PromptVFX: Text-Driven Fields for Open-World 3D Gaussian Animation

Mert Kiray*,1,2,3 Paul Uhlenbruck*,1 Nassir Navab1,2 Benjamin Busam1,2,3

[1]Technical University of Munich [2]Munich Center for Machine Learning (MCML) [3]Obsphera

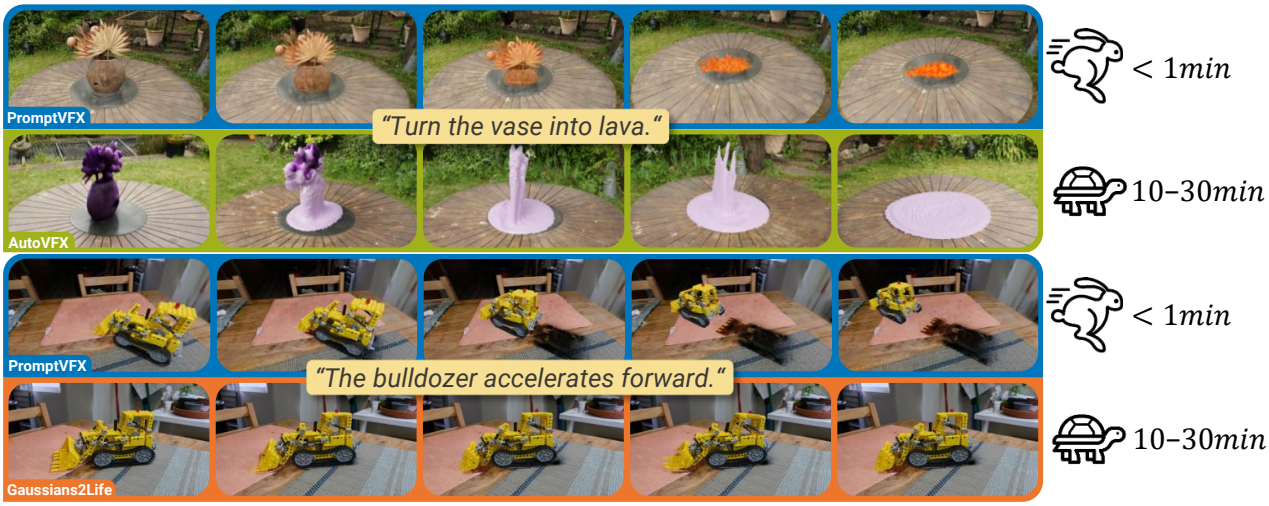

Figure 1. **Comparison of PromptVFX with existing text-driven 3D animation methods.** Our approach generates high-quality animations in seconds, whereas others require 30× more time due to diffusion or physics-based simulation. Exact prompts are provided in the supplementary material.

Abstract

Visual effects (VFX) are key to immersion in modern films, games, and AR/VR. Creating 3D effects requires specialized expertise and training in 3D animation software and can be time consuming. Generative solutions typically rely on computationally intense methods such as diffusion models which can be slow at 4D inference. We reformulate 3D animation as a field prediction task and introduce a text-driven framework that infers a time-varying 4D flow field acting on 3D Gaussians. By leveraging large language models (LLMs) and vision-language models (VLMs) for function generation, our approach interprets arbitrary prompts (e.g., "make the vase glow orange, then explode") and instantly updates color, opacity, and positions of 3D Gaussians in real time. This design avoids overheads such as mesh extraction, manual or physics-based simulations and allows both novice and expert users to animate volumetric scenes with minimal effort on a consumer device even in a web browser [1]. Experimental results show that simple textual instructions suffice to generate compelling time-varying VFX, reducing the manual effort typically required for rigging or advanced modeling. We thus present a fast and accessible pathway to language-driven 3D content creation that can pave the way to democratize VFX further. Code available at https://obsphera.github.io/promptvfx/.

1. Introduction

"Today, with computer-generated visual effects, everything is possible. So we've seen everything. If it can be imagined, it can be put on screen."[2] The creation of 3D animation is omnipresent, yet it requires specialized software and expert knowledge like the ones from visual effect studios Campisi typically works with for his Hollywood productions.

VFX artists rely on offline physical simulation software for precise motion and collision dynamics which are computationally expensive and prohibitively slow on consumer hardware [10]. Recent trends in generative AI [42, 44] make

*Authors contributed equally to this work.

[1]Web Demo: http://promptvfx.duckdns.org

[2]Gabriel Campisi. Producer, Screenwriter, Director, and Author.

Table 1. **Comparison of visual editing methods.** The table shows existing methods sorted by features regarding input/output, interfaces, effect capacities, and system properties. PromptVFX (Ours) is the only method that simultaneously fullfills all features while allowing for interactive user refinement.

	Input & Output						Editing Capacities				Properties			
Method	Real World Video Editing	Free-Viewpoint Rendering	Editing Interface	Viewpoint Consistent	Continuous Time Rep.	Open-world Query	Interactive Refinement	Appearance Change	Object Animations	Particle Effects	Zero-Shot	No Complex Preprocessing	Diffusion Model Free	Animation Software Free
Visual Programming [14]	✓		Language			✓		✓			✓			
FRESCO [58]	✓		Language			✓		✓			✓	✓		✓
ClimateNeRF [26]	✓	✓	Scripts	✓				✓		✓		✓		
GaussianEditor [9]	✓	✓	GUI	✓		✓		✓				✓		✓
Gaussian Grouping [60]	✓	✓	GUI	✓		✓		✓						✓
PhysGaussian [53]	✓	✓	GUI	✓	✓		✓						✓	
VR-GS [19]	✓	✓	GUI	✓	✓		✓							
Gaussian Splashing [13]	✓	✓	GUI	✓						✓			✓	
DMRF [37]	✓	✓	GUI	✓					✓	✓		✓	✓	✓
Instruct-N2N [15]	✓	✓	Language	✓		✓		✓						✓
DGE [8]	✓	✓	Language	✓		✓		✓						✓
Chat-Edit-3D [12]	✓	✓	Language	✓		✓	✓	✓				✓		✓
Gaussians2Life [50]	✓	✓	Language	✓		✓		✓	✓			✓		✓
DreamGaussians4D [41]	✓	✓	Language	✓	✓	✓		✓	✓			✓		✓
AutoVFX [18]	✓	✓	Language	✓	✓	✓		✓	✓	✓			✓	
PromptVFX (Ours)	✓	✓	Language	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

the creative process more accessible to a wider audience. The pipelines typically employ large diffusion models to generate or edit 3D scenes with text prompts [35, 50]. Computation cost and multi-view inconsistencies are obstacles on the way to realistic real-time interaction [43, 57].

