Controllable Dynamic 3D Shape Generation via 3D Trajectories and Text

Jaeyeong Kim **Ines Kim** **Jahyeok Koo** **Seungryong Kim**

KAIST AI

Project Page: `https://cvlab-kaist.github.io/T2Mo`

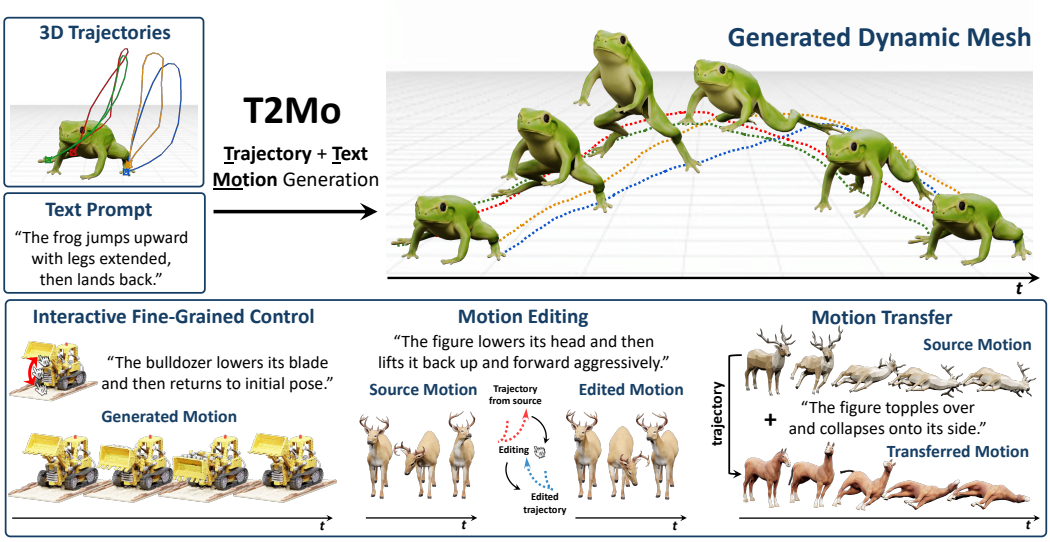

Figure 1: We present **T2Mo**, a generative framework for controllable dynamic 3D shape generation guided by 3D trajectories and text. By combining spatial trajectory conditioning with semantic text conditioning, T2Mo generates spatially controlled motions. The versatility of trajectories also enables diverse applications, including interactive fine-grained control, motion editing, and motion transfer.

Abstract

We introduce **T2Mo**, a generative framework for controllable dynamic 3D shape generation conditioned on 3D trajectories and text. Due to the inherent ambiguity of language, generating precisely intended motions using text alone remains challenging. To address this, we adopt 3D trajectories as controllable spatial guidance, specifying the exact paths along which selected points should move. By combining both, T2Mo generates object motions that spatially adhere to the given trajectories while globally reflecting the text semantics. To robustly handle trajectory inputs with arbitrary configurations—ranging from dense to sparse and unevenly distributed—we further propose a shape-grounded trajectory embedding that maps an input trajectory set into a shape-aware token set covering the entire object. We conduct extensive comparisons against text-based baselines and cascaded video-based baselines that combine trajectory-guided video generation with video-to-dynamic mesh generation. Quantitative and qualitative evaluations, along with user studies, demonstrate that our approach produces motions that more faithfully follow the given prompts with higher expressiveness while preserving motion quality.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

1 Introduction

Dynamic 3D shape generation has become an important research problem because of its wide-ranging applications in AR/VR, gaming, filmmaking, and other domains that require temporally coherent 3D assets. Recent advances in feed-forward 3D generative models [12, 23–25, 27, 49, 56, 58] have made it possible to synthesize dynamic 3D shapes more efficiently than earlier optimization-based methods [2, 26, 29, 55] by extending 3D generative models in the temporal dimension [19, 37, 53] or generating motion of a given 3D object [4, 42, 47, 48]. Despite these advances, controllability of dynamic shape generation remains underexplored, as current models lack explicit mechanisms to precisely specify the intended motion. Providing users with intuitive control over how objects move remains an open challenge, motivating the need for controllable dynamic 3D shape generation.

To achieve controllability in visual generation tasks, recent approaches employ text conditioning as high-level semantic guidance alongside additional control signals to mitigate the inherent ambiguity of language. For instance, in video synthesis and human motion generation, prior works introduce auxiliary controls such as bounding boxes [30, 41], motion vectors [5, 10, 40, 54, 59], segmentation masks [46], or kinematic constraints [11, 20, 35, 39]. While these methods enable controllable generation by allowing users to specify desired motions in pixel space or via domain-specific modalities like skeletons, they are highly tailored to their respective domains and cannot be directly applied to dynamic 3D mesh generation. Consequently, providing intuitive motion control for dynamic 3D generation remains an open problem.

We address this gap by representing user control signals directly on the mesh through *3D trajectories*. By encoding the path of a point on the target mesh in 3D space across frames, 3D trajectories provide an intuitive and interactive way to express motion control without requiring any additional transformations. Building on this representation, we introduce **T2Mo**, a generative framework for controllable dynamic 3D generation of general objects. The framework conditions the generative model on a set of user-provided trajectories together with a text prompt that provides global motion semantics. This enables our framework to support controllable motion generation that adheres to the user-provided trajectories while remaining consistent with the global semantics of the text prompt.

However, user-provided trajectories may take diverse configurations in practice—from dense to sparse and unevenly distributed—depending on the target motion, the object geometry, and the application. To robustly generate plausible motions across such varying trajectory configurations, the model should process each trajectory adaptively with respect to the input shape, while remaining consistent regardless of how the input trajectories are configured. For example, a trajectory placed on the leg of a person with the prompt "walking" should affect only that leg part, whether it is the sole input or one among many. To this end, we design a *shape-grounded trajectory embedding* that maps the input trajectory set into a predefined fixed-size, shape-aware token set covering the entire shape independently of the input trajectory configuration. Intuitively, this input trajectory-agnostic coverage of the entire shape gives consistent conditioning regardless of the input configuration, while attaching each trajectory to a shape-aware token localizes its effect in a geometry-aware manner.

We evaluate T2Mo on a diverse set of objects and compare it against text-based baselines [42, 47] and cascaded baselines which combine trajectory-conditioned video generation and video-to-dynamic mesh generation [4, 37, 59] through quantitative metrics including VBench [17], trajectory alignment, and motion magnitude [48], alongside user study. The comparison indicates that introducing trajectory guidance improves both controllability and expressiveness by providing explicit motion cues, without compromising the overall quality of the generated motion. We further showcase qualitative results, demonstrating that our framework enables a diverse range of motion controls and applications.

In summary, our contributions are as follows:

- We present **T2Mo**, a generative framework for controllable dynamic 3D shape generation of general objects, conditioned on *3D trajectories* and a *text* prompt.
- We introduce a **shape-grounded trajectory embedding** that maps arbitrarily configured input trajectories into a predefined shape-aware token set covering the entire object, providing consistent conditioning regardless of configuration while preserving geometric locality.
- Extensive experiments and user preference studies demonstrate that our framework enables various user-defined motion controls, improving both controllability and expressiveness over recent baselines without compromising the quality of the generated motion.

2 Related work

Dynamic 3D generation. Similar to static 3D generation, early works on dynamic 3D generation predominantly adopt optimization-based formulations [2, 26, 28, 29, 55]. Such methods commonly distill a 2D generative prior through Score Distillation Sampling (SDS) [32] and its variants [43]. However, due to their limitations such as view inconsistency and the Janus problem [14], subsequent works shifted toward extending 2D generative models to multi-view settings [18, 33, 50, 57] for dynamic 3D reconstruction [9, 15, 16, 36, 38, 45]. Recently, as feed-forward 3D shape generation models [12, 23–25, 27, 49, 56, 58] have shown remarkable progress, there have been attempts to leverage these models to extend static 3D generation into dynamic generation [37, 53]. Alternatively, an emerging line of work orthogonally decomposes this task, independently handling static 3D asset creation and per-vertex trajectory synthesis [4, 19, 42, 47]. Building on this idea, recent text-guided dynamic 3D generation methods adopt the approach of animating predefined 3D meshes. Specifically, AnimateAnyMesh [47] and AnimateAnyMesh++ [48] utilize a rectified flow model with trajectory VAE to generate motion conditioned on a text and an initial shape. BiMotion [42] introduces a B-spline-based motion representation to handle motion of arbitrary frame lengths. Adopting this decoupling approach, we generate per-vertex displacements over time for a given input shape. This formulation is particularly suitable for controllable generation, as it allows users to directly provide guidance based on the underlying geometry. Although these advancements successfully generate dynamic 3D shapes from given contexts in a feed-forward manner, they remain limited in providing the sufficient controllability required for practical applications. Motivated by this limitation, we propose a controllable generation approach for general objects through trajectory-based guidance.

Trajectory-guided motion control. Owing to their spatiotemporal expressiveness, point trajectories have been widely adopted as a versatile representation for controllable generation across a range of tasks. Video generation, one of the most representative tasks, commonly employs sparse or dense trajectories in 2D pixel space as a spatial guidance of object or camera motion. Existing approaches integrate such trajectory guidance through feature-level modulation [5], ControlNet-style conditioning [10, 40], or specialized trajectory encoders [44, 46, 54, 59]. More recent extensions move beyond 2D image-plane trajectories by introducing 3D pose sequence [8] or trajectory with depth [40], mainly focusing on specifying the object-level global position trajectory. Similar principles have been applied to human motion synthesis [11, 20, 21, 35, 39, 51], utilizing trajectories as spatial constraints to guide global paths or specific actions, often conjugated with other condition modalities. However, these methods are largely limited to articulated human body motion, where the control mechanism is tightly coupled to a predefined skeletal structure. In the context of 4D generation, although TC4D [3] proposes trajectory-conditioned generation, trajectories are used solely to define the scene-level per-object position paths, rather than to model local object motion itself, leaving the local, intra-object motion uncontrolled. Unlike previous works, we propose a general-purpose framework that establishes 3D trajectories as a versatile interface for motion control.

3 Method

3.1 Overview

Our goal is to enable explicit control over dynamic 3D shape generation using two complementary signals: a *set of 3D trajectories* that specify point-wise motion on the mesh and a *text prompt* that conveys high-level motion semantics. However, in practice, user-provided trajectories are often sparse and unevenly distributed, and naively injecting them often fails to yield a well-localized conditioning signal on the shape. To handle such variability, we introduce **shape-grounded trajectory embedding**, which converts input trajectory set with arbitrary configuration into a predefined shape-aware token set covering the entire shape. These tokens, together with text embeddings, condition a diffusion-based generative model built on DiT [31], which synthesizes per-vertex displacements for a given static mesh. This section first formalizes the task (Sec. 3.2), then describes the shape-grounded trajectory embedding and generative architecture (Sec. 3.3), and outlines training and inference (Sec. 3.4).

