2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Efficient Trajectory-Conditioned Text-to-4D Gaussian Splatting

Lin Shao[1,2], Fan Lu[1*], Haiyun Wei[1], Sanqing Qu[1], Alois Knoll[3], Guang Chen[1,2*]

Abstract— **Recent text-to-4D generation methods have achieved remarkable progress thanks to advances in text-to-video models. Existing approaches typically reconstruct 4D scenes from generated videos or distill them from pre-trained text-to-video models. However, these methods often restrict the scene to a local region or lack spatial controllability. TC4D pioneered trajectory-controllable 4D asset generation by decomposing motion into global transformation and local deformation. While it achieves high visual quality, TC4D suffers from extremely low generation efficiency due to its NeRF-based framework. To overcome this limitation, we propose Efficient TC4DGS, which replaces NeRF with 4D Gaussian Splatting (4DGS) to significantly improve efficiency. Nevertheless, the discrete representation of 4DGS makes optimization challenging, leading to noticeable degradation in visual and motion quality. Thus, we propose a HexPlane-based 4D representation combined with a key-node control scheme. By computing the deformation only for the control nodes and getting overall deformation through interpolation, we greatly improve generation efficiency while maintaining quality. Compared with TC4D, the previous SOTA, we have improved the generation efficiency by 13× (reducing the generation time from 26 hours to 2 hours), while also achieving superior performance in terms of the dynamic quality of the generated objects.**

I. INTRODUCTION

4D object generation has extensive applications in fields such as virtual reality (VR), video games, industrial design, and autonomous driving simulation. With the rapid evolution of diffusion models generating images or videos of extraordinary quality from simple human instructions has become increasingly accessible. Concurrently, methods represented by NeRF [1] and 3DGS [2] have enabled the reconstruction of high-quality 3D assets. Given these synergistic advancements, the time is ripe for advancing the frontiers of 4D object generation.

In the context of trajectory-conditioned generation, while recent methods [3], [4] have achieved considerable progress in generating 2D video under diverse trajectory conditions, current frameworks [5] still struggle to generate complex 4D dynamic objects with precise trajectory control. Commonly, if in-place 4D motion—originally synthesized under a fixed viewpoint—is directly mapped to complex trajectories via rigid spatial transformations, a misalignment between local motion and global displacement inevitably arises. This is often manifested as unnatural motion artifacts, such as the foot-sliding effect. Therefore, trajectory-conditioned 4D object generation represents a more formidable task than traditional 4D generation, carrying significant implications for steering future research in dynamic scene synthesis.

[1] Tongji University, [2] Shanghai Innovation Institute, [3] Technical University of Munich.
[*] Corresponding author

The generation of 4D objects introduces a temporal dimension in addition to spatial dimensions. Consequently, as the complexity of object motion increases, the computational resources required for object generation increase substantially. Using implicit representations (e.g., NeRF [1]) to model 4D objects can achieve high generation quality, but generating a single dynamic object typically requires extensive time for training neural radiance fields [6]. Furthermore, due to the inherent properties of implicit representations, editing and controlling the motion of generated objects is relatively challenging. In contrast, explicit representations, such as 3DGS [2], offer a promising alternative to significantly enhance modeling efficiency. However, high-precision 3DGS objects consist of dense Gaussian primitives, making the optimization of static 3DGS models for dynamic effects highly hardware-dependent.

To address these challenges, we propose a trajectory-conditioned 4D Gaussian generation framework leveraging control point optimization.This approach facilitates the generation of high-precision 4DGS models, achieving an favorable trade-off between visual quality and computational efficiency in 4D object generation. The primary characteristics of our proposed method are illustrated in Fig. 1.

Given a 3DGS model of any size and a motion trajectory of any length, we first decompose the global motion into trajectory motion and local deformation. To solve the difficulty of generating long trajectories, we split the long trajectory into several short segments via decomposition. For each trajectory segment, we enploy a joint-sampling policy to extract key control nodes, whose positions are optimized through a HexPlane-based [7] deformation field during training. Subsequently, the deformation of these control nodes serves as a reference for the overall Gaussian deformation via k-Nearest Neighbor (kNN) interpolation. Finally, we incorporate the trajectory motion into the deformed 4D Gaussian Spaltting model for rendering, and compute the guidance loss distilled from the VideoCrafter [8] decoder to optimize the deformation network. Representative examples are presented in Fig. 2.

In summary, our key contributions are as follows:

- We propose an efficient framework for trajectory-conditioned 4D dynamic Gaussian generation, which successfully handles complex and extended trajectories through trajectory decomposition.
- We introduce a local motion optimization strategy leveraging sparse control nodes and a HexPlane-based deformation field, effectively bridging the gap between discrete 4DGS representations and continuous motion synthesis.