To truly democratize 3D animation, a fast and intuitive system that permits open-ended text instructions with instant feedback is required. To this end, promising frameworks [15, 18] have been created that integrate large (vision) language models (L(V)LMs) to parse user prompts. However, they equally depend on either physics-based simulations or diffusion training loops, preventing real-time editing. So far, there is no solution established that accepts any textual command (e.g., "rotate the statue, then change its color to gold in three seconds") that immediately updates a 3D scene without offline simulation or optimization.

In this paper, we propose **PromptVFX**, a text-driven framework for 3D animation that provides these features. Centered on Gaussian Splatting [22], our system translates high-level textual instructions into time-varying transformations on a set of Gaussians without generating new geometry or launching physics simulators. A time-varying field is applied to a Gaussian splatting scene ultimately changing centre positions, colour and opacity of individual Gaussians. When instructed to "raise the vase by two meters," the system directly modifies the Gaussian centre positions over time. This simple yet generic idea naturally supports real-time rendering such that the effect of a user prompt can be visualized within seconds rather than minutes or hours.

A key enabler of our approach is its open-text interface, powered by an LLM and optionally a LVLM for appearance-grounding. In contrast to methods that restrict users to a narrow command set or domain-specific scripts, we allow any natural language request ("make the tree shake, then fade to transparent"). The LLM parses these instructions into text-driven fields that change the position, color and opacitiy of scene parts. By pairing VFX-transformations with a robust foundation model that has been trained on a large body of data, we can capitalize on the fact that "we've seen everything" (Campisi) and create generic cinematic or stylized effects within a single pipeline without training new models or worrying about 3D consistency of hallucinated content. Our contributions can be summarized as follows:

- **Open world VFX with text driven fields.** We recast 3D animation as time varying transformations on Gaussian splats to enable volumetric effects without retraining or simulation.
- **Training-free 4D animations.** Our zero-shot workflow breaks natural language instructions into animation phases, generates transformation functions via LLM and refines them using visual language model feedback with no per scene training.
- **Real-time interactive editing.** Updating Gaussian parameters directly removes mesh extraction, diffusion loops or simulators. Animations are delivered in under one minute on a single GPU or in a browser allowing for instant user feedback.

We demonstrate our method on various scenes and user prompts, showcasing dynamic, visually consistent animations generated within seconds of receiving text instructions in Figure 1. Beyond accelerating the workflow for experienced artists, this real-time, open-text interface makes 3D animation accessible to a significantly broader audience. We strongly believe that our work can help unify intuitive language interfaces with efficient 3D animation.

2. Related Work

PromptVFX builds upon recent advancements in generative models for text-driven content creation, 4D content generation, 3D editing, physics-based VFX, and interactive interfaces for scene manipulation. We review key contributions in these areas and position our work in relation to them. Table 1 summarizes key features of related editing tools.

Generative text-to-pixel models. Early generative models such as Denoising Diffusion Probabilistic Models

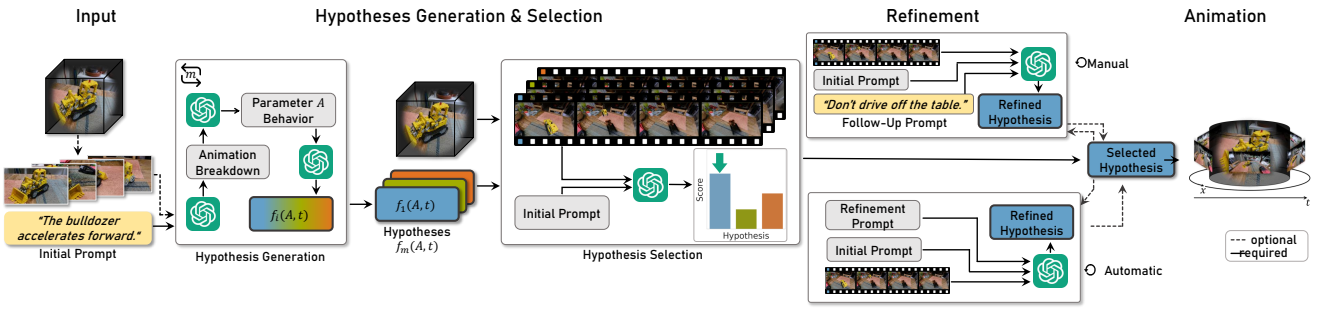

Figure 2. **Overview of the PromptVFX pipeline.** Given a user-provided textual prompt, the system first decomposes it into structured animation phases. A large language model (LLM) then generates parametric functions that define the motion, color, and opacity changes of 3D Gaussians over time. To handle ambiguity, multiple animation hypotheses are generated and evaluated using a vision-language model (VLM) or user feedback. The selected animation is further refined through automatic and interactive text-based corrections, ensuring high-quality, real-time results. More details on prompt formulation and scoring are provided in the supplementary material.

(DDPM) [17] and latent diffusion models [42] have demonstrated remarkable success in text-to-image generation. These approaches have been extended to video generation through text-to-video diffusion models [5, 6, 45], which generate temporally coherent image sequences but remain computationally expensive. Diffusion models have also been adapted to synthesize images from different viewpoints of a scene [3, 16]. However, enforcing multi-view consistency is not straightforward and generated results can change appearance under varied 3D viewpoint. Several works attempt to enforce view consistency through explicit constraints [25, 35, 54].