3.2 Problem formulation

We formalize controllable dynamic 3D shape generation as generating a sequence of deformed meshes given a static source mesh and user-provided controls. Let $\mathcal{M}_0 = (\mathbf{V}_0, \mathbf{F}_0)$ be a source

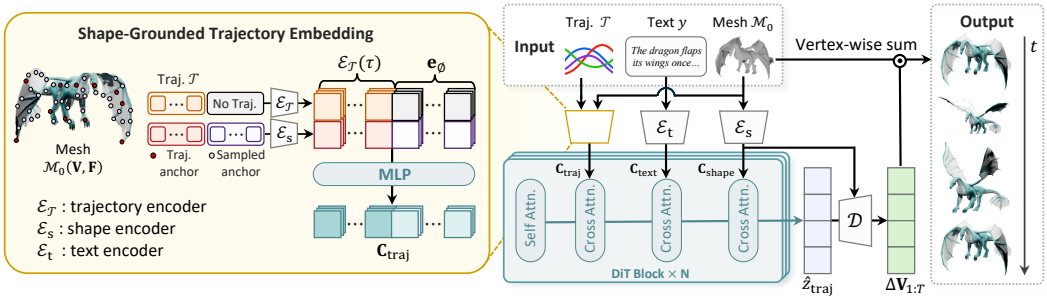

Figure 2: Overview of T2Mo. Given a source mesh $\mathcal{M}_0$, condition 3D trajectories $\mathcal{T}$, and a text prompt y, our model generates per-vertex displacements $\Delta\mathbf{V}_{1:T}$ via a DiT [31] backbone conditioned on shape $\mathbf{C}_{\text{shape}}$, text $\mathbf{C}_{\text{text}}$, and trajectory tokens $\mathbf{C}_{\text{traj}}$ through cross-attention, yielding the output mesh sequence $\{\mathcal{M}_i\}_{i=1}^{T}$ via the decoder $\mathfrak{D}$. The trajectory condition is constructed by our *shape-grounded trajectory embedding* (yellow box): each input trajectory is bound to its source point on the mesh (red dots), and the remaining anchors are sampled by farthest point sampling (white dots). For each anchor, the shape feature from $\mathcal{E}_s$ is concatenated with either a trajectory feature from $\mathcal{E}_\mathcal{T}$ or a learnable null embedding $\mathbf{e}_\varnothing$, and projected through an MLP to form the trajectory conditions $\mathbf{C}_{\text{traj}}$.

mesh with N vertices $\mathbf{V}_0 \in \mathbb{R}^{N\times 3}$ and M faces $\mathbf{F}_0 \in \mathbb{Z}^{M\times 3}$. The task is to generate per-vertex displacements $\Delta\mathbf{V}_{1:T} = \{\Delta\mathbf{V}_t\}_{t=1}^{T}$, where each $\Delta\mathbf{V}_t \in \mathbb{R}^{N\times 3}$, over T frames. Each output frame is reconstructed as $\mathcal{M}_t = (\mathbf{V}_0 + \Delta\mathbf{V}_t, \mathbf{F}_0)$. The user provides a text prompt y describing global motion semantics and a set of 3D trajectories $\mathcal{T} = \{\tau_k\}_{k=1}^{K}$ of arbitrary cardinality K. Each trajectory $\tau_k = (\mathbf{p}_{k,0}, \mathbf{p}_{k,1}, \dots)$ is a sequence of 3D waypoints, where $\mathbf{p}_{k,i} \in \mathbb{R}^3$ denotes the i-th waypoint.

Following prior work [42, 47, 48], we compress the static mesh and its motion into latent codes using a variational autoencoder (VAE). The VAE encoders map the source mesh and displacement sequence into a shape $\mathbf{z}_{\text{shape}}$ and a trajectory latent $\mathbf{z}_{\text{traj}}$ with two separate streams in the VAE encoder:

$$\mathbf{z}_{\text{shape}} = \mathcal{E}_{\text{shape}}(\mathcal{M}_0), \qquad \mathbf{z}_{\text{traj}} = \mathcal{E}_{\text{traj}}(\Delta\mathbf{V}_{1:T}) \tag{1}$$

where $\mathcal{E}_{\text{shape}}$ and $\mathcal{E}_{\text{traj}}$ denote the shape and trajectory streams, respectively. At inference, a generative model $\mathcal{G}_\theta$ produces a trajectory latent $\hat{\mathbf{z}}_{\text{traj}}$ from Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, conditioned on $\mathbf{z}_{\text{shape}}$ and the text and trajectory condition tokens $\mathbf{C}_{\text{text}}$ and $\mathbf{C}_{\text{traj}}$:

$$\hat{\mathbf{z}}_{\text{traj}} = \mathcal{G}_\theta(\boldsymbol{\epsilon};\, \mathbf{z}_{\text{shape}},\, \mathbf{C}_{\text{text}}, \mathbf{C}_{\text{traj}}), \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \qquad \Delta\mathbf{V}_{1:T} = \mathcal{D}(\mathbf{z}_{\text{shape}},\, \hat{\mathbf{z}}_{\text{traj}}), \tag{2}$$

where $\mathcal{D}$ is the VAE decoder. For notational consistency, we hereafter define $\mathbf{C}_{\text{shape}} := \mathbf{z}_{\text{shape}}$.

3.3 T2Mo: Trajectory and text-guided dynamic 3D generation

Fig. 2 illustrates the overall architecture of **T2Mo**, including the generative model and the shape-grounded trajectory embedding.

Shape-grounded trajectory embedding. Simply treating each trajectory as an independent token neglects the geometry of the shape and leads to poorly localized conditioning (Fig. 5). Instead, we sample a fixed set of N_{anchor} anchor points $\{(\mathbf{s}_i, \mathbf{n}_i)\}_{i=1}^{N_{\text{anchor}}}$ on the surface, where each anchor consists of a position $\mathbf{s}_i \in \mathbb{R}^3$ and a surface normal $\mathbf{n}_i \in \mathbb{R}^3$. Specifically, we first place the source points $\{\mathbf{p}_{k,0}\}_{k=1}^{|\mathcal{T}|}$ of the given trajectory set $\mathcal{T}$ as *trajectory anchors*. We then fill the remaining anchor points with *sampled anchors* obtained by farthest point sampling seeded with the trajectory anchors. By construction, anchor $i \le |\mathcal{T}|$ corresponds to trajectory τ_i. For each anchor i, we obtain a per-anchor embedding $\mathbf{e}_i$ from a trajectory embedding $\mathcal{E}_{\text{traj}}(\tau_i)$ when i is a trajectory anchor ($i \le |\mathcal{T}|$), and from a learnable null embedding $\mathbf{e}_\varnothing$ otherwise. The conditioning token $\mathbf{c}_i$ for each anchor is then formed by concatenating the shape encoder's output $\mathcal{E}_{\text{shape}}(\mathbf{s}_i, \mathbf{n}_i)$ with $\mathbf{e}_i$, followed by an MLP:

$$\mathbf{c}_i = \mathrm{MLP}\big(\,\mathrm{concat}(\mathcal{E}_{\text{shape}}(\mathbf{s}_i, \mathbf{n}_i),\ \mathbf{e}_i)\big), \quad \text{where} \quad \mathbf{e}_i = \begin{cases} \mathcal{E}_{\text{traj}}(\tau_i) & \text{if } i \le |\mathcal{T}|, \\ \mathbf{e}_\varnothing & \text{otherwise.} \end{cases} \tag{3}$$

The resulting matrix $\mathbf{C}_{\text{traj}} \in \mathbb{R}^{N_{\text{anchor}}\times D}$ with feature dimension D provides consistent geometry-aware conditioning tokens that cover the entire shape and indicate which regions are guided by trajectories.

Table 1: Quantitative comparison with text-conditioned baselines. The *Cond.* column indicates the type of input conditions used by each method (text or trajectory). Our method outperforms the baselines [42, 47] across the user study, most VBench [17] metrics, and AMD [48], while remaining competitive on motion smoothness (MS).

Method	Cond.		User Study ↑			VBench ↑				Motion Mag. ↑
	Text	Traj	PA	MP	ME	OC	AQ	MS	DD	AMD
AnimateAnyMesh [47]	✓	✗	2.170	2.667	2.560	0.168	0.539	**0.992**	0.585	0.043
BiMotion [42]	✓	✗	2.657	3.040	2.813	0.170	0.536	0.991	0.329	0.046
Ours (text-only)	✓	✗	-	-	-	0.171	0.541	0.986	0.657	0.083
Ours	✓	✓	**4.507**	**4.067**	**4.340**	**0.175**	**0.544**	0.984	**0.743**	**0.147**

Generative architecture. Our generative model builds on DiT [31] and employs cross-attention to integrate the shape, trajectory and text conditions. For each DiT block, we compute keys and values from the shape tokens $\mathbf{C}_{\text{shape}}$ (via $\mathcal{E}_{\text{shape}}$), the trajectory tokens $\mathbf{C}_{\text{traj}}$ (via shape-grounded trajectory embedding), and text tokens $\mathbf{C}_{\text{text}}$ (via a CLIP text encoder [34]). We initialize the text and shape branches with weights from [42] and add the trajectory branch through a gated cross-attention [1] to smoothly incorporate trajectory guidance:

$$\begin{aligned} \mathbf{h} &\leftarrow \mathbf{h} + \tanh(\alpha) \cdot \mathrm{Attn}(\mathbf{Q}_{\mathbf{h}}, \mathbf{K}_{\text{traj}}, \mathbf{V}_{\text{traj}}), \\ \mathbf{Q}_{\mathbf{h}} = \mathbf{h}\mathbf{W}_q, \quad \mathbf{K}_{\text{traj}} &= \mathbf{C}_{\text{traj}}\mathbf{W}_k, \quad \mathbf{V}_{\text{traj}} = \mathbf{C}_{\text{traj}}\mathbf{W}_v, \end{aligned} \tag{4}$$

where $\mathbf{h}$ denotes the hidden state and $\alpha \in \mathbb{R}$ is a learnable scalar gate initialized to zero, so the backbone initially ignores trajectory input.

3.4 Training and inference pipeline

Training. We train the model in two stages, inspired by the conditional diffusion training schemes in prior work [44, 54, 59]. Stage one conditions on trajectories assigned to all N_{anchor} anchors, teaching the model to use dense guidance. Stage two progressively sparsifies the guidance by randomly sampling n active trajectories per sample, with the sampling range gradually reduced during training. We train the generative model using a rectified flow objective [7]. Given a noisy latent $\hat{\mathbf{z}}_s = (1-s)\,\mathbf{z}_0 + s\,\mathbf{z}_1$ interpolated from random noise $\mathbf{z}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and a clean latent $\mathbf{z}_1$ with a scalar $s \sim \mathcal{U}(0,1)$, the model predicts the velocity $\mathbf{v}_s = \mathbf{z}_1 - \mathbf{z}_0$ and minimizes:

$$\mathcal{L} = \mathbb{E}_{\mathbf{z}_0, \mathbf{z}_1, s, \mathbf{C}}\Big[\, \big\| \mathbf{v}_\theta(\hat{\mathbf{z}}_s,\, s,\, \mathbf{C}) - (\mathbf{z}_1 - \mathbf{z}_0) \big\|_2^2 \,\Big], \tag{5}$$

where $\mathbf{C} = \{\mathbf{C}_{\text{traj}}, \mathbf{C}_{\text{text}}, \mathbf{C}_{\text{shape}}\}$ denotes the conditioning tokens defined in Sec. 3.3.

Inference. At inference time, we solve the deterministic ODE associated with rectified flow. We apply classifier-free guidance (CFG) [13] only with respect to the text condition. Specifically, v_θ^{traj} is predicted while retaining the shape and trajectory conditions and zeroing only the text-token embeddings, whereas v_θ^{both} retains all three conditions. Thus, there is no fully unconditional branch. We compute the guided velocity as:

$$v_\theta^{\text{cfg}} = v_\theta^{\text{traj}} + \gamma \left(v_\theta^{\text{both}} - v_\theta^{\text{traj}} \right),$$

where we use $\gamma = 3.0$. We do not apply guidance to the trajectory branch because the trajectory explicitly specifies the desired motion magnitude, and amplifying this condition can lead to exaggerated motion.