979-8-3315-8160-2/26/$31.00 ©2026 IEEE

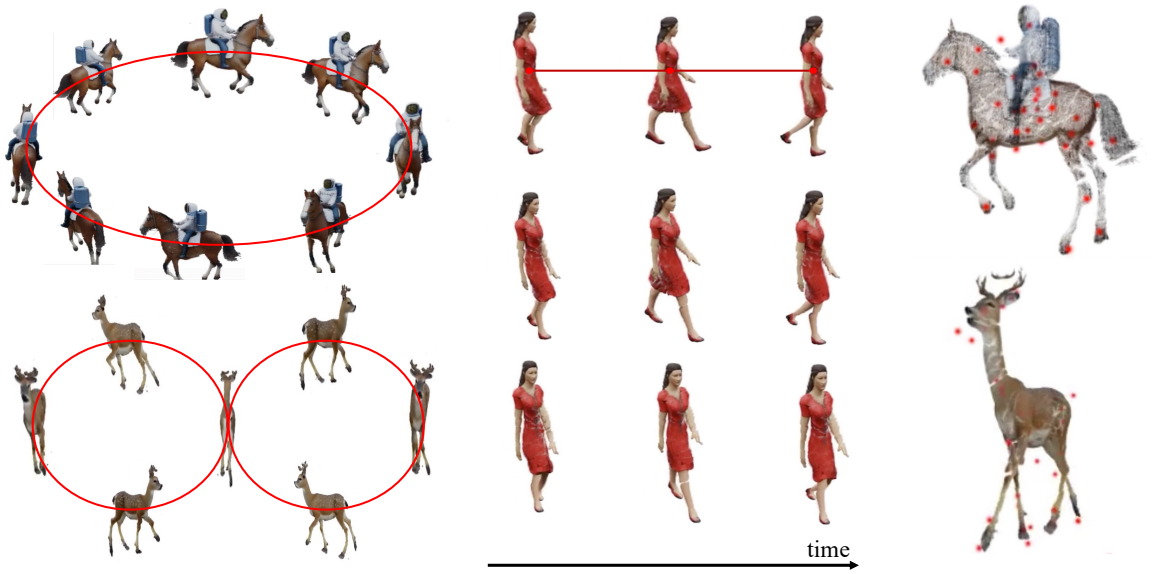

Fig. 1: **Efficient TC4DGS.** Our method efficiently generates high-quality 4D dynamic objects under trajectory control. The main features of our method include strict track following, superior multiview consistency and remarkable computational efficiency achieved through our control-nodes design.

- Extensive experiments demonstrate that our method achieves superior visual fidelity and motion coherence while achieving up to 13× speed-up in training efficiency under identical experimental settings.

II. Related Work

Diffusion-based image and video generation. Diffusion models have emerged as a dominant approach in media generation by modeling semantic information from images through iterative forward noise addition and reverse denoising processes [9]–[11]. Stable Diffusion [10] represents a landmark contribution in diffusion-based generation, employing latent diffusion to compress high-dimensional images into a compact latent space. This approach effectively captures essential image information while significantly reducing computational complexity.

In the domain of video generation, diffusion-based frameworks have similarly witnessed extensive adoption. Stable Video [12] and VideoCrafter [8] advance temporal consistency and visual quality in text-driven video generation by enhancing diffusion-based editing mechanisms and decoupling appearance from motion at the data level, respectively. These methods provide robust temporal priors that are instrumental for downstream 4D generation tasks.

4D generation. The emergence of 3D reconstruction and generation frameworks such as Neural Radiance Fields [1], [13], [14] and 3D Gaussian Splatting [2], [15], [16] has revolutionized the generation of high-quality static 3D assets. However, research focusing on the generation of high-quality 4D objects remains relatively limited.

HexPlane [7] represents a foundational representation for 4D object modeling, establishing a feasible theoretical framework by projecting 4D spatio-temporal volumes onto multiple orthogonal planes. Building upon the concept of explicit dynamics, SCGS [17] proposes a pipeline that optimizes dynamic Gaussians via a sparse set of control points. Specifically, it extracts control points of moving objects from video data and subsequently trains a control point-based deformation network through control point sampling and Gaussian rendering, thereby enabling precise reconstruction of dynamic objects.

We have noted several text-to-video-based 4D generation methods, such as [18], [19], which leverage video diffusion models to synthesize deterministic reference videos. These methods subsequently optimize a 4D deformation field to spatio-temporally align with the generated video priors. DG4D [20] introduces a pipeline for lifting text or images into 4D Gaussian objects. Its core mechanism involves utilizing diffusion-based supervision to regularize the HexPlane-based deformation of static Gaussian models. These approaches have established a robust foundation for subsequent research.

Trajectory-conditioned generation. With the emergence of variants of diffusion models like ControlNet [21], the prospects of trajectory-guided video generation have become highly promising. Nevertheless, current video diffusion models still exhibit significant limitations in generating long-duration sequences, particularly those featuring moving objects under explicit trajectory constraints [22]–[24]. These limitations often prevent them from meeting the requirements of trajectory-conditioned 4D generation tasks. Furthermore, constructing complex multi-object scenes governed by dis-

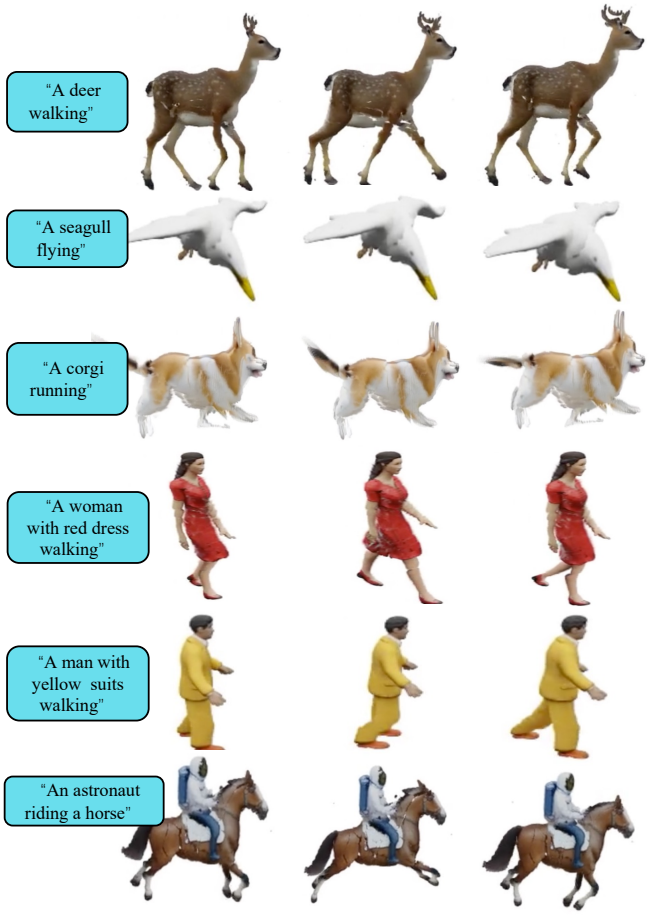

Fig. 2: **Examples.** Our method is capable of generating diverse 4D objects following text prompts.

tinct trajectories remains a formidable challenge. [25]–[27].