Dynamic 3D generation. The representation of 3D Gaussian splatting [22] has emerged as an efficient alternative to implicit neural representations like NeRF [33]. Recent works extend Gaussian Splatting to dynamic 4D content [11, 27, 32, 51], enabling animated reconstructions with controlled deformations [21, 36]. However, dynamic 3D content generation still lags behind video-based approaches in realism, typically focusing on isolated foreground or single objects [2, 29, 46]. Current approaches can require multiple hours of optimization for a few seconds of scene content [61].

Diffusion-based 3D editing. Editing static 3D scenes has been explored through diffusion-based refinement, such as InstructNeRF2NeRF [15], which adapts the Pix2Pix paradigm [7] for NeRF-based representations. Similar approaches exist for Gaussian splatting [47], but these methods remain computationally expensive and require retraining for each edit with many 2D diffusion steps from various perspectives.

Visual effects with simulations. Conventional VFX systems rely on physics engines for scene interactions, but their applications to generative 3D content remain limited. Existing approaches incorporate rigid-body physics [49, 52], particle dynamics [13, 26], and elastic deformations [63], all constraint to specific physical interactions. Spring-Gaus [64], for instance, applies spring-mass dynamics to Gaussian splats in scenarios of objects falling onto a plane. More flexible approaches such as PhysGaussian [53] extend capabilities to a few simulation types. An indirect way to facilitate object interactions is to first extract explicit meshes from the scene representation [19, 34, 55, 56, 62]. This allows to use existing mesh-based physics simulators at the cost of error propagation and compute. Unlike these works, our method enables open-ended, user-driven animations without requiring physical constraints, explicit simulations or the preliminary extraction of object meshes.

Interfaces: from code to language. Traditional 3D editing interfaces rely on GUI-based or script-driven tools [9, 19, 26, 39, 60], which often require expert knowledge. This prevents the models to be accessible to a wider user group. Recent approaches integrate LLMs for structured editing by splitting a complex task into smaller sub-elements [12, 48] at the cost of computational overhead. In contrast, our system offers an intuitive direct natural language interface allowing for democratization of 3D content editing while being fast enough to iterate during the creative process.

Towards open-world 3D scene editing. Recent works have explored ways towards more flexible scene editing of Gaussian Splatting scenes. Animate3D [20] allows the animation of single assets but is constrained by dataset limitation and training. Gaussians2Life [50] extends DreamGaussian4D [41] by applying 2D diffusion models to generate motion sequences, which are then lifted to 3D, making them view-consistent at the cost of computational complexity. Our approach bypasses diffusion-based inference by directly modifying Gaussian parameters using field-based transformations which enables interactive updates. Similarly, AutoVFX [18] integrates LLM-generated scripts into Blender-based simulations, producing highly realistic effects at the cost of compute. While producing highly realistic scenes for supported effects, this approach is limited to predefined VFX modules. In contrast, we apply transformations

directly on Gaussian splats, providing a more flexible and efficient framework for text-driven animation.

While previous works have demonstrated powerful generative capabilities in 3D content creation, they often suffer from slow inference times, limited editability, or reliance on predefined physical models. Our method introduces a novel approach by reformulating 3D animation as a field prediction task, allowing interactive, open-world text-driven modifications without additional training or per-scene optimization.

3. Methodology

Our goal is to enable real-time, text-driven animation of 3D scenes without relying on diffusion-based optimization or physics simulators. In this section, we describe our approach in four parts:

(1) representation of 3D scenes as Gaussian splats,
(2) formulation of continuous time-dependent animation fields,
(3) LLM-based translation of open-domain text into parametric fields, and
(4) pipeline details ensuring efficient, user-friendly performance.

Figure 2 provides an overview of our zero-shot pipeline.

3.1. 3D Representation via Gaussian Splatting

We adopt a Gaussian splatting (3DGS) representation for each object or region in our scene. Concretely, an object is approximated by a set of elliptical Gaussians $\{G_i\}_{i=1}^{n}$, where each G_i is defined by:

- $\mu_i \in \mathbb{R}^3$: the center position in 3D,
- $\Sigma_i \in \mathbb{R}^{3\times3}$: a covariance (or scale/rotation) tensor,
- $\mathbf{c}_i \in \mathbb{R}^3$: an RGB color vector (or spherical harmonic coefficients for view dependence),
- $\alpha_i \in \mathbb{R}$: an opacity or density parameter.

This explicit point-based representation can be rendered in real time via fast rasterization [22], and it enables direct manipulation of positions, colors, and opacities without requiring expensive mesh extraction. Thus, any user-specified animation can dynamically update the attributes $\mathbf{A}$ with $\mathbf{A}_i = (\mu_i, \mathbf{c}_i, \alpha_i)$ on the fly.

To animate a specific object within the scene, we first identify its corresponding Gaussians. Our framework remains agnostic to the selection method, enabling seamless integration with automated 3D segmentation techniques. Methods such as LangSplat [38], Gaussian Grouping [59], or GARField [23] can be employed to localize objects within the scene and associate them with their respective Gaussians. In this work, we assume that the segmentation of the target object is provided, with its corresponding Gaussians pre-selected for animation.

3.2. Time-Dependent Fields for Animation

Rather than generating new geometry or using a physics simulator, we define animation through a continuous *time-varying* field f operating on each Gaussian attribute:

$$\begin{aligned} f &: \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R} \times [0,T] \mapsto \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R} \\ f\left(\mathbf{A}_i, t\right) &= \left(f^{\mu}\left(\mu_i, t\right), f^{\mathbf{c}}\left(\mathbf{c}_i, t\right), f^{\alpha}\left(\alpha_i, t\right)\right) \\ &= \left(\mu_i(t), \mathbf{c}_i(t), \alpha_i(t)\right) =: \mathbf{A}_i(t) \end{aligned} \tag{1}$$

where i indicates a particular Gaussian, and $t \in [0,T], T \in \mathbb{R}$ spans the animation's duration. These fields allow structured transformations such as translation, color transitions, or opacity changes over time.

3.3. LLM-Based Translation of Text Instructions

We now describe how open-domain text instructions are converted into parametric fields that update each Gaussian's center, color, and opacity. Our pipeline comprises multiple stages, each realized by a specialized LLM prompt. If additional renderings of the object are available, we can also involve a VLM for more precise attribute grounding. The detailed text instructions to the L(V)LMs that are mentioned here schematically can be found in the supplementary material.