4 Experiments

4.1 Experimental setup

Implementation details. We train a rectified flow-based generative model on the BiMO dataset, building on the pretrained VAE of [42]. Our model consists of 12 DiT blocks and is trained for 100K iterations at a learning rate of 5×10^{-5}, using the Adam optimizer [22] with weight decay

Table 2: Quantitative comparison with cascaded video-based baselines. Our method achieves substantially lower trajectory error in both 2D pixel and 3D spaces, and outperforms the cascaded baselines across the user study, most VBench [17] metrics, and AMD [48].

Method	Traj. Error ↓				User Study ↑			VBench ↑				Motion Mag. ↑
	ADE_{3D}	FDE_{3D}	ADE_{2D}	FDE_{2D}	PA	MP	ME	OC	AQ	MS	DD	AMD
Tora [59] + ActionMesh [37]	0.791	0.734	92.55	83.52	1.780	1.687	2.040	0.171	0.506	0.988	0.486	0.052
Tora [59] + Motion 3-to-4 [4]	0.605	0.675	69.19	77.64	1.940	1.800	1.903	0.166	0.522	**0.991**	0.486	0.042
Ours	**0.089**	**0.071**	**12.52**	**8.52**	**4.577**	**4.177**	**4.437**	**0.175**	**0.544**	0.984	**0.743**	**0.147**

1×10^{-2}. We linearly warm up the learning rate over the first $1K$ iterations and then decay it to 1×10^{-5} via scheduled annealing. Training is performed on four NVIDIA RTX 3090 GPUs in 3 days. For each object, we randomly sample 4096 points from the source mesh and further select 512 of them via farthest point sampling to be encoded into latent tokens by the VAE encoder. For the shape-grounded trajectory embedding, we use $N_{\text{anchor}} = 64$ anchor points. The text condition is encoded with CLIP ViT-L/14 [34], using a maximum token length of 77. For sampling, we use $\gamma = 3.0$ for classifier-free guidance [13]. To improve robustness against noisy and coarse input trajectories, we perturb each conditioning trajectory with random Gaussian noise, and apply two smoothing techniques: 1D Gaussian filtering and trajectory simplification. Further augmentation details are provided in Appendix B.

Baselines. We compare our model against text-conditioned baselines, AnimateAnyMesh [47] and BiMotion [42]. To further assess controllability, we additionally consider video-based baselines that incorporate trajectory conditioning indirectly by combining trajectory-guided video generation with video-based dynamic 3D generation. Specifically, we adopt Tora [59], a trajectory-conditioned video generator based on CogVideoX 5B [52] that accepts trajectories at arbitrary granularity, and pair it with two video-to-dynamic mesh generation models, ActionMesh [37] and Motion 3-to-4 [4].

Evaluation protocol. Since BiMO [42] provides no evaluation set for dynamic meshes, we conduct quantitative evaluation on 30 animated meshes sourced from Objaverse [6] and Sketchfab, a scale consistent with prior works [42, 47], which use 10 and 20 meshes, respectively. We assess the generated dynamic meshes along three perspectives: (i) *motion quality*, the visual fidelity of the generation; (ii) *motion expressiveness*, how dynamic the generated motion is; and (iii) *prompt alignment*, how faithfully the model follows the given text or trajectory condition. Following [18, 42, 47, 48], we evaluate these perspectives with VBench [17], reporting four sub-metrics: *aesthetic quality* (AQ) and *motion smoothness* (MS) for quality, *dynamic degree* (DD) for expressiveness, and *overall consistency* (OC) for prompt alignment. We additionally report *Average Motion Distance* (AMD) [48], which captures motion magnitude as the average inter-frame vertex displacement. We conduct a user study with 30 participants who rate the animations along three axes—*prompt alignment* (PA), *motion plausibility* (MP), and *motion expressiveness* (ME)—on a 5-point Likert scale ranging from 1 (*very poor*) to 5 (*excellent*) on 10 randomly chosen samples per baseline type, text-conditioned and video-generation-based baselines. For trajectory-conditioned baselines, we additionally report *Average Displacement Error* (ADE) and *Final Displacement Error* (FDE), which are the average ℓ_2 distance between each input trajectory and its corresponding generated vertices, taken over all frames and at the final frame, respectively. We measure these metrics in the 2D pixel space and 3D space for video-based baselines. All evaluations are run on 16-frame videos rendered from fixed viewpoints; for VBench, we average the metrics over videos from two side viewpoints to mitigate viewpoint-dependent variance such as occlusion. Importantly, to reflect realistic settings where dense trajectories are unlikely to be provided, all quantitative evaluations use sparse trajectory sets of at most five trajectories per object; to rule out cherry-picking, these are sampled by a predefined rule-based procedure without per-instance selection or human intervention. For qualitative examples, only the trajectories shown in the figures are used. Additional details are provided in Appendix C.

4.2 Comparisons

Quantitative comparison with text-based baselines. Tab. 1 summarizes the quantitative comparison between our model and the text-based generation baselines [42, 47]. Our method outperforms these baselines across the user study, most VBench [17] metrics, and AMD [48]. Since existing text-conditioned baselines do not accept 3D trajectories, this comparison is intended as a task-level

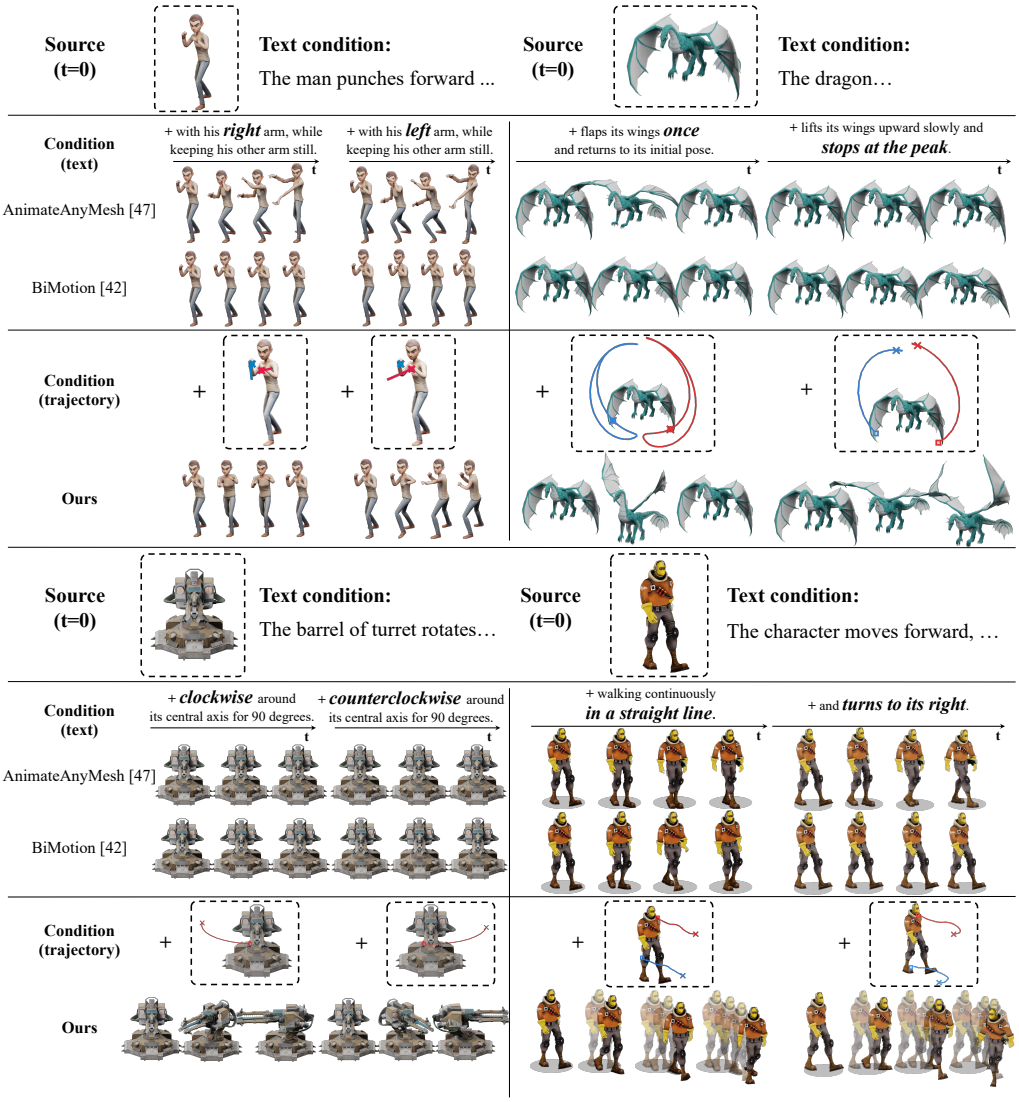

Figure 3: Qualitative comparison with text-based baselines. The baselines [42, 47] are conditioned on a pair of text prompts describing different specific motion details, while our model is guided by sparse trajectories representing those details. Our model uses only the visualized trajectories as inputs—at most two per example. By providing direct spatial cues, our method faithfully generates the intended motion, whereas the text-conditioned baselines fail to produce expressive motions aligned with the text and show little meaningful difference across the varying prompts.

Table 3: Component-level trajectory error analysis of the cascaded video-based baselines. Left (a): trajectory deviation introduced during video generation in the 2D image plane. Right (b): trajectory deviation introduced during video-to-3D reconstruction with ground-truth video as oracle input.

(a) Video generation stage (2D)

Method	ADE_{2D} ↓	FDE_{2D} ↓
Tora (raw video)	34.63	24.30
Ours (projected to 2D)	**12.52**	**8.52**

(b) Video-to-3D reconstruction stage

Method	ADE_{3D} ↓	FDE_{3D} ↓
ActionMesh (GT video, oracle)	0.324	0.267
Motion 3-to-4 (GT video, oracle)	0.303	0.218
Ours (direct 3D trajectory)	**0.089**	**0.071**

comparison of controllable generation systems; a matched-input text-only comparison is provided in Appendix C.1. These results indicate that, by providing direct cues for the intended motion through the input trajectories, motion generated by our method follows the text prompt (PA, OC)

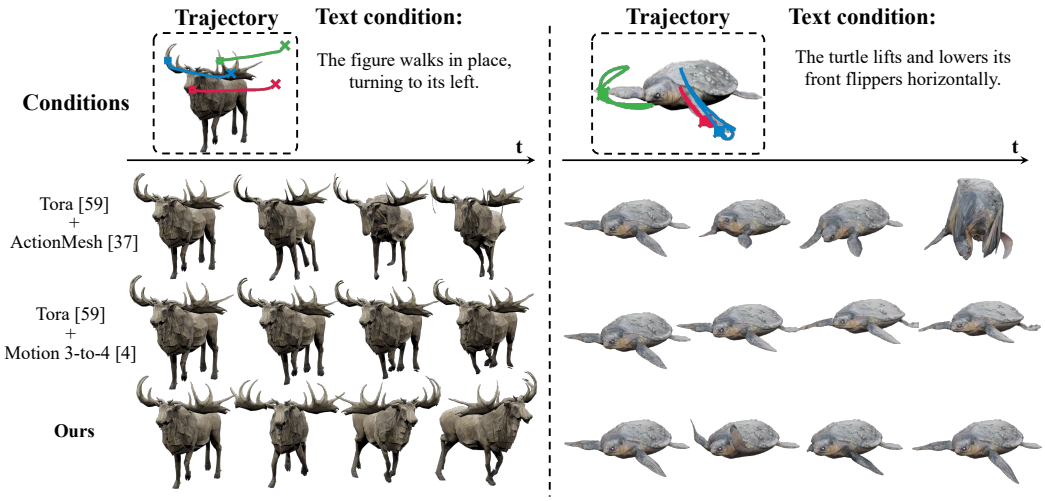

Figure 4: Qualitative comparison with video-based baselines. We compare our method against baselines that we construct by combining controllable video generation [59] with video-based dynamic 3D generation [4, 37]. These baselines take 2D trajectories projected onto the given camera view as input. The constructed pipelines fail to synthesize dynamic 3D shapes that faithfully follow the trajectory guidance and struggle to preserve the visual structure of the input shape during motion.