TC4D [6], the prior SOTA in this domain, addresses the problem by partitioning global trajectories and employing VideoCrafter as segment-wise supervision. This allows for the optimization of a 4D NeRF representation, facilitating the generation of trajectory-consistent dynamic content. However, inherent to the implicit nature of NeRF, TC4D suffers from relatively slow generation speed, prohibitively high computational overhead, and limited editability.

III. System Design

Our 4D generation pipeline can be divided into two phases: static model generation phase and dynamic deformation generation phase. While the former leverages an open-source framework for initial 3DGS generation, the latter achieves state-of-the-art dynamic trajectory generation performance even under strict VRAM constraints. This is underpinned by the synergistic integration of several key designs: a HexPlane-based deformation field, motion interpolation via sparse control nodes, trajectory decomposition and temporal perturbation sampling. An overview of the structure is illustrated in Fig. 3, while the underlying logic is formalized in Algorithm 1.

A. HexPlane-based deformation field

To endow static models with deformable capabilities, we attach additional deformation field parameters to 3D Gaussian Splatting models.

For the deformation field, we adopt a HexPlane architecture following Dream Gaussian 4D [20], which efficiently represents the 4D spatio-temporal volume by projecting it onto six orthogonal 2D planes. By representing the 4D field as a weighted sum of a set of learnable 4D basis functions, it naturally imposing a smoothness constraint during optimization. We input the 4D coordinates (x, y, z, t) of Gaussians into the HexPlane representation to obtain their grid features, and then derive deformation information, such as positional displacement, rotational changes and scaling changes, via an MLP decoder. Empirical results indicate that for our task, the optimization efficiency of the HexPlane-based deformation field is improved by five times compared to that of a naive MLP-based deformation field, while maintaining superior deformation consistency.

B. Motion interpolation of key control nodes

High-quality 4D Gaussian objects often feature dense pointcloud-like representations. For instance, a single Gaussian model generated by LucidDreamer may comprise upward of one million Gaussian primitives. Under such conditions, the direct optimization of high-dimensional Gaussian deformation fields via diffusion priors remains computationally prohibitive and prone to optimization instability.

Since the dynamics of complex 3D entities generally adhere to underlying structural priors—such as articulated skeletal constraints—rather than exhibiting stochastic behavior, the unconstrained optimization of a global Gaussian deformation field may prove both computationally inefficient and parametrically redundant. Furthermore, due to the instability of diffusion loss, when high-quality Gaussian objects possess intricate texture features (e.g., spots and stripes), directly optimizing global Gaussian deformations may lead to severe texture distortion.

Thus, we initialize the deformation fields only for a sparse set of key control nodes. By optimizing the deformation fields of control nodes and then using the kNN algorithm to interpolate and control global Gaussians, we address the challenge of optimizing large scale global Gaussians. Meanwhile, since global Gaussians within the same spatial proximity are likely to be controlled by identical Gaussian control points, thereby exhibiting similar deformation characteristics, our design significantly mitigates texture distortion.

Eq. (1) indicates the specific calculation process of the deformation of Gaussians, where d represents the deformation of an arbitrary Gaussian point (x, y, z) at time t. p_i denotes the i-th nearest control node identified via kNN search, and $deform(p_i, t)$ gives the deformation of p_i at time t, computed using the HexPlane-based deformation field mentioned above, and r_i gives the distance between p_i and the Gaussian point. We assume that the smaller the r-value, the greater the influence of the control node on the Gaussian point. Thus, via an exponential operation with base