3.3.1 Design Phase

A user provides an animation description (e.g., "move the vase up for two seconds, then dissolve it over one second"). We first request that the LLM break down this abstract instruction into animation phases, specifying approximate timings and actions. For instance, it might produce:

- `Phase 1 (0-2s)`: "translate vase upward,"
- `Phase 2 (2-3s)`: "fade from opaque to transparent."

In doing so, the LLM interprets the user's high-level goals into a structured set of phases. If images of the object are provided, it can tailor the breakdown accordingly (e.g., factoring in bounding-box dimensions or shape cues). Simple static text instructions guarantee a structured output.

3.3.2 Field Generation Phase

The output from the desing phase is now translated by LLM calls into $\mu_i(t), \mathbf{c}_i(t), \alpha_i(t)$ (cf. Eq. 1), still considering user-specified durations and transformations. For example, the geometry function might linearly interpolate the vase's z-coordinate from z_0 to $z_0 + 2$ over $t \in [0,2]$, while the opacity function steadily decreases α from 1 to 0 during $t \in [2,3]$.

3.3.3 Animation Hypotheses Generation & Scoring

Some textual instructions might be ambiguous (e.g., "make the vase shake randomly"). For those, we can generate m

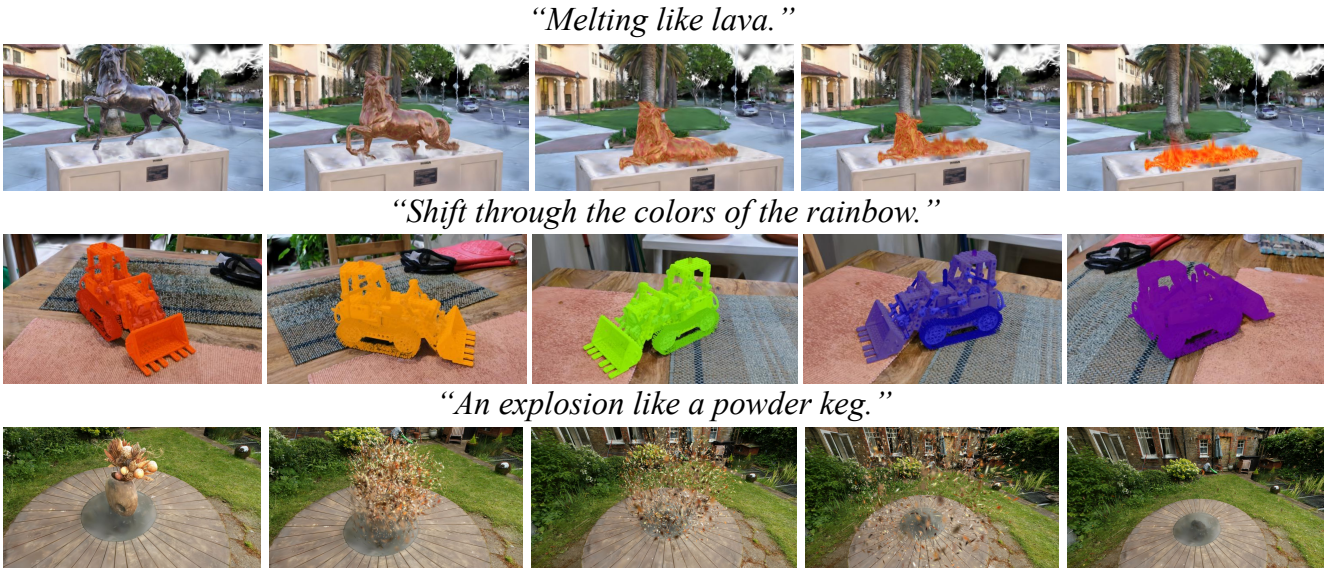

Figure 3. **Qualitative results** showcasing the diversity and fidelity of animations generated by PromptVFX across different scenes and user prompts. Exact prompts used to generate these animations are provided in the supplementary material.

variations $f_1, f_2, \ldots, f_m$ of these parametric functions

$$f_j^{\mu}\left(\mu_i, t\right), f_j^{\mathbf{c}}\left(\mathbf{c}_i, t\right), f_j^{\alpha}\left(\alpha_i, t\right), \quad j \in \{1, 2, \ldots, m\}. \tag{2}$$

Each variation applies slightly different motion curves or color transitions. We then render each candidate animation (producing, for instance, a short sequence of frames) and feed these frames, along with the original text prompt, to a VLM that scores how well each animation matches the intended description on a 0–100 scale. We select the highest-scoring candidate as our base animation. We also include two optional refinement processes to improve the output.

3.3.4 Automatic Refinement

To further refine the chosen base animation, we prompt a specialized VLM that compares the user's description and the rendered frames. It identifies discrepancies (e.g., "the vase does not rise high enough") and adjusts our field functions $\mu_i(t), \mathbf{c}_i(t), \alpha_i(t)$ to more closely match the user's instructions.

3.3.5 Conversational Refinement

If a user observes misinterpretations of the intended animation, open-text feedback can be provided (e.g., "spin faster in the first second, fade more quickly in the final half-second"). We combine this feedback with the existing field A_i and animation frames, asking the VLM to produce another iteration of refined functions. This conversational loop can continue until the user is satisfied. This enables rapid iteration cycles with a natural text-driven interface due to iteration without computational overhead from diffusion models or physics-based simulations.

4. Experiments

4.1. Experimental Details

Dataset & Preprocessing. We evaluate our method on real-world scenes from Mip-NeRF360 [4] (the *garden vase* and *bulldozer*), *bear* scene from Instruct-NeRF2NeRF [15] and an additional *horse* scene from Tanks and Temples [24]. Each scene is reconstructed via a Gaussian splatting pipeline, providing a set of Gaussians encoding geometry and color. When animating, user instructions (e.g., "the vase") are mapped to corresponding Gaussians.