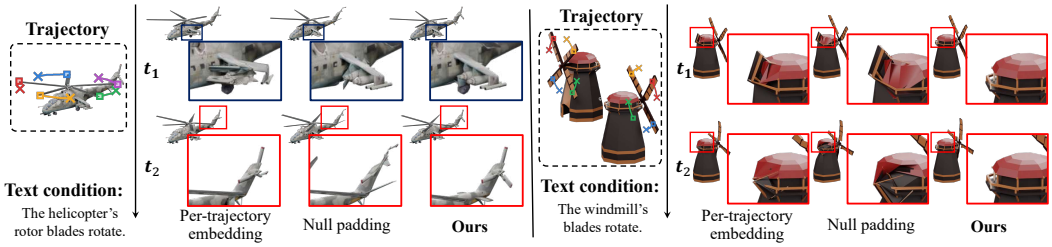

Figure 5: Qualitative comparison of trajectory embedding methods. We compare our shape-grounded trajectory embedding against two naive variants: per-trajectory embedding and null-padded per-trajectory embedding. With the naive variants, the trajectory conditions leak into unintended regions and produce noticeable artifacts, losing geometric locality. For visualization clarity, only the first 10% part of each trajectory is displayed.

more faithfully and produces higher quality motions (MP, AQ) and expressive motion (ME, DD, AMD) than the baselines. Although the baselines achieve slightly higher motion smoothness (MS) scores, MS rewards small frame-to-frame changes; together with the low ME in user study, AMD and dynamic degree (DD) scores of the baselines, this suggests that their MS scores stem from near-static or insufficiently dynamic motions.

Quantitative comparison with video-based baselines. Tab. 2 reports an end-to-end comparison with cascaded baselines constructed by combining trajectory-conditioned video generation [59] with video-to-dynamic-3D generation [4, 37]. Across VBench and AMD, these cascaded pipelines produce less expressive motion and exhibit substantially larger trajectory errors than our direct 3D trajectory-conditioned approach. We emphasize that Tab. 2 is an approach-level comparison: it evaluates the complete cascade that is required when using existing video-based methods, rather than attributing its error to a particular component. To disentangle the sources of this error, Tab. 3 separately analyzes the two stages of the cascade. In the video-generation stage, Tora already deviates from the projected input trajectories in the 2D image plane. In the subsequent reconstruction stage, ActionMesh and Motion 3-to-4 still exhibit non-negligible 3D trajectory errors even when provided with ground-truth videos as oracle inputs. These stage-wise errors are measured in their respective 2D and 3D spaces and are therefore not directly additive; rather, the analysis shows that deviation from the control signal can arise at both intermediate stages. T2Mo directly conditions on the input 3D trajectories and bypasses these intermediate representations.

Qualitative comparison with text-based baselines. We present qualitative results in Fig. 3 to compare the controllability, motion quality, and expressiveness of our generated motions against existing methods. To demonstrate the superior controllability of our trajectory-based approach over text-conditioned baselines [42, 47], we task the models with generating specific motions: the text-based baselines are provided with detailed text descriptions, while our model is guided by sparse trajectories. As shown in the Fig. 3, while the text-conditioned baselines struggle to realize finely specified motions from text prompts alone, our method successfully produces the intended motion by leveraging direct spatial guidance. This includes detail specification (top left), temporal control (top right), directional guidance (bottom left), and scene-level trajectory guidance (bottom right).

Qualitative comparison with video-based baselines. We further compare with the video-based baselines [4, 37, 59] in Fig. 4. Our method generates motion that reflects both the trajectory and text conditions, whereas the combined baselines neither preserve the visual quality of the input shape nor faithfully follow the given trajectory prompt.

Table 4: Ablation study on the trajectory embedding. Top: comparison of trajectory embedding variants. **Bottom**: effect of the number of anchor points N_{anchor}.

Method	Traj. Error ↓		VBench ↑			
	ADE_{3D}	FDE_{3D}	OC	AQ	MS	DD
Trajectory embedding strategy						
Per-traj. emb.	0.159	0.124	0.173	0.541	**0.985**	**0.771**
Null padding	0.101	0.080	0.174	0.540	0.984	0.743
Ours ($N_{\text{anchor}} = 64$)	**0.089**	**0.071**	**0.175**	**0.544**	0.984	0.743
Number of anchor points for training						
Ours ($N_{\text{anchor}} = 32$)	0.091	**0.070**	**0.176**	**0.548**	**0.984**	0.714
Ours ($N_{\text{anchor}} = 64$)	**0.089**	0.071	0.175	0.544	**0.984**	**0.743**
Ours ($N_{\text{anchor}} = 128$)	0.092	0.074	0.175	0.544	**0.984**	0.714
Ours ($N_{\text{anchor}} = 256$)	0.090	0.071	**0.176**	0.543	**0.984**	**0.743**

Table 5: Stress test on conditioning trajectories. *Downsampled*: uniformly subsampled to N waypoints. The first and last waypoints are preserved. *Perturbed*: Gaussian noise with standard deviation σ is added to each waypoint in the normalized $[-0.9, 0.9]$ coordinate space.

Trajectory Type	Traj. Error ↓		VBench ↑			
	ADE_{3D}	FDE_{3D}	OC	AQ	MS	DD
Original	**0.089**	**0.071**	**0.175**	**0.544**	0.984	**0.743**
Downsampled ($N = 8$)	0.093	0.080	0.174	**0.544**	0.985	0.729
Downsampled ($N = 4$)	0.121	0.090	0.174	0.543	**0.986**	0.700
Perturbed ($\sigma = 0.05$)	0.093	0.075	**0.175**	**0.544**	0.984	**0.743**
Perturbed ($\sigma = 0.1$)	0.100	0.085	0.174	0.542	0.984	**0.743**

4.3 Ablation study and analysis

In this section, we analyze the roles of the conditioning modalities and key design choices of T2Mo, as well as its robustness to imperfect trajectory inputs. We first examine the complementary effects of text and trajectory conditioning, then evaluate the proposed shape-grounded trajectory embedding and its sensitivity to the number of anchor points, and finally stress-test the model with sparse and noisy trajectories. Additional controlled user studies and evaluations with user-drawn trajectories, alternative control-point configurations, source-point perturbations, missing trajectory segments are provided in Appendix C.1.

Effect of conditioning modalities. We further analyze the complementary roles of text and trajectory conditions in Table 6. With trajectory conditioning, both the full model and the trajectory-only variant closely follow the prescribed paths, indicating that trajectories provide the primary spatial control signal. In contrast, removing trajectory conditioning substantially increases the trajectory error and reduces motion magnitude (AMD), despite preserving comparable appearance-quality metrics. Text conditioning mainly affects the unconstrained regions of the shape: while the trajectory-only variant achieves strong trajectory adherence, it shows slightly lower object consistency and aesthetic quality, suggesting that text provides semantic context that helps regularize motion beyond the explicitly guided regions. These results support the complementary use of trajectories for spatial control and text for semantic motion guidance.

Table 6: Ablation of conditioning modalities.

Condition	Traj. Error ↓		VBench ↑				AMD ↑
	ADE_{3D}	FDE_{3D}	OC	AQ	MS	DD	
Text + Trajectory	0.089	**0.071**	**0.175**	**0.544**	0.984	0.743	0.147
Trajectory only	**0.087**	0.073	0.172	0.531	0.981	**0.829**	**0.151**
Text only	0.471	0.372	0.171	0.541	**0.986**	0.657	0.083

Trajectory embedding method. To validate our shape-grounded trajectory embedding method, we compare it against two variants: (i) *per-trajectory embedding*, which directly tokenizes each input trajectory and produces a variable number of condition tokens; and (ii) *per-trajectory embedding with null padding*, which pads the null tokens to a fixed size to remove the effect of token sparsity.

As shown in the top section of Tab. 4, our shape-grounded trajectory embedding achieves the lowest trajectory alignment error. It also obtains slightly higher VBench scores in aesthetic quality (AQ) and overall consistency (OC). The per-trajectory variant without padding obtains a higher dynamic degree (DD); however, this comes alongside worse trajectory alignment error, and the qualitative comparisons in Fig. 5 show that the additional motion often appears in regions not specified by the input trajectories. This suggests that, when a spatially inconsistent and unevenly distributed trajectory set is naively injected, the model struggles to provide localized control over the intended regions. While this phenomenon is rarely observed in standard video generation with similar formulation, it emerges as a critical challenge in dynamic 3D generation. We suggest that this is because trajectory conditioning in video generation operates within a fixed 2D pixel space, where the background inherently provides dense, global guidance. Specifically, the background explicitly indicates unconditioned regions, and all trajectories are naturally spatially aligned within this grid. In our 3D setting, however, the spatial domain is highly shape-dependent, and the sparse input trajectories alone cannot provide sufficient global coverage across the entire surface.

Effect of the number of anchor points. We further conduct an ablation study on the number of anchor points to examine its effect on performance. We report the results of varying N_{anchor} from 32 to 256 in the second section of Tab. 4. The performance remains stable across this range, indicating that our shape-grounded trajectory embedding is robust to the choice of N_{anchor}. We set $N_{\text{anchor}} = 64$ in all experiments as it achieves strong trajectory alignment with comparable VBench scores.

Trajectory quality stress test. In real-world settings, user-provided trajectories may be noisy or coarse compared to the ground-truth trajectories used during training. To evaluate the robustness of our model against such imperfect inputs, we test two perturbations: (i) *downsampled* trajectories, obtained by uniformly sampling N waypoints ($N < 16$) while preserving the first and last waypoints; and (ii) *perturbed* trajectories, obtained by adding Gaussian noise with a standard deviation of σ to each waypoint (within a $[-0.9, 0.9]$ normalized coordinate space). In both cases, trajectory errors are computed against the original ground-truth trajectories. As reported in Tab. 5, while trajectory errors (ADE, FDE) inevitably increase, this degradation is marginal, and the video-based VBench metrics remain largely preserved. This indicates that the model robustly synthesizes the intended motion even when conditioned on coarse or moderately noisy trajectories.

Inference efficiency. We additionally evaluate the inference time and peak memory usage of T2Mo across meshes with different vertex counts. As reported in Table 7, T2Mo introduces only a modest computational overhead over the text-conditioned BiMotion baseline. For example, at 5K vertices, T2Mo requires 2.16 seconds and 1.22 GiB of peak GPU memory, compared with 2.00 seconds and 1.10 GiB for BiMotion. Even at 15K vertices, inference requires 5.98 seconds and 1.35 GiB. All measurements use 64 sampling steps and are averaged over three runs under the same hardware and inference configuration. These results indicate that the additional trajectory-conditioning branch incurs relatively small computational overhead.