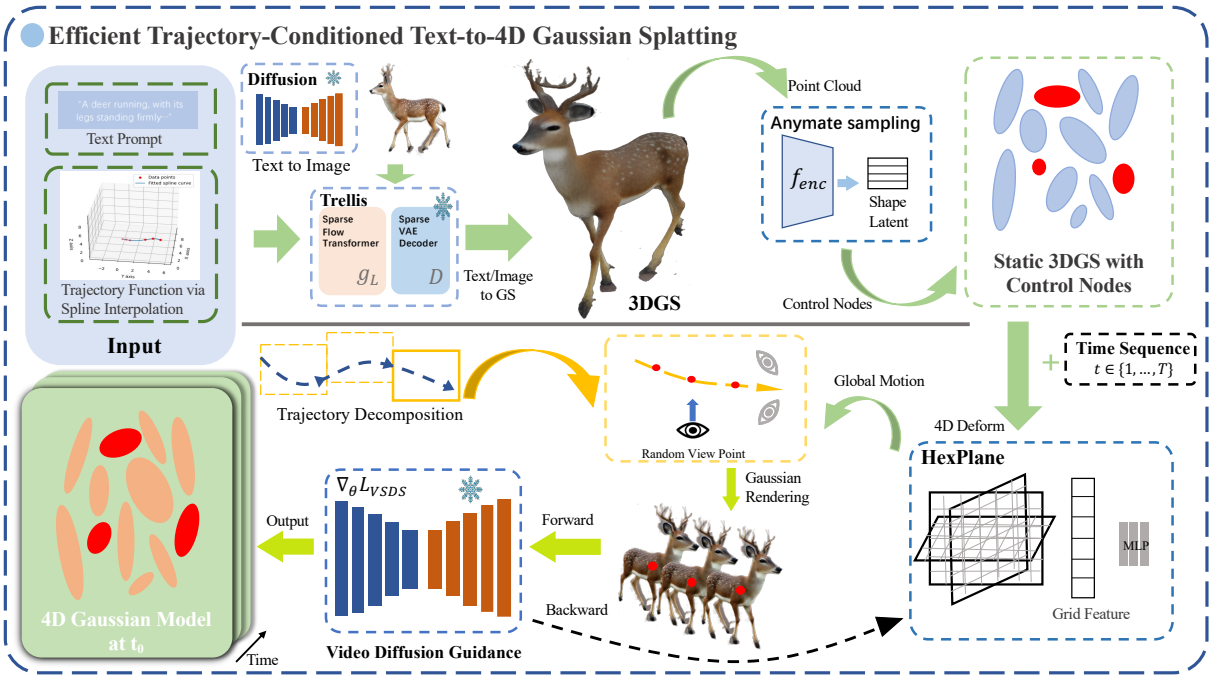

Fig. 3: **Overall structure of Efficient TC4DGS.** We take a text prompt and a trajectory as input, utilizing Trellis [28] as the core 3DGS generation engine. Then, Anymate sampling is employed to extract control nodes from the static Gaussian point cloud. We apply a HexPlane-based deformation network to get the 4D representation. After rendering videos from the 4DGS model on the decomposed trajectory, we use VideoCrafter to optimize the deformation field through VSDS guidance.

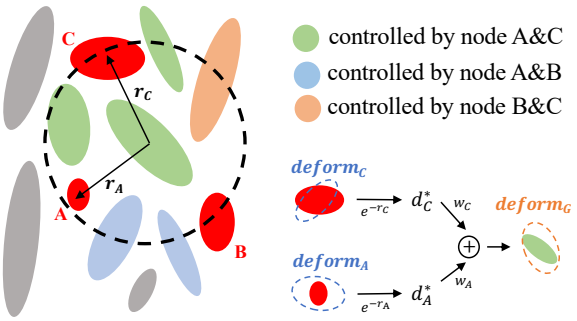

Fig. 4: **Node-controlled motion interpolation.** The deformation of dense Gaussians is computed by aggregating the deformation of k nearest control nodes with distance bias e^{-r_i} and weight bias w_i.

e, we scale the numerical range of the r-value's influence on the Gaussian point to the interval $[0, 1]$. Furthermore, we use w_i as a learnable weight to dynamically adjust the control strength of each control node. Fig. 4 illustrates the process of computing Gaussian deformation via control node interpolation at $k = 2$.

$$d(x, y, z, t) = \sum_{i=0}^{k} (w_i * \mathbf{e}^{-r_i} * deform(p_i, t)). \quad (1)$$

Specifically, for control node initialization, we leverage Anymate [29], a framework designed for 3D model rigging that utilizes a fine-tuned Point-BERT [30] architecture. This approach has demonstrated outstanding performance across multiple tests. In our dynamic generation task, the Anymate Sampling strategy significantly outperforms the baseline Farthest Point Sampling (FPS) approach. Consequently, we integrate the Anymate framework for automated rigging, utilizing the derived joint coordinates as the initial control points for our deformation field.

C. Trajectory decomposition

As noted in TC4D, we face two primary challenges in generating motions for complex and long trajectories, namely: 1) Long trajectories may exceed the optimization length constraints of video diffusion models; 2) Basic video diffusion models have limited capability to optimize video of moving objects with complex trajectories.

To address these two challenges, we decouple global motion into local motion within the object's motion bounding box and global displacement outside this bounding box. For global displacement, we implement a chunk-based optimization strategy with overlapping segments to ensure temporal continuity. Specifically, we derive a smooth trajectory function via spline interpolation based on the key points of the given trajectory. For each timestamp, we sample on this spline function to acquire the center coordinates of the 4D object. Within each chunk, we use the U-Net decoder of the video diffusion model to denoise each video clip, thereby achieving optimization of local motion without generating reference videos. To further enforce multi-view consistency, we perform random azimuthal rotations on the

camera position for Gaussian rendering at each training iteration.

In Eq. (2), M represents the global pose of the model at time t. T denotes the global translation at time t given by the spline function; R is the rotation matrix transforming $\hat{p}$ to T', where $\hat{p}$ denotes the frontal orientation of the static model at $t = 0$ and $T' = \mathrm{d}T/\mathrm{d}t$ denotes the tangent vector of the spline trajectory. In the equation, $\mathbf{p}$ represents all coordinates of the static Gaussian model.

$$\mathrm{M}(\mathbf{p}, \mathrm{t}) = \mathrm{T}(t) + \mathrm{R}(\hat{p}, T')\mathbf{p}. \tag{2}$$

D. Temporal perturbation sampling

We observe that two parameters directly influence the optimization results when using video diffusion models to optimize video motion, namely the motion timespan and the time interval between adjacent frames.

While an extended motion duration facilitates the optimization of continuity and global consistency, a higher temporal resolution—characterized by smaller inter-frame intervals—permits the diffusion model to execute fine-grained adjustments, thereby mitigating local motion artifacts. Since both requirements demand greater VRAM support, GPU devices frequently face the challenge of balancing motion timespan and frame interval, as it is difficult to optimize both parameters simultaneously.

We propose a training strategy based on temporal perturbation sampling. During two stages of training, we applied a small-range random perturbation to the motion timestamps. By introducing these random perturbations to the temporal inputs of the deformation field throughout the various optimization phases, we effectively regularize the model. This approach bolsters the deformation field's temporal robustness and enhances its generalization to dynamic variations, ensuring smoother transitions and preventing the model from overfitting to discrete timestamps.

Through this sampling strategy, we achieve significantly more fine-grained motion optimization. Furthermore, this approach enables the synthesis of 4D Gaussian objects with arbitrarily high frame rates. By querying the learned deformation field at arbitrary temporal offsets, our method facilitates seamless interpolation and stable rendering.