Baselines. We compare against Gaussians2Life [50], which uses video diffusion to synthesize 2D motion lifted to 3D Gaussians via optimization, and AutoVFX [18], which generates physically simulated animations through Blender scripts, requiring mesh extraction and offline rendering. Both pipelines depend on heavy preprocessing and lack interactivity. In contrast, our method applies functional transformations directly to Gaussians, enabling real-time animation without diffusion or physics simulation.

Implementation Notes. Our method is implemented in Python and leverages GPT-4 [1] to parse text prompts and generate animation functions. All experiments are conducted on a single NVIDIA RTX 4090 GPU.

4.2. Qualitative Evaluation

We demonstrate the capabilities of PromptVFX across a variety of scenes and user instructions. Figure 3 shows that our system can generate animations involving motion,

Figure 4. **Qualitative comparison** with baselines on different scenes and user prompts. Our method achieves high-fidelity visual transformations and realistic motion, outperforming AutoVFX and Gaussians2Life. Exact prompts and additional qualitative comparisons are provided in the supplementary material.

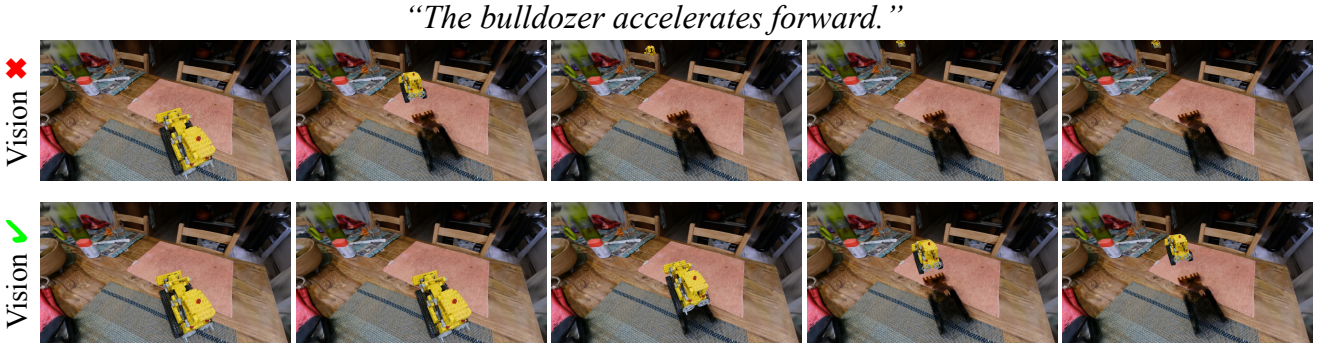

Figure 5. **Impact of VLM** feedback on animation accuracy. Without visual feedback, the bulldozer incorrectly accelerates off the table, failing to account for scene constraints. With VLM refinement, the motion is corrected to remain contextually appropriate.

color changes, and opacity variations in response to natural language prompts. Each animation is produced by updating Gaussian parameters such as position and appearance in real time, enabling interactive feedback and rapid iteration during

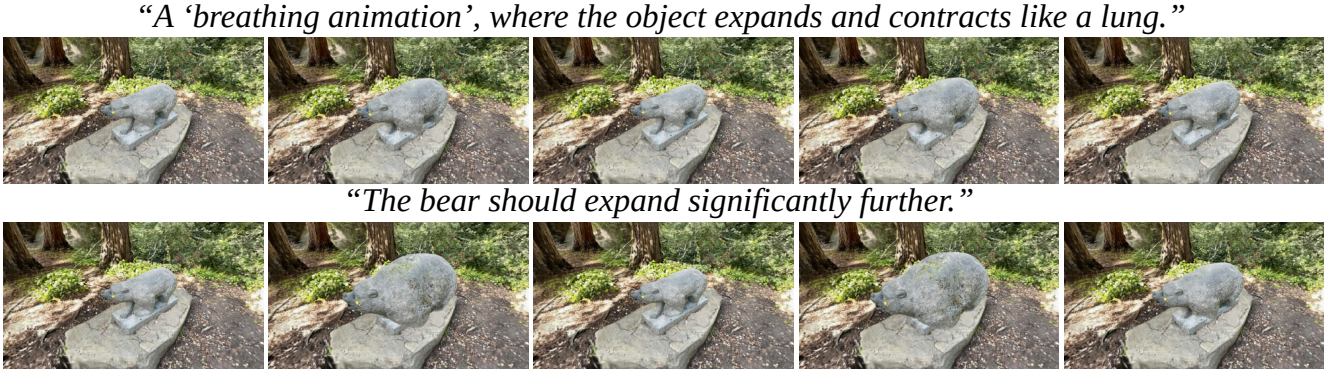

Figure 6. **Animation refinement** is demonstrated through iterative user interaction. After generating an initial animation, the system enables users to provide follow-up prompts for further adjustments, enhancing control and precision in the final animation.

Table 2. Comparison of methods across CLIP similarity, VQAScore, and User Study ratings. User Study includes Text Alignment and Animation Quality scores (1–5 Likert scale, normalized). Bold indicates best per prompt and metric.

Prompt	Method	CLIP [40]	VQAScore [28]	User Study	
				Text Alignment	Animation Quality
Turn the vase into lava.	Gaussians2Life [50]	**0.206**	0.196	0.043	0.235
	AutoVFX [18]	0.159	0.444	0.615	0.485
	Ours	0.171	**0.715**	**0.750**	**0.593**
The bear expands and contracts like a lung.	Gaussians2Life [50]	0.238	0.258	0.235	0.285
	AutoVFX [18]	**0.239**	**0.283**	0.073	0.315
	Ours	0.204	**0.283**	**0.693**	**0.465**
The bulldozer accelerates forward.	Gaussians2Life [50]	**0.308**	0.681	0.278	0.235
	AutoVFX [18]	0.172	0.451	0.035	0.058
	Ours	0.227	**0.685**	**0.893**	**0.685**

the creative process.

