INFUSING MOTION INTELLIGENCE INTO STRUCTURED 3D WORLDS FOR 4D SCENE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Generating a 4D scene from a single image requires endowing a structured 3D world with meaningful and coherent dynamics. Existing image-to-4D methods predominantly introduce dynamics through video generation and subsequently reconstruct 3D scenes. While video models provide strong motion priors, their dynamics are formed in 2D space, making scene geometry, physical constraints, and fine-grained motion control difficult to explicitly access and verify. In contrast, explicit 3D representations provide a structured world in which geometry, physical conditions, and motion states can be directly represented and manipulated, but lack the rich motion intelligence required to translate semantic instructions into meaningful dynamics. We propose **SceneWeaver**, a 3D-native 4D generation framework that bridges this gap through a *Coordinator-Executor-Verifier* architecture. Given an image and a prompt, a coordinator interprets the motion intent and translates it into structured conditions describing *where* the subject should move, *how* it should move, and *how* the environment should affect its motion. Two specialized executors then generate the corresponding global world-space trajectory and local articulated motion within the explicit 3D scene. Specifically, a potential-augmented Neural ODE, which is geometry- and physics-aware, generates the global trajectory by jointly modeling interpreted motion control, scene geometry, and environmental effects, while an LLM-based motion planner produces fine-grained articulated motion from structured motion descriptions. Crucially, the explicit 3D representation also enables numerical verification of intermediate motion states. A verifier identifies geometric, physical, and motion-related failures, localizes root causes and enables targeted repair through coordinator. This establishes a closed-loop generation in which 3D structure serves not only as the intermediate results, but also as the basis for verification and recovery. Experiments demonstrate that SceneWeaver shows promising controllability, motion quality in generated 4D scenes, offering an alternative paradigm for image-to-4D task.

1 INTRODUCTION

Generating a 4D scene from a single image is a fundamental problem for enabling immersive digital experiences and applications in fields such as virtual reality Bryson (2013), film production, and interactive media Deliyannis (2012). In particular, image-to-4D generation aims to transform a static image into a temporally evolving scene in which a target subject performs the desired behavior. An ideal system should not merely produce a visually plausible video, but should generate a coherent 4D world in which the subject can move within an explicit environment, remain geometrically consistent over time, and be controlled according to the intended behavior requirement.

A straightforward solution is to first synthesize a video and subsequently lift it into a 3D space. Recent image-to-4D approaches predominantly follow this paradigm, leveraging increasingly powerful video-generation priors to introduce temporal dynamics Zhao et al. (2023); Sun et al. (2024); Gupta et al. (2024); Gao et al. (2024); Zhao et al. (2025); Chen et al. (2025a;b); Wu et al. (2024); Liu et al. (2025). Such a video-first formulation benefits from the rich motion priors encoded in large-scale video models, allowing them to produce visually convincing temporal changes. However, ***visual plausibility does not necessarily imply a coherent 4D world***. Since the dynamics are primarily formed in 2D video latent space, scene geometry and subject motion are implicitly entangled. Geometry may drift across timeAckermann et al. (2026); Wang et al. (2025); Yang et al. (2026), and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

physical constraints are difficult to explicitly impose during generation. Moreover, the resulting dynamics are tied to the generated image sequence, making fine-grained control over the subject's trajectory, articulated pose, and interaction with the surrounding environment difficult. Although a subsequent reconstruction stage can recover a 3D representation, it cannot fundamentally restore the explicit structural information that was unavailable during dynamic generation.

These limitations motivate us to reverse the conventional generation order: rather than generating dynamics in video space and reconstructing the world afterward, we generate dynamics directly within an explicit 3D world. Such a representation provides several fundamental advantages. All temporal states can be defined in a common canonical space, multi-view rendering is naturally supported, geometric and physical constraints can be explicitly imposed, and motion can be represented through interpretable quantities such as world-space trajectories and articulated poses. However, moving the generation process into 3D introduces a different challenge: ***3D representations provide rich knowledge of how the world is structured, but lack the motion intelligence needed to determine how that world should evolve.*** A prompt like "dog runs along the windy beach" does not directly specify a valid world-space trajectory, articulated motion, or the environmental conditions under which the behavior should occur. Thus, the central challenge becomes how to bridge semantic motion intent expressed in prompt and structured dynamic states required by an explicit 3D world.