E. Loss design

We employ the VSDS loss provided by a video diffusion model as the primary supervisory signal, facilitating stable and consistent convergence of the deformation field.

Specifically, given a random viewpoint o, a video sequence $\mathbf{v}_o$ is rendered from the 4D Gaussian model parameterized with θ. Then, depending on the diffusion timestamp t, a random noise ϵ is added to the video.The precise definition of VSDS loss is given by Eq. (3).

$$\nabla_\theta L_{\text{VSDS}} = \mathbb{E}_{t,\epsilon,o}[(\hat{\epsilon}(\mathbf{v}_{t,o}, t, \mathbf{y}, o) - \epsilon)\frac{\partial \mathbf{v}_o}{\partial \theta}]. \tag{3}$$

The expectation $\mathbb{E}$ is calculated by all timestamps, noise and viewpoint. $\hat{\epsilon}$ is the noise predicted by the video diffusion model based on the diffusion timestamp t, the text embedding

Algorithm 1 Efficient TC4DGS

Require:
- $\mathbf{P}_{\text{3D}}$ ▷ Static 3DGS representations
- $\mathbf{N}$ ▷ Control Nodes initialized with Anymate
- T ▷ Global trajectory parameterized with a spline
- D ▷ Initial deformation field parameterized with θ

Output:
- D^* ▷ Optimized HexPlane-based deformation field

1: Sample p_i and $p_{i+\delta t}$ from T
2: Apply rigid transform by Eq. (2)
3: Calculate local deformation $\mathbf{P}^*_{\text{4D}}$ by two steps
   - Apply D to control nodes $\mathbf{N}$ so that we can get the deformation of nodes $deform_{\mathbf{N}}$
   - Calculate deformation of $\mathbf{P}_{\text{3D}}$ by Eq. (1) therefore get $\mathbf{P}^*_{\text{4D}}$
4: Combine rigid tansform and local deformation to get global transformation $\mathbf{P}_{\text{4D}}$
5: Render video from $\mathbf{P}_{\text{4D}}$ with differential Gaussian rasterization
6: Calculate $\nabla_\theta L_{\text{4D}}$ and $\nabla_\theta L_{\text{quality}}$ by Eq. (3) and Eq. (4)
7: Update D
8: Repeat steps 1-7

$\mathbf{y}$ and a random viewpoint o, while $\frac{\partial \mathbf{v}_o}{\partial \theta}$ stands for the gradient of the rendered video $\mathbf{v}_o$ with regard to θ.

Furthermore, we use auxiliary losses including track loss, ARAP loss, and SSIM loss jointly to help optimize the deformation field as Eq. (4), Eq. (5), Eq. (6), and Eq. (7).

$$\nabla_\theta L_{\text{assist}} = \lambda_{track} L_{\text{track}} + \lambda_{arap} L_{\text{ARAP}} + \lambda_{ssim} L_{\text{SSIM}}, \tag{4}$$

$$L_{\text{track}} = ||\mathbf{P}^*_{\text{4D}}||_1 = \sum_{i=1}^{n} (|\Delta x_i| + |\Delta y_i| + |\Delta z_i|), \tag{5}$$

$$L_{\text{ARAP}} = \sum_i \sum_{j \in \mathcal{N}(i)} ||(\mathbf{p}_i^{'} - \mathbf{p}_j^{'}) - \mathbf{R}_i(\mathbf{p}_i - \mathbf{p}_j)||_2^2, \tag{6}$$

$$L_{\text{SSIM}} = 1 - \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}. \tag{7}$$

The track loss is designed to regularize the motion of non-critical Gaussians by quantifying the spatial drift between their canonical coordinates and their deformed positions at a specific time step. Specifically, Eq. (5) defines this loss as the L_1 penalty applied to the spatial coordinate offsets, thereby promoting sparsity in unnecessary displacements.

ARAP loss [17] alleviates excessive local deformation distortion during deformation optimization by constraining displacement changes between control points while allowing rotational changes. As shown in Eq. (6), $\mathcal{N}(i)$ denotes the set of the k nearest neighbors of node i. $\mathbf{p}_i$ and $\mathbf{p}_j$ are the original coordinates of nodes i and j, respectively, while

$\mathbf{p}'_i$ and $\mathbf{p}'_j$ denote the corresponding deformed coordinates. The term $\mathbf{R}_i$ denotes the optimal local rotation matrix associated with node i. This formulation ensures that the local neighborhood undergoes a transformation that is as close to a rigid body motion as possible.

SSIM loss calculates structural differences between images, thereby reducing local texture distortion and unnecessary deformation. In Eq. (7), x,y refers to the rendered image of 4D Gaussians for each frame and that of static Gaussians

IV. Experimental Results

We conducted a series of experiments to verify the effectiveness of our core method, including ablation experiments on each core component, comparative experiments with the previous SOTA method TC4D and human evaluations. Among these, quantitative metrics were basically provided by VideoScore [31], a video quality assessment approach based on multimodal large language models. The overall results can be seen from Table I. As detailed in Table I, we evaluate the generated videos across three primary dimensions. Visual Quality assesses the overall perceptual fidelity of the video, which is heavily dependent on texture details and spatial resolution. Dynamic Degree evaluates the naturalness and fluidity of temporal movements and structural deformations. Finally, text-to-video Alignment measures the semantic consistency between the generated visual content and the conditioning text prompt.

All our method was trained under 24GB VRAM unless otherwise stated, as higher VRAM provide relatively minor advantage. Ours* demonstrates the performance of our method while training for as long as TC4D.

A. Ablation study

First, we conduct comprehensive ablation studies to validate the effectiveness of our combined loss formulation. Fig. 5 presents qualitative comparisons illustrating the visual impact of each key component.