Comparison with Baselines. Figure 4 provides a visual comparison between PromptVFX and two recent methods, Gaussians2Life [50] and AutoVFX [18]. Unlike these approaches, which rely on diffusion models or physics simulations and require offline processing, our framework applies direct functional transformations to the Gaussians. This design results in more responsive editing, better temporal coherence, and consistent appearance transformations across views. PromptVFX delivers high-quality animations that align with user prompts while maintaining the speed and flexibility required for interactive content creation.

4.3. Quantitative Evaluation

Evaluating 3D animation quality is difficult due to the absence of ground-truth sequences. Following prior work [8, 18, 50], we report CLIP similarity [40] as a frame-level proxy for text-to-image alignment. However, CLIP does not capture motion or temporal coherence. We therefore also report VQAScore, a video-level metric shown to better align with human judgment [28]. It estimates the probability that a model answers "Yes" to the prompt: *Does this video align with the described animation: "{prompt}"?*, denoted as $\mathbb{P}(\text{"Yes"} \mid \text{video, prompt})$.

Table 2 shows that **PromptVFX** achieves the highest VQAScore [28] across all evaluated prompts, demonstrating better text-video alignment from a temporal and semantic perspective. While Gaussians2Life [50] ranks higher in CLIP [40] for some cases due to preserving frame-level appearance, it does not always reflect coherent or realistic motion.

To further evaluate perceptual quality, we conducted a user study with 35 participants, following the protocol from AutoVFX [18]. Participants rated the generated videos on two criteria: *Text Alignment* and *Animation Quality*, using a 1-5 Likert scale (normalized to [0, 1] using min-max normalization). As shown in Table 2, PromptVFX outperforms all baselines across both criteria and all prompts, demonstrating its effectiveness in producing coherent and semantically grounded 3D animations.

4.4. Ablation Studies

We conduct several ablation studies to evaluate the impact of different components in our pipeline.

4.4.1 Impact of Vision-Language Model Feedback

To assess the role of VLM-based feedback, we compare animations generated with and without visual refinement. As

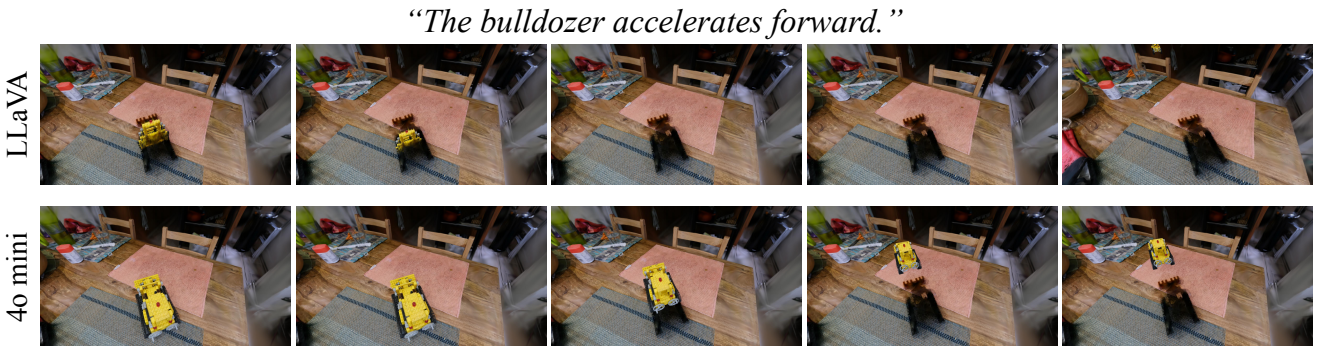

Figure 7. **VLM comparison** for animation quality effect with different vision-language models (VLMs) that interpret the prompt. Given the same input text, GPT-4o-mini produces a coherent bulldozer acceleration, while LLaVA struggles to maintain object consistency and motion realism.

shown in Figure 5, without visual feedback, the bulldozer accelerates off the table, failing to respect scene constraints. When rendered frames are provided to the VLM for refinement, the animation is corrected to maintain a physically plausible motion. This highlights the importance of incorporating visual guidance in aligning animations with scene context.

4.4.2 User-Guided Animation Refinement

We examine the benefits of allowing iterative user feedback in refining animations. In Figure 6 an initial animation is generated based on the text prompt, after which the user provides a follow-up adjustment. The refined animation better captures the intended motion, demonstrating how interactive refinements improve animation precision. This experiment underscores the importance of keeping the user in the loop for fine-grained control over the animation generation process.

4.4.3 Effect of VLM Choice on Animation Quality

We compare different vision-language models used in our pipeline. Figure 7 illustrates that GPT-4o-mini produces a fluent bulldozer acceleration, while LLaVA [30, 31] struggles with maintaining object consistency and motion realism.

This suggests that model selection significantly affects animation quality and highlights the need for well-designed system prompts tailored to the VLM's capabilities.

4.4.4 Effect of Hypothesis Sampling

We analyze how the number of generated animation hypotheses influences performance. Figure 8 shows that increasing the number of candidate animations improves VLM scores. The VLM score is obtained by querying a vision-language model to evaluate how well snapshots of the animation from different viewpoints match the original text prompt, providing a more holistic assessment of animation coherence. However, we observe diminishing returns beyond a certain number of hypotheses, indicating that an optimal balance exists between diversity and computational efficiency.

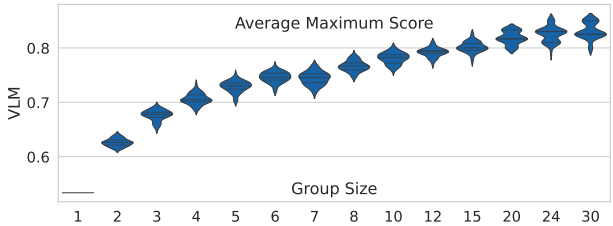

Figure 8. **Best Scores for various amount of hypotheses.** Impact of the number of hypotheses on VLM scores. As the group size increases, VLM scores improve.

5. Limitations

While PromptVFX delivers fast, interactive text-driven volumetric animations, it has several constraints. It cannot support collisions or contact interactions. It can only animate existing Gaussians, so particle effects such as smoke or fire cannot be generated. Without semantic scene understanding, relational prompts such as "place the vase under the table" may not be interpreted correctly. In future work, we plan to integrate physics priors, enable dynamic splat insertion for particle effects, and incorporate semantic object labels to address these limitations.