Table 7: Inference time and peak VRAM across different mesh resolutions. All methods use 64 sampling steps, and results are averaged over three runs on an NVIDIA H100 GPU.

# Vertices	AnimateAnyMesh	BiMotion	Ours
512	1.463s / 1.054 GiB	1.666s / 1.070 GiB	1.924s / 1.179 GiB
2K	1.644s / 1.054 GiB	1.908s / 1.072 GiB	2.030s / 1.182 GiB
5K	2.365s / 1.199 GiB	1.995s / 1.098 GiB	2.164s / 1.219 GiB
15K	9.677s / 2.784 GiB	5.505s / 1.223 GiB	5.976s / 1.348 GiB

5 Conclusion

We presented T2Mo, a generative framework for controllable dynamic 3D shape generation conditioned on 3D trajectories and a text prompt. The key component is a shape-grounded trajectory embedding that maps an arbitrarily configured trajectory set into a predefined shape-aware token set covering the entire shape, which enables stable conditioning across varying trajectory configurations. Across both quantitative and qualitative evaluations, our framework produces motion that more faithfully reflects the given prompts than the text-conditioned and video-based cascaded baselines, while supporting various motion controls and applications. We believe this approach makes a practical step toward enabling intuitive and expressive control in dynamic 3D shape generation.

Acknowledgements

This research was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2019-II190075, RS-2024-00509279, RS-2025-II212068, RS2023-00227592, RS-2025-02214479, RS-2024-00457882, RS-2025-25441838, RS2025-02217259, RS-2026-25519202, RS-2024-00436680) and the Culture, Sports, and Tourism RD Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports and Tourism (RS-2024-00345025, RS-2024-00333068), the InnoCORE program of the Ministry of Science and ICT (N10250156) and National Research Foundation of Korea (RS-2024-00346597).

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

[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] Sherwin Bahmani, Xian Liu, Wang Yifan, Ivan Skorokhodov, Victor Rong, Ziwei Liu, Xihui Liu, Jeong Joon Park, Sergey Tulyakov, Gordon Wetzstein, Andrea Tagliasacchi, and David B. Lindell. Tc4d: Trajectory-conditioned text-to-4d generation. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision – ECCV 2024*, pages 53–72, Cham, 2025. Springer Nature Switzerland.

[4] Hongyuan Chen, Xingyu Chen, Youjia Zhang, Zexiang Xu, and Anpei Chen. Motion 3-to-4: 3d motion reconstruction for 4d synthesis. *arXiv preprint arXiv:2601.14253*, 2026.

[5] Ruihang Chu, Yefei He, Zhekai Chen, Shiwei Zhang, Xiaogang Xu, Bin Xia, Dingdong Wang, Hongwei Yi, Xihui Liu, Hengshuang Zhao, et al. Wan-move: Motion-controllable video generation via latent trajectory guidance. *arXiv preprint arXiv:2512.08765*, 2025.

[6] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13142–13153, 2023.

[7] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024. URL `https://openreview.net/forum?id=FPnUhsQJ5B`.

[8] Xiao Fu, Xian Liu, Xintao Wang, Sida Peng, Menghan Xia, Xiaoyu Shi, Ziyang Yuan, Pengfei Wan, Di Zhang, and Dahua Lin. 3dtrajmaster: Mastering 3d trajectory for multi-entity motion in video generation. *arXiv preprint arXiv:2412.07759*, 2024.

[9] Junyao Gao, Jiaxing Li, Wenran Liu, Yanhong Zeng, Fei Shen, Kai Chen, Yanan Sun, and Cairong Zhao. Charactershot: Controllable and consistent 4d character animation. *arXiv preprint arXiv:2508.07409*, 2025.

[10] Daniel Geng, Charles Herrmann, Junhwa Hur, Forrester Cole, Serena Zhang, Tobias Pfaff, Tatiana Lopez-Guevara, Yusuf Aytar, Michael Rubinstein, Chen Sun, Oliver Wang, Andrew Owens, and Deqing Sun. Motion prompting: Controlling video generation with motion trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–12, June 2025.

[11] Ziyan Guo, Zeyu Hu, De Wen Soh, and Na Zhao. Motionlab: Unified human motion generation and editing via the motion-condition-motion paradigm. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13869–13879, 2025.

[12] Junlin Han, Jianyuan Wang, Andrea Vedaldi, Philip Torr, and Filippos Kokkinos. Flex3D: Feed-forward 3D generation with flexible reconstruction model and input view curation. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267

of *Proceedings of Machine Learning Research*, pages 21898–21915. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/han25i.html.

[13] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[14] Susung Hong, Donghoon Ahn, and Seungryong Kim. Debiasing scores and prompts of 2d diffusion for view-consistent text-to-3d generation. *Advances in Neural Information Processing Systems*, 36: 11970–11987, 2023.

[15] Hanzhuo Huang, Yuan Liu, Ge Zheng, Jiepeng Wang, Zhiyang Dou, and Sibei Yang. Mvtokenflow: High-quality 4d content generation using multiview token flow. *arXiv preprint arXiv:2502.11697*, 2025.

[16] Zehuan Huang, Haoran Feng, Yang-Tian Sun, Yuan-Chen Guo, Yan-Pei Cao, and Lu Sheng. Animax: Animating the inanimate in 3d with joint video-pose diffusion models. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pages 1–13, 2025.

[17] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21807–21818, 2024.

[18] 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, 2024.

[19] Zeren Jiang, Chuanxia Zheng, Iro Laina, Diane Larlus, and Andrea Vedaldi. Mesh4d: 4d mesh reconstruction and tracking from monocular video. *arXiv preprint arXiv:2601.05251*, 2026.

[20] Kacper Kania, Marek Kowalski, et al. Trajevae: Controllable human motion generation from trajectories. *arXiv preprint arXiv:2104.00351*, 2021.

[21] Korrawe Karunratanakul, Konpat Preechakul, Supasorn Suwajanakorn, and Siyu Tang. Guided motion diffusion for controllable human motion synthesis. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2151–2162, 2023.

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[23] Zeqiang Lai, Yunfei Zhao, Haolin Liu, Zibo Zhao, Qingxiang Lin, Huiwen Shi, Xianghui Yang, Mingxin Yang, Shuhui Yang, Yifei Feng, et al. Hunyuan3d 2.5: Towards high-fidelity 3d assets generation with ultimate details. *arXiv preprint arXiv:2506.16504*, 2025.

[24] Weiyu Li, Jiarui Liu, Hongyu Yan, Rui Chen, Yixun Liang, Xuelin Chen, Ping Tan, and Xiaoxiao Long. Craftsman3d: High-fidelity mesh generation with 3d native diffusion and interactive geometry refiner. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5307–5317, 2025.

[25] Weiyu Li, Xuanyang Zhang, Zheng Sun, Di Qi, Hao Li, Wei Cheng, Weiwei Cai, Shihao Wu, Jiarui Liu, Zihao Wang, et al. Step1x-3d: Towards high-fidelity and controllable generation of textured 3d assets. *arXiv preprint arXiv:2505.07747*, 2025.

[26] Xuan Li, Qianli Ma, Tsung-Yi Lin, Yongxin Chen, Chenfanfu Jiang, Ming-Yu Liu, and Donglai Xiang. Articulated kinematics distillation from video diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 17571–17581, 2025.

[27] Yangguang Li, Zi-Xin Zou, Zexiang Liu, Dehu Wang, Yuan Liang, Zhipeng Yu, Xingchao Liu, Yuan-Chen Guo, Ding Liang, Wanli Ouyang, et al. Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models. *arXiv preprint arXiv:2502.06608*, 2025.

[28] Zhiqi Li, Yiming Chen, and Peidong Liu. Dreammesh4d: Video-to-4d generation with sparse-controlled gaussian-mesh hybrid representation. *Advances in Neural Information Processing Systems*, 37:21377–21400, 2024.

[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*, pages 8576–8588, 2024.

[30] Wan-Duo Kurt Ma, John P Lewis, and W Bastiaan Kleijn. Trailblazer: Trajectory control for diffusion-based video generation. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024.

[31] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

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

[33] Hao Qin, Luyuan Chen, Ming Kong, Mengxu Lu, and Qiang Zhu. Distilling multi-view diffusion models into 3d generators. *arXiv preprint arXiv:2504.00457*, 2025.

[34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/radford21a.html`.

[35] Davis Rempe, Mathis Petrovich, Ye Yuan, Haotian Zhang, Xue Bin Peng, Yifeng Jiang, Tingwu Wang, Umar Iqbal, David Minor, Michael de Ruyter, et al. Kimodo: Scaling controllable human motion generation. *arXiv preprint arXiv:2603.15546*, 2026.

[36] Jiawei Ren, Kevin Xie, Ashkan Mirzaei, Hanxue Liang, Xiaohui Zeng, Karsten Kreis, Ziwei Liu, Antonio Torralba, Sanja Fidler, Seung W Kim, et al. L4gm: Large 4d gaussian reconstruction model. *Advances in Neural Information Processing Systems*, 37:56828–56858, 2024.

[37] Remy Sabathier, David Novotny, Niloy J Mitra, and Tom Monnier. Actionmesh: Animated 3d mesh generation with temporal 3d diffusion. *arXiv preprint arXiv:2601.16148*, 2026.

[38] Qi Sun, Zhiyang Guo, Ziyu Wan, Jing Nathan Yan, Shengming Yin, Wengang Zhou, Jing Liao, and Houqiang Li. Eg4d: Explicit generation of 4d object without score distillation. *arXiv preprint arXiv:2405.18132*, 2024.

[39] Weilin Wan, Zhiyang Dou, Taku Komura, Wenping Wang, Dinesh Jayaraman, and Lingjie Liu. Tlcontrol: Trajectory and language control for human motion synthesis. In *European Conference on Computer Vision*, pages 37–54. Springer, 2024.

[40] Hanlin Wang, Hao Ouyang, Qiuyu Wang, Wen Wang, Ka Leong Cheng, Qifeng Chen, Yujun Shen, and Limin Wang. Levitor: 3d trajectory oriented image-to-video synthesis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12490–12500, 2025.

[41] Jiawei Wang, Yuchen Zhang, Jiaxin Zou, Yan Zeng, Guoqiang Wei, Liping Yuan, and Hang Li. Boximator: Generating rich and controllable motions for video synthesis. *arXiv preprint arXiv:2402.01566*, 2024.

[42] Miaowei Wang, Qingxuan Yan, Zhi Cao, Yayuan Li, Oisin Mac Aodha, Jason J Corso, and Amir Vaxman. Bimotion: B-spline motion for text-guided dynamic 3d character generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10152–10164, 2026.

[43] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in neural information processing systems*, 36:8406–8441, 2023.

[44] Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024.

[45] Rundi Wu, Ruiqi Gao, Ben Poole, Alex Trevithick, Changxi Zheng, Jonathan T Barron, and Aleksander Holynski. Cat4d: Create anything in 4d with multi-view video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26057–26068, 2025.

[46] Weijia Wu, Zhuang Li, Yuchao Gu, Rui Zhao, Yefei He, David Junhao Zhang, Mike Zheng Shou, Yan Li, Tingting Gao, and Di Zhang. Draganything: Motion control for anything using entity representation. In *European Conference on Computer Vision*, pages 331–348. Springer, 2024.