In experiments, when the track loss was set too small or not used at all, it led to a corner case where the Gaussian model completely deviated from the camera's rendering perspective during dynamic optimization, thereby causing optimization failure. When λ_{track} was set above a threshold related to the model scale, trajectory deviation issues can be largely avoided. Further increasing λ_{track} can almost completely eliminate trajectory deviation. However, this came at the cost of reduced motion quality. Empirically, setting the weight λ_{track} to 1000 strikes an optimal balance between tracking fidelity and dynamic realism.

As demonstrated in Table I, applying an appropriate ARAP loss effectively regularizes local deformations, yielding noticeable improvements in both temporal consistency and structural plausibility. Conversely, an excessively high ARAP penalty over-constrains the model, thereby degrading its overall dynamic realism. In practice, we set the weighting factor λ_{arap} within the range of $[100, 1000]$, depending on the specific object category.

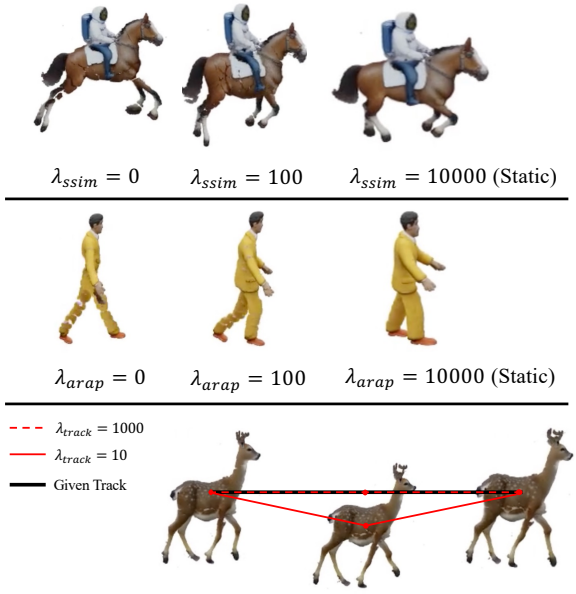

Fig. 5: **Ablation study on main loss design.** The SSIM loss reduces overall motion distortion and texture loss. The ARAP loss enhances the ability to control exaggerated tensile deformation while maintaining a proper amount of rotational deformation. The track loss keeps the object close to the given trajectory.

Qualitatively, an appropriate configuration of SSIM loss can reduce unnecessary and unnatural deformation of the model. Quantitatively, SSIM loss can improve visual quality and text-to-video alignment. Similarly, an excessively large SSIM loss configuration causes the model to degrade dynamic quality, making it tend to be static overall. We find an appropriate λ_{ssim} to be among 100 to 1000.

B. Comparative experiments with TC4D

We conducted quantitative comparisons with TC4D across 10 use cases under both low and high VRAM conditions. The experiments were performed under two distinct hardware configurations representing high and low VRAM regimes: an NVIDIA H20 GPU (96GB VRAM) and an NVIDIA RTX 3090 GPU (24GB VRAM).

Our results were all trained with the RTX 3090 GPU since we found that our method performed similarly under low and high VRAM. Fig. 6 indicated that we achieved significantly higher dynamic quality than TC4D under both high and low VRAM settings. This outcome underscores the efficiency of our approach, effectively liberating long-duration 4D video generation from its traditional dependence on high-end industrial GPUs. By decoupling sequence length from memory overhead, our method facilitates high-fidelity 4D synthesis on consumer-grade hardware without compromising temporal coherence or motion complexity.

To minimize the impact of static models on evaluations, we used Trellis's Image-to-3D method to reconstruct the models generated in TC4D's Stage 2. However, due to the limitations of the Image-to-3D method, the static models we generated inevitably lag behind TC4D to some extent

TABLE I: **Quantitative Results by VideoScore and Human.** Ours* demonstrates the performance of our method when training for as long as TC4D.

Method	Visual Quality	Dynamic Degree	Text-to-Video Alignment	Time
Ours (RTX 3090)	2.438	**3.039**	**2.563**	**2h**
Ours*(RTX 3090)	2.422	**3.047**	**2.578**	25h
TC4D (RTX 3090)	2.547	2.821	2.453	26h
TC4D (H20)	2.500	2.852	2.453	26h
Ours	2.531	3.016	2.719	–
w/o track loss (fail)	–	–	–	–
w/o ARAP loss	2.516	3.031	2.703	–
w/o SSIM loss	2.312	3.031	2.391	–
Human test (Ours)	3.00	**2.33**	2.89	**2h**
Human test (Ours*)	2.67	**4.33**	**4.00**	25h
Human test (TC4D)	4.56	2.44	3.33	26h

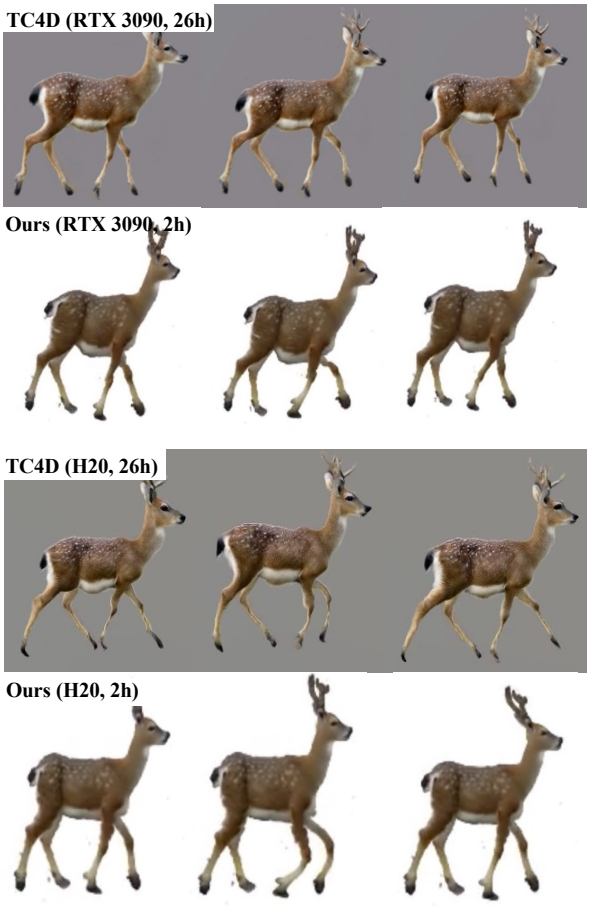

Fig. 6: **Comparative experiments with TC4D.** We achieve similar visual quality and superior dynamic with significantly higher efficiency.

in terms of visual quality, as is shown in Table I.