6. Conclusion

This paper introduces PromptVFX, a framework that reformulates 3D animation as a field prediction task by applying time-varying transformations directly to Gaussian splats. Unlike diffusion-based or physics-driven pipelines, our method enables real-time, text-driven animation through parametric updates generated by large language and vision-language models. The approach reduces manual effort, supports fast iteration, and produces high-fidelity, temporally consistent results. By lowering the barriers to 3D animation, this work takes a step toward democratizing open-ended visual effects creation.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 5

[2] Sherwin Bahmani, Ivan Skorokhodov, Victor Rong, Gordon Wetzstein, Leonidas Guibas, Peter Wonka, Sergey Tulyakov, Jeong Joon Park, Andrea Tagliasacchi, and David 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*, pages 7996–8006, 2024. 3

[3] Sherwin Bahmani, Ivan Skorokhodov, Aliaksandr Siarohin, Willi Menapace, Guocheng Qian, Michael Vasilkovsky, Hsin-Ying Lee, Chaoyang Wang, Jiaxu Zou, Andrea Tagliasacchi, et al. Vd3d: Taming large video diffusion transformers for 3d camera control. *CoRR*, 2024. 3

[4] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *CVPR*, 2022. 5

[5] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *CoRR*, 2023. 3

[6] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22563–22575, 2023. 3

[7] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18392–18402, 2023. 3

[8] Minghao Chen, Iro Laina, and Andrea Vedaldi. Dge: Direct gaussian 3d editing by consistent multi-view editing. In *European Conference on Computer Vision*, pages 74–92. Springer, 2024. 2, 7

[9] Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin. Gaussianeditor: Swift and controllable 3d editing with gaussian splatting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21476–21485, 2024. 2, 3

[10] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2019. 1

[11] Devikalyan Das, Christopher Wewer, Raza Yunus, Eddy Ilg, and Jan Eric Lenssen. Neural parametric gaussians for monocular non-rigid object reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10715–10725, 2024. 3

[12] Shuangkang Fang, Yufeng Wang, Yi-Hsuan Tsai, Yi Yang, Wenrui Ding, Shuchang Zhou, and Ming-Hsuan Yang. Chat-edit-3d: Interactive 3d scene editing via text prompts. In *European Conference on Computer Vision*, pages 199–216. Springer, 2024. 2, 3

[13] Yutao Feng, Xiang Feng, Yintong Shang, Ying Jiang, Chang Yu, Zeshun Zong, Tianjia Shao, Hongzhi Wu, Kun Zhou, Chenfanfu Jiang, and Yin Yang. Gaussian splashing: Unified particles for versatile motion synthesis and rendering. *arXiv preprint arXiv:2401.15318*, 2024. 2, 3

[14] Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14953–14962, 2023. 2

[15] Ayaan Haque, Matthew Tancik, Alexei Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 2, 3, 5

[16] Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for text-to-video generation. *CoRR*, 2024. 3

[17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 3

[18] Hao-Yu Hsu, Zhi-Hao Lin, Albert Zhai, Hongchi Xia, and Shenlong Wang. Autovfx: Physically realistic video editing from natural language instructions. *arXiv preprint arXiv:2411.02394*, 2024. 2, 3, 5, 7

[19] Ying Jiang, Chang Yu, Tianyi Xie, Xuan Li, Yutao Feng, Huamin Wang, Minchen Li, Henry Lau, Feng Gao, Yin Yang, et al. Vr-gs: A physical dynamics-aware interactive gaussian splatting system in virtual reality. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–1, 2024. 2, 3

[20] Yanqin Jiang, Chaohui Yu, Chenjie Cao, Fan Wang, Weiming Hu, and Jin Gao. Animate3d: Animating any 3d model with multi-view video diffusion. *Advances in Neural Information Processing Systems*, 37:125879–125906, 2025. 3

[21] HyunJun Jung, Nikolas Brasch, Jifei Song, Eduardo Perez-Pellitero, Yiren Zhou, Zhihao Li, Nassir Navab, and Benjamin Busam. Deformable 3d gaussian splatting for animatable human avatars, 2023. 3

[22] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 2, 3, 4

[23] Chung Min* Kim, Mingxuan* Wu, Justin* Kerr, Matthew Tancik, Ken Goldberg, and Angjoo Kanazawa. Garfield: Group anything with radiance fields. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 4

[24] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017. 5

[25] Zhengfei Kuang, Shengqu Cai, Hao He, Yinghao Xu, Hongsheng Li, Leonidas J Guibas, and Gordon Wetzstein. Collaborative video diffusion: Consistent multi-video generation with camera control. In *Advances in Neural Information Processing Systems*, pages 16240–16271. Curran Associates, Inc., 2024. 3

[26] Yuan Li, Zhi-Hao Lin, David Forsyth, Jia-Bin Huang, and Shenlong Wang. Climatenerf: Extreme weather synthesis in neural radiance field. In *Proceedings of the IEEE/CVF*

International Conference on Computer Vision, pages 3227–3238, 2023. 2, 3

[27] Youtian Lin, Zuozhuo Dai, Siyu Zhu, and Yao Yao. Gaussian-flow: 4d reconstruction with dynamic 3d gaussian particle. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21136–21145, 2024. 3

[28] Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation. In *European Conference on Computer Vision*, pages 366–384. Springer, 2024. 7

[29] Huan Ling, Seung Wook Kim, Antonio Torralba, Sanja Fidler, and Karsten Kreis. Align your gaussians: Text-to-4d with dynamic 3d gaussians and composed diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8576–8588, 2024. 3

[30] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. 8

[31] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024. 8

[32] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *2024 International Conference on 3D Vision (3DV)*, pages 800–809, 2024. 3

[33] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: representing scenes as neural radiance fields for view synthesis. *Commun. ACM*, 65(1):99–106, 2021. 3