[47] Zijie Wu, Chaohui Yu, Fan Wang, and Xiang Bai. Animateanymesh: A feed-forward 4d foundation model for text-driven universal mesh animation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13557–13568, October 2025.

[48] Zijie Wu, Chaohui Yu, Fan Wang, and Xiang Bai. Animateanymesh++: A flexible 4d foundation model for high-fidelity text-driven mesh animation, 2026. URL `https://arxiv.org/abs/2604.26917`.

[49] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21469–21480, 2025.

[50] Kevin Xie, Amirmojtaba Sabour, Jiahui Huang, Despoina Paschalidou, Greg Klar, Umar Iqbal, Sanja Fidler, and Xiaohui Zeng. Videopanda: Video panoramic diffusion with multi-view attention. *arXiv preprint arXiv:2504.11389*, 2025.

[51] Yiming Xie, Varun Jampani, Lei Zhong, Deqing Sun, and Huaizu Jiang. Omnicontrol: Control any joint at any time for human motion generation. In *International Conference on Learning Representations*, volume 2024, pages 28176–28194, 2024.

[52] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

[53] Jiraphon Yenphraphai, Ashkan Mirzaei, Jianqi Chen, Jiaxu Zou, Sergey Tulyakov, Raymond A Yeh, Peter Wonka, and Chaoyang Wang. Shapegen4d: Towards high quality 4d shape generation from videos. *arXiv preprint arXiv:2510.06208*, 2025.

[54] Shengming Yin, Chenfei Wu, Jian Liang, Jie Shi, Houqiang Li, Gong Ming, and Nan Duan. Dragnuwa: Fine-grained control in video generation by integrating text, image, and trajectory. *arXiv preprint arXiv:2308.08089*, 2023.

[55] Heng Yu, Chaoyang Wang, Peiye Zhuang, Willi Menapace, Aliaksandr Siarohin, Junli Cao, Laszlo A 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, 2024.

[56] Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Transactions On Graphics (TOG)*, 42(4):1–16, 2023.

[57] Haiyu Zhang, Xinyuan Chen, Yaohui Wang, Xihui Liu, Yunhong Wang, and Yu Qiao. 4diffusion: Multi-view video diffusion model for 4d generation. *Advances in Neural Information Processing Systems*, 37: 15272–15295, 2024.

[58] Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. Clay: A controllable large-scale generative model for creating high-quality 3d assets. *ACM Transactions on Graphics (TOG)*, 43(4):1–20, 2024.

[59] Zhenghao Zhang, Junchao Liao, Menghao Li, ZuoZhuo Dai, Bingxue Qiu, Siyu Zhu, Long Qin, and Weizhi Wang. Tora: Trajectory-oriented diffusion transformer for video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2063–2073, June 2025.

Appendix

A Implementation details

Training details. Inspired by previous work on trajectory-guided generation [44, 54, 59], we adopt a two-stage training strategy while fixing $N_{\text{anchor}} = 64$ throughout training. In the first stage, lasting 25K iterations, all 64 anchors are assigned active trajectories, allowing the model to first learn to exploit dense trajectory guidance. In the second stage, we progressively sparsify the trajectory condition over the remaining 75K iterations. During the first 37.5K iterations of this stage, the number of active trajectories n is sampled uniformly from $\{2, \dots, n_{\max}\}$, where $n_{\max}$ is linearly annealed from 32 to 2. During the final 37.5K iterations, we fix $n = 2$. Anchors without an active trajectory receive the learnable null embedding $\mathbf{e}_\emptyset$. To encourage the model to utilize informative trajectory guidance, a fixed proportion $\rho = 0.7$ of the sampled trajectories is drawn from the top 30% of vertices ranked by motion magnitude, while the remainder is sampled uniformly from all vertices.

Architecture details. We build on the pretrained B-spline-based VAE of [42] and reuse the same VAE to embed the trajectory conditions for efficient training. The weights of the VAE encoder, including its two streams $\mathcal{E}_{\text{shape}}$ and $\mathcal{E}_{\text{traj}}$, are not updated during trajectory conditioning. We follow the original implementation for all transformations involved in the B-spline-based representation. Specifically, the cross-attention weight sharing in the original implementation is retained when computing the latent embeddings $\mathbf{z}_{\text{shape}}$ and $\mathbf{z}_{\text{traj}}$, but is omitted when encoding the trajectory condition to reduce the influence of the spatial configuration of the conditioning trajectories and the dilution introduced by the null embeddings. With this cross-attention sharing disabled, the only remaining cross-trajectory interaction is removed, so the trajectory stream processes the conditioning trajectories in parallel without any coupling between them; encoding them jointly is therefore equivalent to encoding each one independently, which makes the per-trajectory embedding $\mathcal{E}_{\text{traj}}(\tau_i)$ in Eq. 3 well-defined regardless of how many conditioning trajectories are provided. The shape stream, in turn, encodes the input shape as in the original VAE, independently of the conditioning trajectories and anchors; the per-anchor shape feature $\mathcal{E}_{\text{shape}}(s_i, n_i)$ denotes the resulting shape feature read out at the anchor location (s_i, n_i). For brevity, we omit the cross-attention weight sharing in our notation. AdaLN [31] conditioned on diffusion timestep is applied independently to the noisy latent $\hat{\mathbf{z}}_{\text{traj}}$ and each conditioning token.

Anchored farthest point sampling. Our shape-grounded trajectory embedding requires a fixed set of N_{anchor} anchor points that (i) include the starting positions of all user-provided trajectories so that each trajectory is grounded at its intended location on the shape, and (ii) cover the rest of the shape to provide geometric context for unconditioned regions. We achieve both with an anchored variant of farthest point sampling. Given a trajectory set $\mathcal{T} = \{\tau_k\}_{k=1}^{|\mathcal{T}|}$, we first take the starting waypoints $\{\mathbf{p}_{k,0}\}_{k=1}^{|\mathcal{T}|}$ as the trajectory anchors and include them as the first $|\mathcal{T}|$ entries of the anchor set. The remaining $N_{\text{anchor}} - |\mathcal{T}|$ anchors are then selected from the mesh vertices $\mathbf{V}_0$ by farthest point sampling seeded with the trajectory anchors, so that newly added anchors are progressively chosen to maximize the minimum distance to the existing anchor set. During training, each trajectory anchor coincides with a mesh vertex; at inference, when a user-provided trajectory does not start exactly on a vertex, its starting position is snapped to the nearest point on the mesh surface before serving as an anchor. This procedure guarantees that the trajectory anchors are preserved exactly while the remaining anchors evenly cover the shape.

Input normalization. To reduce sensitivity to the raw mesh resolution and scale, each mesh is represented by 4,096 sampled surface points, from which 512 latent tokens are selected by farthest point sampling. The mesh and its trajectories are jointly normalized by centering the bounding box and scaling its longest side to 1.8, resulting in coordinates within $[-0.9, 0.9]^3$. The resulting model input size is therefore independent of the original mesh vertex count.

B Data augmentation

To improve robustness and encourage the model to actively utilize trajectory information, we apply two categories of data augmentations. All augmentations described below are applied only when

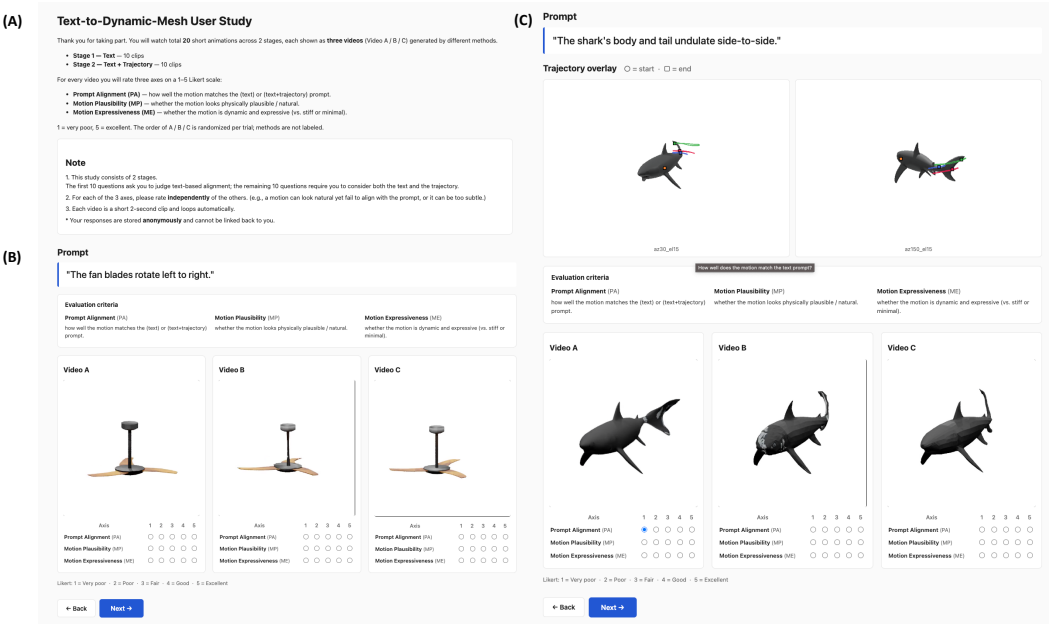

Figure 6: User study interface. (A) Initial instruction page describing the rating task and evaluation criteria. **(B)** Rating page for text-conditioned baselines, where participants rate three videos generated by different methods given a text prompt. **(C)** Rating page for cascaded video-based baselines, where participants additionally see the input trajectory overlay alongside the text prompt. Videos from compared methods are presented side-by-side in a randomly shuffled order.

Table 8: User study results. Each cell shows the mean ± standard deviation over all participants and examples. *Left*: text-based baselines [42, 47]. *Right*: video-based baselines [4, 37, 59].

Methods	PA↑	MP↑	ME↑
AnimateAnyMesh [47]	2.170±1.260	2.667±1.452	2.560±1.243
BiMotion [42]	2.657±1.525	3.040±1.536	2.813±1.499
Ours	**4.507±0.782**	**4.067±1.055**	**4.340±0.880**

Methods	PA↑	MP↑	ME↑
ActionMesh [37]	1.780±1.004	1.687±1.042	2.040±1.053
Motion 3-to-4 [4]	1.940±1.071	1.800±1.025	1.903±0.944
Ours	**4.577±0.657**	**4.177±0.921**	**4.437±0.775**

trajectories are provided as conditions, since they perturb the data distribution that text alone could otherwise capture and may degrade text-conditioned generation. First, we apply global geometric and temporal augmentations. We inject random noise into the input shape (positions and normals) as well as the target and conditioning trajectories, alongside random horizontal rotation and flipping. Additionally, we apply temporal cropping that retains at least 50% of the full sequence length to preserve the original motion semantics. These transformations not only enhance overall robustness but also encourage the model to resolve spatial ambiguities—which cannot be inferred from text alone—by relying on the trajectory guidance. Second, to ensure the model is robust to the coarse and noisy trajectories typically provided by users, we apply targeted augmentations exclusively to the conditioning trajectories. This involves injecting Gaussian noise, followed by two smoothing techniques: a simplification step that subsamples a random subset of intermediate waypoints while retaining the endpoints, and 1D Gaussian-kernel smoothing applied along the temporal axis with a randomly chosen kernel width, which suppresses high-frequency jitter and yields smoother trajectories closer to those a user would draw.