C. Human evaluation

To further validate the subjective quality of our results, we conducted an extensive human evaluation involving both peer researchers and diverse non-expert volunteers. The video shown to the volunteers was generated with RTX 3090. As Table I shows, it is obvious that we achieved similar visual quality and dynamic quality with much lower cost of VRAM, which basically matches the result of VideoScore. The marginal discrepancy between VideoScore and human ratings likely stems from inherent variances in subjective perception, particularly differing sensitivities to temporal artifacts versus static textural details. Notably, when the training duration of our framework is extended to match that of TC4D (Ours*), we achieve significantly superior performance across more qualitative dimensions, underscoring the efficiency and scalability of our optimization strategy.

V. Conclusion and Future Work

We propose a pipeline for generating high-fidelity 4D Gaussian objects using text, image, and trajectory information as inputs. By means of trajectory decomposition, HexPlane-based deformation field design, motion interpolation of control nodes, along with temporal perturbation sampling, we have achieved state-of-the-art generation efficiency and motion performance.

Future work can be carried out from the following perspectives. First, to ensure the rigor of comparative experiments, we employed VideoCrafter as the baseline video generation model for supervision following TC4D. Replacing the video supervision model with improved frameworks such as [32], [33] is expected to further enhance the performance of dynamic optimization. Second, although the control points sampled by Anymate exhibit less redundancy compared to those from FPS sampling, thus providing better control, our experiments reveal that the Anymate initialization method still introduces a substantial number of redundant control points unrelated to motion. This may constitute a bottleneck for further improving motion quality. Future work could explore the incorporation of real-time pruning and splitting of control points during training. Lastly, our work focuses on generating moving objects that adhere to trajectory specifications and does not address background modeling [34]–[36]. Integrating trajectory-conditioned object generation into 3D environments in a semantically coherent manner presents a challenging yet promising direction for future research.

ACKNOWLEDGMENT

This work was supported by the National Key Research and Development Program of China (No.

2024YFE0211000), in part by the National Natural Science Foundation of China (No. 62372329, 62506263, 62506264), in part by the Shanghai Scientific Innovation Foundation (No. 23DZ1203400), in part by the China Postdoctoral Science Foundation (No. BX20250383, GZB20250385, 2025M771530, 2025M771539), in part by the Open Found of the Engineering Research Center of Intelligent Swarm Systems, Ministry of Education (ZZU-CISS-2024001), in part by Tongji-Qomolo Autonomous Driving Commercial Vehicle Joint Lab Project, and in part by Xiaomi Young Talents Program.

REFERENCES

[1] 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," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[2] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering." *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139–1, 2023.

[3] J. Wu, X. Li, Y. Zeng, J. Zhang, Q. Zhou, Y. Li, Y. Tong, and K. Chen, "Motionbooth: Motion-aware customized text-to-video generation," *Advances in Neural Information Processing Systems*, vol. 37, pp. 34 322–34 348, 2024.

[4] X. Shi, Z. Huang, F.-Y. Wang, W. Bian, D. Li, Y. Zhang, M. Zhang, K. C. Cheung, S. See, H. Qin *et al.*, "Motion-i2v: Consistent and controllable image-to-video generation with explicit motion modeling," in *ACM SIGGRAPH 2024 Conference Papers*, 2024, pp. 1–11.

[5] Y. Zhao, Z. Yan, E. Xie, L. Hong, Z. Li, and G. H. Lee, "Animate124: Animating one image to 4d dynamic scene," *arXiv preprint arXiv:2311.14603*, 2023.

[6] S. Bahmani, X. Liu, W. Yifan, I. Skorokhodov, V. Rong, Z. Liu, X. Liu, J. J. Park, S. Tulyakov, G. Wetzstein *et al.*, "Tc4d: Trajectory-conditioned text-to-4d generation," in *European Conference on Computer Vision*. Springer, 2024, pp. 53–72.

[7] A. Cao and J. Johnson, "Hexplane: A fast representation for dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 130–141.

[8] H. Chen, Y. Zhang, X. Cun, M. Xia, X. Wang, C. Weng, and Y. Shan, "Videocrafter2: Overcoming data limitations for high-quality video diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 7310–7320.

[9] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*, vol. 1, no. 2, p. 3, 2022.

[10] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[11] X. Wang, T. Darrell, S. S. Rambhatla, R. Girdhar, and I. Misra, "Instancediffusion: Instance-level control for image generation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 6232–6242.

[12] W. Chai, X. Guo, G. Wang, and Y. Lu, "Stablevideo: Text-driven consistency-aware diffusion video editing," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 23 040–23 050.

[13] W. Hu, Y. Wang, L. Ma, B. Yang, L. Gao, X. Liu, and Y. Ma, "Tri-miprf: Tri-mip representation for efficient anti-aliasing neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 19 774–19 783.

[14] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM transactions on graphics (TOG)*, vol. 41, no. 4, pp. 1–15, 2022.

