. Xing, and S. Lu, "Weakly supervised 3d open-vocabulary segmentation," 2024. [Online]. Available: https://arxiv.org/abs/2305.14093
[24] S. Kobayashi, E. Matsumoto, and V. Sitzmann, "Decomposing nerf for editing via feature field distillation," 2022. [Online]. Available: https://arxiv.org/abs/2205.15585
[25] V. Tschernezki, I. Laina, D. Larlus, and A. Vedaldi, "Neural feature fusion fields: 3d distillation of self-supervised 2d image representations," 2022. [Online]. Available: https://arxiv.org/abs/2209.03494
[26] B. Li, K. Q. Weinberger, S. Belongie, V. Koltun, and R. Ranftl, "Language-driven semantic segmentation," 2022. [Online]. Available: https://arxiv.org/abs/2201.03546
[27] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," 2021. [Online]. Available: https://arxiv.org/abs/2104.14294
[28] W. Shen, G. Yang, A. Yu, J. Wong, L. P. Kaelbling, and P. Isola, "Distilled feature fields enable few-shot language-guided manipulation," 2023. [Online]. Available: https://arxiv.org/abs/2308.07931
[29] J. Kerr, C. M. Kim, K. Goldberg, A. Kanazawa, and M. Tancik, "Lerf: Language embedded radiance fields," 2023. [Online]. Available: https://arxiv.org/abs/2303.09553
[30] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," 2021. [Online]. Available: https://arxiv.org/abs/2103.00020
[31] M. Qin, W. Li, J. Zhou, H. Wang, and H. Pfister, "Langsplat: 3d language gaussian splatting," 2024. [Online]. Available: https://arxiv.org/abs/2312.16084
[32] J.-C. Shi, M. Wang, H.-B. Duan, and S.-H. Guan, "Language embedded 3d gaussians for open-vocabulary scene understanding," 2023. [Online]. Available: https://arxiv.org/abs/2311.18482
[33] D. Xu, H. Liang, N. P. Bhatt, H. Hu, H. Liang, K. N. Plataniotis, and Z. Wang, "Comp4d: Llm-guided compositional 4d scene generation," 2024. [Online]. Available: https://arxiv.org/abs/2403.16993
[34] Y. Hu, C. Luo, and Z. Chen, "Make it move: Controllable image-to-video generation with text descriptions," 2022. [Online]. Available: https://arxiv.org/abs/2112.02815
[35] K. Black, M. Nakamoto, P. Atreya, H. Walke, C. Finn, A. Kumar, and S. Levine, "Zero-shot robotic manipulation with pretrained image-editing diffusion models," 2023. [Online]. Available: https://arxiv.org/abs/2310.10639
[36] X. Gu, C. Wen, W. Ye, J. Song, and Y. Gao, "Seer: Language instructed video prediction with latent diffusion models," 2024. [Online]. Available: https://arxiv.org/abs/2303.14897
[37] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 3, pp. 1623–1637, 2020.
[38] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 12 179–12 188.
[39] H. Yang, D. Huang, W. Yin, C. Shen, H. Liu, X. He, B. Lin, W. Ouyang, and T. He, "Depth any video with scalable synthetic data," *arXiv preprint arXiv:2410.10815*, 2024.