C Evaluation details

Evaluation metrics. We represent each generated motion as a sequence of T deformed meshes. Let $\mathbf{V}_t \in \mathbb{R}^{N\times 3}$ denote the vertex positions of the generated mesh at frame $t \in \{1, \ldots, T\}$, where N is the number of vertices and $(\mathbf{V}_t)_i$ is the 3D position of vertex i.

For evaluation, we are given a mesh with motion together with K target trajectories used as conditioning. Each trajectory has its source point on the mesh surface, where the source point is

included in mesh vertices, i.e., $\mathbf{p}_{k,0} \in \mathbf{V}_0$. We denote the corresponding vertex index by i_k, so that $(\mathbf{V}_0)_{i_k} = \mathbf{p}_{k,0}$. The generated vertex $(\mathbf{V}_t)_{i_k}$ is then compared with the target waypoint $\mathbf{p}_{k,t}$ at each frame.

Average Motion Distance (AMD) measures the average inter-frame displacement over all vertices:

$$\text{AMD} = \frac{1}{T-1} \sum_{t=1}^{T-1} \frac{1}{N} \sum_{i=1}^{N} \|(\mathbf{V}_{t+1})_i - (\mathbf{V}_t)_i\|_2 \,. \tag{6}$$

AMD measures the overall amount of generated motion, not trajectory-tracking accuracy. It is computed in 3D world coordinates and reported in the same units as the mesh.

Average Displacement Error (ADE) and *Final Displacement Error (FDE)* measure trajectory-tracking accuracy for the K controlled vertices. For the k-th trajectory, we define

$$\text{ADE}_k = \frac{1}{T} \sum_{t=1}^{T} \|(\mathbf{V}_t)_{i_k} - \mathbf{p}_{k,t}\|_2 \,, \qquad \text{FDE}_k = \|(\mathbf{V}_T)_{i_k} - \mathbf{p}_{k,T}\|_2 \,. \tag{7}$$

The final ADE and FDE are obtained by averaging over all input trajectories:

$$\text{ADE} = \frac{1}{K} \sum_{k=1}^{K} \text{ADE}_k, \qquad \text{FDE} = \frac{1}{K} \sum_{k=1}^{K} \text{FDE}_k. \tag{8}$$

For the ADE_{3D} and FDE_{3D}, the above distances are computed directly in world coordinates. For the 2D variant, both the generated vertex and the target waypoint are projected to the image plane using the evaluation camera projection $\pi : \mathbb{R}^3 \to \mathbb{R}^2$:

$$\text{ADE}_{\text{2D},k} = \frac{1}{T} \sum_{t=1}^{T} \|\pi((\mathbf{V}_t)_{i_k}) - \pi(\mathbf{p}_{k,t})\|_2 \,, \qquad \text{FDE}_{\text{2D},k} = \|\pi((\mathbf{V}_T)_{i_k}) - \pi(\mathbf{p}_{k,T})\|_2 \,. \tag{9}$$

The 2D errors are reported in pixels and are averaged over the fixed evaluation viewpoints.

Condition trajectory selection. For quantitative evaluation, we obtain the input trajectories for our model automatically, without human intervention, by sampling from the per-vertex ground-truth trajectories using a two-stage predefined rule-based procedure. This process is performed to avoid bias introduced by subjective trajectory selection. In the first stage, we score each vertex by its cumulative path length over the sequence and draw $N_{\text{cand}} = 64$ candidates from the top-30% high-motion vertices, with sampling probabilities biased toward higher-motion vertices by a power law with exponent $\beta = 2$. In the second stage, the candidates are clustered using K-Means on a feature composed of each vertex's source position and its per-frame displacements. The number of clusters $K_c \in \{3, 4, 5\}$ is selected by the silhouette score, and the trajectory closest to each cluster centroid is retained as the representative. The resulting set $\mathcal{R}$, with $|\mathcal{R}| = K_c \leq 5$, is used as the trajectory input for quantitative evaluation. For qualitative figures, we use the same automatic procedure unless otherwise specified. When specified trajectories are used, we explicitly visualize the conditioning trajectories in the figure. The average number of conditioning trajectories is 4.01 in quantitative evaluation, and no hyperparameter search was performed to optimize the evaluation results.

User study protocol. Since subtle aspects of motion are difficult to evaluate with automatic pipelines, we conduct an anonymous user study to obtain human perceptual judgments. The study involves 30 participants, each rating the generated animations along three axes: *Prompt Alignment* (PA), *Motion Plausibility* (MP), and *Motion Expressiveness* (ME). Ratings are reported on a 5-point Likert scale, where 1 denotes *very poor* and 5 denotes *excellent*. We evaluate against two groups of baselines: text-conditioned baselines [42, 47] that take only text as input, and cascaded video-based baselines [4, 37, 59] that take both text and trajectories. For each group, we independently sample 10 objects at random from the evaluation set. For the text-based comparison, prompt alignment (PA) is assessed against the text prompt. For the video-based baselines, PA is assessed against both the trajectory condition and the text prompt, since these models receive both conditions. In total, each participant answers $(10 + 10) \times 3 \times 3 = 180$ questions, covering 20 objects, three methods, and three rating axes. As shown in Fig. 6, participants are first presented with the rating instructions,

Table 9: Additional controlled user studies. Both studies use 20 participants and 10 examples and follow the same perceptual evaluation protocol as the main user study. The two panels correspond to independent studies and their absolute scores should not be compared across panels.

(a) Matched-input text-only comparison

Method	PA ↑	MP ↑	ME ↑
AnimateAnyMesh	1.92	2.20	2.30
BiMotion	2.36	2.56	2.38
Ours (text-only)	**3.70**	**3.84**	**3.85**

(b) Trajectory embedding comparison

Method	PA ↑	MP ↑	ME ↑
Per-traj. embedding	3.57	2.82	3.34
Null padding	3.93	3.33	3.81
Ours (SGTE)	**4.42**	**4.06**	**4.30**

Table 10: Evaluation with user-drawn trajectories on the same five-object subset. ADE/FDE are measured against the respective input trajectories.

Method	OC ↑	AQ ↑	MS ↑	DD ↑	AMD ↑	PA ↑	MP ↑	ME ↑	ADE_{3D} ↓	FDE_{3D} ↓
GT-sampled	0.201	0.534	0.988	0.400	0.130	4.53	4.39	4.49	0.110	0.048
User-drawn	0.199	0.534	0.989	0.627	0.127	4.54	4.15	4.13	0.111	0.119

after which they rate each object using a video and the corresponding prompt. For each object, the videos from all compared methods are presented side-by-side in a randomly shuffled order so that participants are blind to the source of each video, and each video is automatically looped. A consolidated summary of the user study results, including standard deviations, is reported in Tab. 8.

Video generation procedure. To evaluate the cascaded video-based baselines which combines trajectory-conditioned video generation model with video-to-dynamic mesh generation model, we first generate trajectory-conditioned videos using Tora [59]. We choose Tora as the video-generation baseline because it is a publicly available trajectory-conditioned video generator that supports arbitrary point-level 2D trajectory control, rather than relying on object-level controls. Since Tora operates in the image plane, we project the input 3D trajectories into 2D pixel coordinates using the same camera viewpoint as the rendered videos used for VBench [17]. To avoid degraded video quality caused by occluded trajectories, we discard any trajectory whose source vertex is occluded in the first frame; occlusion is determined by comparing the projected depth of each vertex with the depth value at the corresponding pixel. The resulting videos, together with the ground-truth source mesh, are then passed to the video-to-dynamic-3D mesh generation pipelines [4, 37], which produce the final dynamic mesh sequence used for evaluation. All baselines are run with their publicly available models and default configurations.

C.1 Additional Experiments

Additional controlled user studies. We conduct two additional controlled user studies to complement the main 30-participant evaluation: (i) a matched-input comparison with text-conditioned baselines and (ii) a comparison of trajectory embedding strategies. Each study includes 20 participants and 10 examples and follows the same perceptual evaluation protocol as the main study, evaluating prompt alignment (PA), motion plausibility (MP), and motion expressiveness (ME). These studies were conducted independently from the main user study and from each other; therefore, absolute rating values should not be compared across studies. The results are summarized in Table 9.

Matched-input text-only comparison. The comparison with text-conditioned baselines in the main paper is intended as a task-level comparison, where our full model additionally receives 3D trajectory guidance. To provide a controlled comparison under matched inference inputs, we evaluate the same trained T2Mo model with trajectory conditioning disabled. Specifically, all trajectory anchors receive the null embedding while all model weights remain unchanged; this variant is therefore not separately trained or fine-tuned for text-only generation. As shown in Table 9(a), T2Mo is consistently preferred over both AnimateAnyMesh and BiMotion across PA, MP, and ME. A Friedman test on per-participant mean ratings shows a significant method effect on all three criteria ($\chi^2(2) \geq 27.8$, $p \leq 9.1 \times 10^{-7}$). Holm-corrected paired Wilcoxon signed-rank tests further confirm that T2Mo is rated significantly higher than both baselines on all three criteria (all corrected $p \leq 2.7 \times 10^{-4}$; matched-pairs rank-biserial $r \geq 0.99$).

Table 11: Additional robustness analysis under variations in trajectory source points, control-point selection, and missing trajectory segments. Source-point perturbations sample each trajectory source within a sphere of radius r around its original position. Random control points are sampled from the top-30% high-motion vertices. For missing-segment inputs, an intermediate fraction ρ of each trajectory is removed and the remaining segments are reconnected.

Analysis	Setting	$ADE_{3D} \downarrow$	$FDE_{3D} \downarrow$	OC ↑	AQ ↑	MS ↑	DD ↑	AMD ↑
Original		0.089	0.071	0.175	0.544	0.984	0.743	0.147
Source-point perturbation	$r = 0.05$	0.091	0.072	0.175	0.543	0.983	0.723	0.146
	$r = 0.10$	0.094	0.072	0.175	0.544	0.983	0.757	0.147
	$r = 0.20$	0.114	0.080	0.175	0.544	0.983	0.771	0.145
Control-point selection	Random 5	0.087	0.067	0.174	0.554	0.982	0.740	0.142
Missing segment	$\rho = 10\%$	0.089	0.072	0.175	0.544	0.983	0.743	0.146
	$\rho = 20\%$	0.089	0.073	0.175	0.543	0.983	0.714	0.146
	$\rho = 30\%$	0.091	0.069	0.175	0.542	0.983	0.714	0.145

Trajectory embedding comparison. We further evaluate whether the proposed shape-grounded trajectory embedding (SGTE) provides perceptual benefits over simpler trajectory fusion strategies. We compare SGTE against per-trajectory tokenization and null padding while keeping all other settings identical. As shown in Table 9(b), SGTE is consistently preferred across PA, MP, and ME. A Friedman test reveals a significant effect of the trajectory embedding design on all three criteria ($\chi^2(2) \geq 24.9, p \leq 3.8 \times 10^{-6}$). Holm-corrected paired Wilcoxon signed-rank tests further confirm that SGTE is rated significantly higher than both alternative embedding strategies on all three criteria (all corrected $p \leq 1.3 \times 10^{-3}$; matched-pairs rank-biserial $r \geq 0.92$).