[34] Yicong Peng, Yichao Yan, Shengqi Liu, Yuhao Cheng, Shanyan Guan, Bowen Pan, Guangtao Zhai, and Xiaokang Yang. Cagenerf: Cage-based neural radiance field for generalized 3d deformation and animation. In *Advances in Neural Information Processing Systems*, pages 31402–31415. Curran Associates, Inc., 2022. 3

[35] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 2, 3

[36] Zhiyin Qian, Shaofei Wang, Marko Mihajlovic, Andreas Geiger, and Siyu Tang. 3dgs-avatar: Animatable avatars via deformable 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5020–5030, 2024. 3

[37] Yi-Ling Qiao, Alexander Gao, Yiran Xu, Yue Feng, Jia-Bin Huang, and Ming C Lin. Dynamic mesh-aware radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 385–396, 2023. 2

[38] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20051–20060, 2024. 4

[39] Ri-Zhao Qiu, Ge Yang, Weijia Zeng, and Xiaolong Wang. Language-driven physics-based scene synthesis and editing via feature splatting. In *European Conference on Computer Vision*, pages 368–383. Springer, 2024. 3

[40] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 7

[41] Jiawei Ren, Liang Pan, Jiaxiang Tang, Chi Zhang, Ang Cao, Gang Zeng, and Ziwei Liu. Dreamgaussian4d: Generative 4d gaussian splatting. *arXiv preprint arXiv:2312.17142*, 2023. 2, 3

[42] 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*, pages 10684–10695, 2022. 1, 3

[43] Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. Zero123++: a single image to consistent multi-view diffusion base model. *arXiv preprint arXiv:2310.15110*, 2023. 2

[44] Zifan Shi, Sida Peng, Yinghao Xu, Andreas Geiger, Yiyi Liao, and Yujun Shen. Deep generative models on 3d representations: A survey. *arXiv preprint arXiv:2210.15663*, 2022. 1

[45] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. In *The Eleventh International Conference on Learning Representations*, 2023. 3

[46] Uriel Singer, Shelly Sheynin, Adam Polyak, Oron Ashual, Iurii Makarov, Filippos Kokkinos, Naman Goyal, Andrea Vedaldi, Devi Parikh, Justin Johnson, et al. Text-to-4d dynamic scene generation. *arXiv preprint arXiv:2301.11280*, 2023. 3

[47] Cyrus Vachha and Ayaan Haque. Instruct-gs2gs: Editing 3d gaussian splats with instructions, 2024. 3

[48] Junjie Wang, Jiemin Fang, Xiaopeng Zhang, Lingxi Xie, and Qi Tian. Gaussianeditor: Editing 3d gaussians delicately with text instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20902–20911, 2024. 3

[49] Yuxi Wei, Zi Wang, Yifan Lu, Chenxin Xu, Changxing Liu, Hao Zhao, Siheng Chen, and Yanfeng Wang. Editable scene simulation for autonomous driving via collaborative llm-agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15077–15087, 2024. 3

[50] Thomas Wimmer, Michael Oechsle, Michael Niemeyer, and Federico Tombari. Gaussians-to-life: Text-driven animation of 3d gaussian splatting scenes. In *2025 International Conference on 3D Vision (3DV)*, 2025. 2, 3, 5, 7

[51] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20310–20320, 2024. 3

[52] Hongchi Xia, Zhi-Hao Lin, Wei-Chiu Ma, and Shenlong Wang. Video2game: Real-time interactive realistic and browser-compatible environment from a single video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4578–4588, 2024. 3

[53] Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4389–4398, 2024. 2, 3

[54] Yiming Xie, Chun-Han Yao, Vikram Voleti, Huaizu Jiang, and Varun Jampani. Sv4d: Dynamic 3d content generation with multi-frame and multi-view consistency. *arXiv preprint arXiv:2407.17470*, 2024. 3

[55] Tianhan Xu and Tatsuya Harada. Deforming radiance fields with cages. In *Computer Vision – ECCV 2022*, pages 159–175, Cham, 2022. Springer Nature Switzerland. 3

[56] Bangbang Yang, Chong Bao, Junyi Zeng, Hujun Bao, Yinda Zhang, Zhaopeng Cui, and Guofeng Zhang. Neumesh: Learning disentangled neural mesh-based implicit field for geometry and texture editing. In *Computer Vision – ECCV 2022*, pages 597–614, Cham, 2022. Springer Nature Switzerland. 3

[57] Jiayu Yang, Ziang Cheng, Yunfei Duan, Pan Ji, and Hongdong Li. Consistnet: Enforcing 3d consistency for multi-view images diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7079–7088, 2024. 2

[58] Shuai Yang, Yifan Zhou, Ziwei Liu, and Chen Change Loy. Fresco: Spatial-temporal correspondence for zero-shot video translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8703–8712, 2024. 2

[59] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. In *ECCV*, 2024. 4

[60] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. In *European Conference on Computer Vision*, pages 162–179. Springer, 2024. 2, 3

[61] Heng Yu, Chaoyang Wang, Peiye Zhuang, Willi Menapace, Aliaksandr Siarohin, Junli Cao, László Jeni, Sergey Tulyakov, and Hsin-Ying Lee. 4real: Towards photorealistic 4d scene generation via video diffusion models. *Advances in Neural Information Processing Systems*, 37:45256–45280, 2025. 3

[62] Yu-Jie Yuan, Yang-Tian Sun, Yu-Kun Lai, Yuewen Ma, Rongfei Jia, and Lin Gao. Nerf-editing: Geometry editing of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18353–18364, 2022. 3

[63] Tianyuan Zhang, Hong-Xing Yu, Rundi Wu, Brandon Y. Feng, Changxi Zheng, Noah Snavely, Jiajun Wu, and William T. Freeman. Physdreamer: Physics-based interaction with 3d objects via video generation. In *Computer Vision – ECCV 2024*, pages 388–406, Cham, 2025. Springer Nature Switzerland. 3

[64] Licheng Zhong, Hong-Xing Yu, Jiajun Wu, and Yunzhu Li. Reconstruction and simulation of elastic objects with spring-mass 3d gaussians. *European Conference on Computer Vision (ECCV)*, 2024. 3