[15] Z. Wang, C. Lu, Y. Wang, F. Bao, C. Li, H. Su, and J. Zhu, "Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation," *Advances in neural information processing systems*, vol. 36, pp. 8406–8441, 2023.

[16] J. Tang, Z. Chen, X. Chen, T. Wang, G. Zeng, and Z. Liu, "Lgm: Large multi-view gaussian model for high-resolution 3d content creation," in *European Conference on Computer Vision*. Springer, 2024, pp. 1–18.

[17] Y.-H. Huang, Y.-T. Sun, Z. Yang, X. Lyu, Y.-P. Cao, and X. Qi, "Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 4220–4230.

[18] S. Bahmani, I. Skorokhodov, V. Rong, G. Wetzstein, L. Guibas, P. Wonka, S. Tulyakov, J. J. Park, A. Tagliasacchi, and D. B. Lindell, "4d-fy: Text-to-4d generation using hybrid score distillation sampling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 7996–8006.

[19] H. Zhang, X. Chen, Y. Wang, X. Liu, Y. Wang, and Y. Qiao, "4diffusion: Multi-view video diffusion model for 4d generation," *Advances in Neural Information Processing Systems*, vol. 37, pp. 15 272–15 295, 2024.

[20] J. Ren, L. Pan, J. Tang, C. Zhang, A. Cao, G. Zeng, and Z. Liu, "Dreamgaussian4d: Generative 4d gaussian splatting," *arXiv preprint arXiv:2312.17142*, 2023.

[21] S. Zhao, D. Chen, Y.-C. Chen, J. Bao, S. Hao, L. Yuan, and K.-Y. K. Wong, "Uni-controlnet: All-in-one control to text-to-image diffusion models," *Advances in Neural Information Processing Systems*, vol. 36, pp. 11 127–11 150, 2023.

[22] Z. Hao, X. Huang, and S. Belongie, "Controllable video generation with sparse trajectories," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7854–7863.

[23] Z. Liu, A. Yanev, A. Mahmood, I. Nikolov, S. Motamed, W.-S. Zheng, X. Wang, L. Van Gool, and D. P. Paudel, "Intragen: Trajectory-controlled video generation for object interactions," *arXiv preprint arXiv:2411.16804*, 2024.

[24] M. Niu, X. Cun, X. Wang, Y. Zhang, Y. Shan, and Y. Zheng, "Mofa-video: Controllable image animation via generative motion field adaptions in frozen image-to-video diffusion model," in *European Conference on Computer Vision*. Springer, 2024, pp. 111–128.

[25] Y. Ma, K. Feng, Z. Hu, X. Wang, Y. Wang, M. Zheng, X. He, C. Zhu, H. Liu, Y. He *et al.*, "Controllable video generation: A survey," *arXiv preprint arXiv:2507.16869*, 2025.

[26] Z. Kuang, S. Cai, H. He, Y. Xu, H. Li, L. J. Guibas, and G. Wetzstein, "Collaborative video diffusion: Consistent multi-video generation with camera control," *Advances in Neural Information Processing Systems*, vol. 37, pp. 16 240–16 271, 2024.

[27] H. Qiu, Z. Chen, Z. Wang, Y. He, M. Xia, and Z. Liu, "Freetraj: Tuning-free trajectory control in video diffusion models," *arXiv preprint arXiv:2406.16863*, 2024.

[28] J. Xiang, Z. Lv, S. Xu, Y. Deng, R. Wang, B. Zhang, D. Chen, X. Tong, and J. Yang, "Structured 3d latents for scalable and versatile 3d generation," 2025. [Online]. Available: https://arxiv.org/abs/2412.01506

[29] Y. Deng, Y. Zhang, C. Geng, S. Wu, and J. Wu, "Anymate: A dataset and baselines for learning 3d object rigging," in *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, 2025, pp. 1–10.

[30] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu, "Point-bert: Pre-training 3d point cloud transformers with masked point modeling," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 19 313–19 322.

[31] X. He, D. Jiang, G. Zhang, M. Ku, A. Soni, S. Siu, H. Chen, A. Chandra, Z. Jiang, A. Arulraj *et al.*, "Videoscore: Building automatic metrics to simulate fine-grained human feedback for video generation," *arXiv preprint arXiv:2406.15252*, 2024.

[32] Z. Yang, J. Teng, W. Zheng, M. Ding, S. Huang, J. Xu, Y. Yang, W. Hong, X. Zhang, G. Feng, D. Yin, Y. Zhang, W. Wang, Y. Cheng, B. Xu, X. Gu, Y. Dong, and J. Tang, "Cogvideox: Text-to-video diffusion models with an expert transformer," 2025. [Online]. Available: https://arxiv.org/abs/2408.06072

[33] Z. Zheng, X. Peng, T. Yang, C. Shen, S. Li, H. Liu, Y. Zhou, T. Li, and Y. You, "Open-sora: Democratizing efficient video production for all," 2024. [Online]. Available: https://arxiv.org/abs/2412.20404

[34] J. Chung, S. Lee, H. Nam, J. Lee, and K. M. Lee, "Luciddreamer: Domain-free generation of 3d gaussian splatting scenes," *arXiv preprint arXiv:2311.13384*, 2023.

[35] S. Zhou, Z. Fan, D. Xu, H. Chang, P. Chari, T. Bharadwaj, S. You, Z. Wang, and A. Kadambi, "Dreamscene360: Unconstrained text-to-3d scene generation with panoramic gaussian splatting," in *European Conference on Computer Vision*. Springer, 2024, pp. 324–342.

[36] W. Li, F. Cai, Y. Mi, Z. Yang, W. Zuo, X. Wang, and X. Fan, "Scenedreamer360: Text-driven 3d-consistent scene generation with panoramic gaussian splatting," *arXiv preprint arXiv:2408.13711*, 2024.