Evaluation with user-drawn trajectories. To assess the practicality of trajectory-based control beyond automatically sampled trajectories, we evaluate T2Mo using trajectories specified through a simple three-view drawing interface. The interface presents the object from three orthogonal views, where trajectory waypoints are synchronized across views, allowing users to specify 3D coordinates through 2D interactions. Seven participants each draw trajectories for five object–text pairs, with at most five trajectories per object. Drawing all trajectories for one object takes $3{:}35 \pm 1{:}00$ minutes on average. Tab. 10 compares generations conditioned on user-drawn trajectories with those using automatically sampled trajectories on the same five-object subset. User-drawn inputs yield similar ADE and preserve comparable VBench metrics, while exhibiting a larger final-frame error (FDE). The perceptual evaluation, conducted with 20 raters, also shows comparable prompt alignment with moderately reduced motion plausibility and expressiveness. Overall, this small-scale study suggests that T2Mo can accommodate manually specified trajectories without requiring trajectories extracted from ground-truth motion, while more comprehensive evaluation of trajectory-authoring interfaces remains an important direction for future work.

Additional robustness analysis. We further examine robustness to variations that may arise when trajectories are specified manually. First, we perturb the source point of each conditioning trajectory within a sphere of radius r around its original location. As shown in Table 11, the quality metrics remain stable across all perturbation levels, while trajectory error increases gradually only for the largest displacement ($r = 0.20$). Second, replacing the rule-based control-point selection with five randomly sampled trajectories from the top-30% high-motion vertices produces comparable trajectory adherence and generation quality, indicating limited sensitivity to the particular choice of controlled points. Finally, we evaluate incomplete trajectories by removing intermediate portions of 10–30% and reconnecting the remaining segments before spline fitting. Performance remains largely stable under these missing-segment inputs. Together with the sparse and noisy trajectory experiments in the main paper, these results indicate that T2Mo is robust to several forms of imperfect trajectory specification.

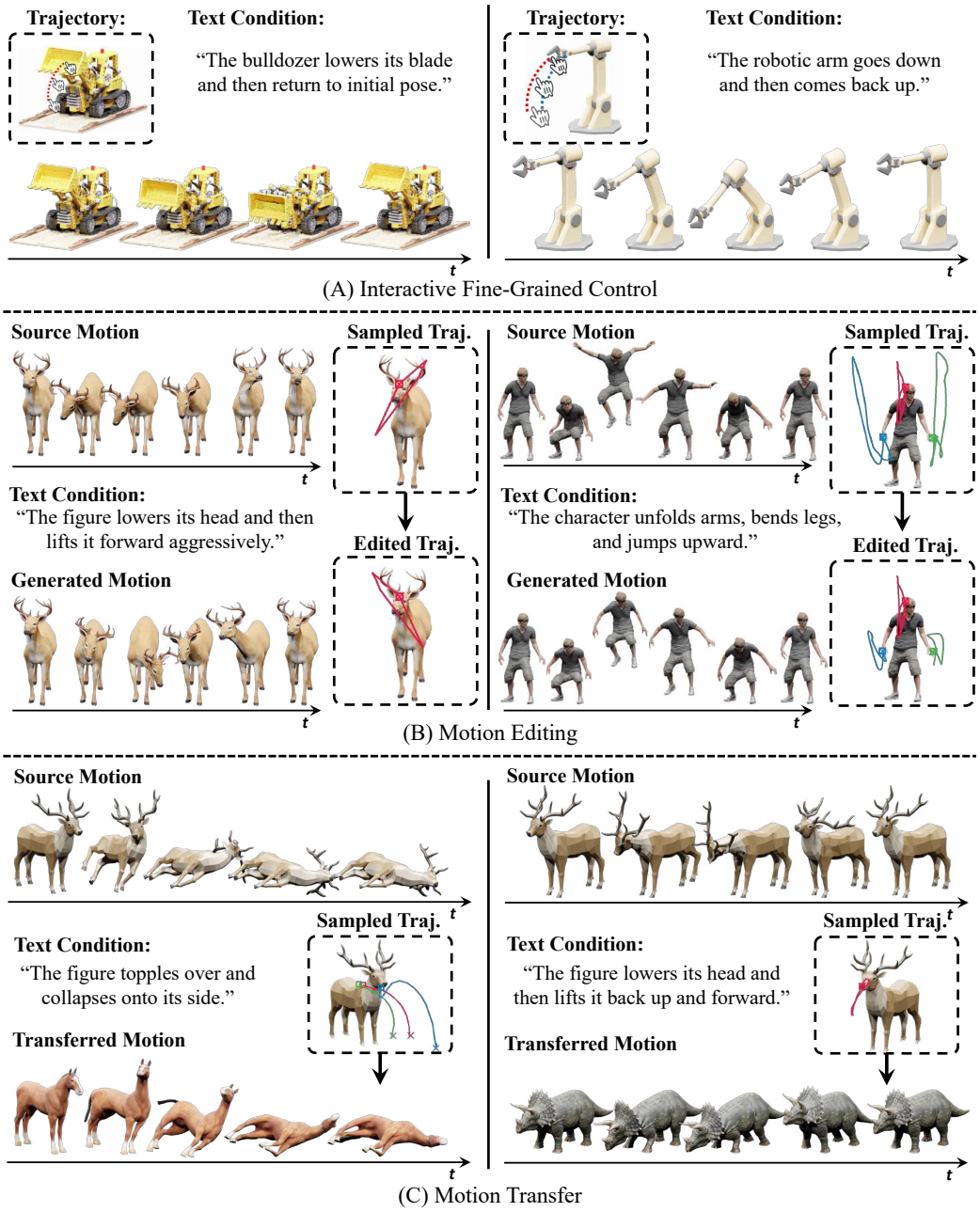

Figure 7: Application showcase. Our framework supports a diverse range of motion controls through joint trajectory and text conditioning. *Top*: fine-grained motion control with user-specified trajectories. *Middle*: motion editing by modifying a few waypoints of representative trajectories. *Bottom*: motion transfer from a source motion to a target mesh via sampled representative trajectories. Our model uses only the visualized trajectories as input.

D Applications

Trajectories provide a flexible medium for expressing various forms of motion control. By accepting trajectory sets of arbitrary configuration, our model exploits this flexibility to enable diverse applications, including interactive fine-grained control, motion editing, and motion transfer.

Interactive fine-grained control. It is challenging to describe the precise motion which a user imagines, and as shown in Fig. 3, models conditioned solely on text often struggle to faithfully reflect

detailed prompts. In contrast, our model accepts explicit spatial trajectories, allowing users to specify motion attributes such as direction and magnitude. As shown in the first row of Fig. 7, by reflecting user-provided trajectories together with the global semantics of the text prompt, our model enables fine-grained motion generation in an interactive manner, specifying the direction and magnitude.

Motion editing. Refining a previously generated or pre-existing motion is a time-consuming and laborious task: without rigging, it requires directly adjusting the movements of many individual vertices, and even with rigging, multiple skeletal motions must be coherently edited manually. Since our model can generate motions consistent with a sparse set of input trajectories, editing reduces to modifying a few waypoints of a small number of representative trajectories in the region of interest. Specifically, given a set of representative trajectories $\mathcal{T}_{\text{rep}} = \{\tau_k\}_{k=1}^{|\mathcal{T}_{\text{rep}}|}$ sampled from the source motion, the user edits a subset of waypoints indexed by $\mathcal{I} \subseteq \{1, \dots, |\mathcal{T}_{\text{rep}}|\} \times \{1, \dots, T\}$. For each $(k, t) \in \mathcal{I}$, the original waypoint $\mathbf{p}_{k,t}$ is replaced with an edited waypoint $\mathbf{p}^{\star}_{k,t}$, while all other waypoints are kept unchanged. This yields an edited trajectory set $\mathcal{T}^{\star}_{\text{rep}}$. We then pass $\mathcal{T}^{\star}_{\text{rep}}$, the text prompt, and a target frame of the source motion as the source mesh to our model, which generates the edited motion. We show examples in the second row of Fig. 7.

Motion transfer. Transferring a source motion to another mesh in a spatially consistent manner is difficult to achieve with text alone, as text cannot impose precise spatial constraints, and is also unstable with video-based approaches due to topology mismatch. In contrast, 3D trajectories provide a topology-agnostic motion representation, since they specify point-wise paths in space rather than depending on the mesh connectivity or the reference topology. By sampling a few representative trajectories from a source object and attaching them to the corresponding vertices of a target mesh, our model performs motion transfer without additional supervision. Specifically, given a source motion defined on a mesh $\mathcal{M}_{\text{src}} = (\mathbf{V}_{\text{src}}, \mathbf{F}_{\text{src}})$ with per-vertex trajectories $\mathcal{T}_{\text{src}} = \{\tau_k\}_{k=1}^{|\mathbf{V}_{\text{src}}|}$, we sample a sparse subset $\mathbf{V}^{\star}_{\text{src}} \subset \mathbf{V}_{\text{src}}$ of representative vertices and their associated trajectories $\mathcal{T}^{\star}_{\text{src}} \subset \mathcal{T}_{\text{src}}$. We attach $\mathcal{T}^{\star}_{\text{src}}$ to the corresponding vertices $\mathbf{V}^{\star}_{\text{tgt}}$ of the target mesh $\mathcal{M}_{\text{tgt}} = (\mathbf{V}_{\text{tgt}}, \mathbf{F}_{\text{tgt}})$. Here, the correspondence between $\mathbf{V}^{\star}_{\text{src}}$ and $\mathbf{V}^{\star}_{\text{tgt}}$ is specified manually by selecting semantically matching points (e.g., corresponding limbs or other shared parts) on the two meshes; since the representative subset is sparse—only three points in our examples—this requires minimal user effort. We then feed the text prompt, the attached trajectories, and $\mathcal{M}_{\text{tgt}}$ to our model to transfer the source motion onto the target mesh. We showcase examples in the last row of Fig. 7, where only three representative trajectories are sampled from the source motion.

E Social impact and limitations

Social impact. This work aims to make dynamic 3D content creation more accessible by enabling users to control mesh motion through sparse 3D trajectories and natural language. Such controllability can reduce the amount of manual labor required for animation, particularly for users without expertise in rigging, keyframing, or skeletal motion editing, thereby supporting more efficient workflows in digital content creation, AR/VR, gaming, filmmaking, education, and simulation. At the same time, controllable dynamic 3D generation may introduce risks. Generated animated assets could be misused to create deceptive, misleading, or harmful visual content, especially when combined with realistic rendering. The ability to edit or transfer motion may also raise concerns about unauthorized reuse of copyrighted assets, character designs, or motion data. In addition, if training data contain biased or inappropriate content, generated motions may inherit or amplify such biases. These risks highlight the importance of responsible data curation, compliance with dataset licenses, clear usage guidelines, and provenance mechanisms such as watermarking or metadata for generated assets. We encourage release practices that include documentation of training data sources, intended use cases, and known limitations.

Limitations. Although T2Mo offers fine-grained controllability via 3D trajectories and text, our formulation predicts per-vertex displacements on a fixed source mesh and therefore assumes a constant topology across frames. Consequently, motions involving topological changes, such as splitting, merging, or the emergence of new components, lie outside the scope of our model. Integrating our shape-grounded trajectory embedding with per-frame generation methods is a promising direction for extending trajectory control to dynamic geometry with varying topology. While our category-agnostic model is evaluated on objects disjoint from the training assets, the range of motions it can

generate remains bounded by the training distribution. In particular, we do not separately evaluate material-dependent physical behaviors such as elastic or plastic soft-body deformation. Incorporating physics-aware data or objectives may improve generalization to such motions.