We address this challenge with **SceneWeaver**, a 3D-native 4D generation framework based on a *Coordinator-Executor-Verifier* architecture. Instead of directly mapping the prompt to an entangled dynamic representation, SceneWeaver first establishes an explicit 3D world containing the independently manipulable subject and background environment. A *Coordinator* then interprets the user's motion intent from three complementary aspects: *where* the subject should move, *how* the subject should move, and *how* the environment should affect the motion. These semantic factors are translated into heterogeneous but structured conditions for downstream motion generation. In particular, the Coordinator produces a trajectory-control sequence describing the desired global movement, a fine-grained motion description specifying local articulated behavior, and environmental conditions describing relevant physical effects. The resulting conditions are executed by two specialized motion Executors. (1) The *Global Trajectory Executor* determines where the subject moves in the 3D world. It employs a potential-field augmented Neural ODE that integrates the interpreted trajectory control with scene geometry and environmental effects, allowing the resulting world-space trajectory to respond to both user intent and physical constraints. (2) In parallel, the *Local Locomotion Executor* determines how the subject moves. Given the articulated subject representation and the fine-grained motion description, an LLM-based intent planner decomposes the behavior into structured joint-level edits and generates a temporally evolving articulated pose sequence. This separation reflects two complementary forms of dynamics: global displacement depends on the world, environment, and movement intent, whereas local articulation depends primarily on the subject's articulated structure and semantic action. The two components are finally recomposed in 3D to form the complete 4D subject motion. (3) Crucially, explicit 3D representation also enables a different generation paradigm for verification. Rather than evaluating only the final rendered video, SceneWeaver maintains structured intermediate 3D states throughout the generation process. A *Verifier* therefore examines the generated trajectory, local articulated motion, and composed 4D scene using geometric and physical constraints. For example, trajectory-level checks can determine whether the subject follows the intended direction or leaves the valid scene region, while locomotion and scene-level checks can identify unnatural poses, penetration, or floating. The resulting diagnostic report records the affected stage and potential root cause, which is fed back to the Coordinator for targeted recovery. The generation process thus forms a closed loop with generation, verification and targeted refinement, which allows failures to be corrected at their source rather than requiring complete regeneration.

This formulation allows SceneWeaver to combine the complementary strengths of video-based and 3D-based generation: motion is introduced through language-guided coordination and specialized executors, while explicit 3D representation provides the structural, geometric, physical, and controllable foundation on which these dynamics are executed and verified. Extensive experiments show that SceneWeaver achieves stronger motion range, controllability, and physical plausibility than existing image-to-4D and video-generation approaches. Our contributions are summarized as follows: (1) We formulate 3D-native 4D generation as dynamic synthesis directly within an explicit 3D world, leveraging its structural, geometric, physical, and controllable properties instead of introducing dynamics primarily in video space. (2) We introduce a *Coordinator-Executor-Verifier*

framework for motion intelligence, which bridges language-level motion intent and heterogeneous 3D motion conditions, and separately generates global world-space trajectories and local articulated motion for compositional and fine-grained control. (3) We introduce 3D-grounded verification and targeted repair, where intermediate trajectories, articulated states, and composed scenes are explicitly evaluated using geometric and physical constraints, enabling failure localization and stage-specific recovery. (4) Extensive experiments demonstrate that SceneWeaver improves motion range, controllability, physical plausibility, and generation efficiency compared with existing image-to-4D and video-generation methods.

2 Related Works

4D content generation. Early models Singer et al. (2023); Zhao et al. (2023); Ren et al. (2023) inject dynamics by distilling video generation priors and mainly focus on single objects. Relying on advanced 3D generation and dynamic scene reconstruction works like Zhang et al. (2024b), more recent works Bahmani et al. (2024); Sun et al. (2024); Gupta et al. (2024); Liu et al. (2025); Gao et al. (2024); Zhao et al. (2025); Chen et al. (2025a;b) extend to scene-level 4D generation. Despite progress, the dynamics are still primarily acquired from video generation, and the fine-grained and stable control over subjects' trajectory, articulated pose remain entangled and difficult to control. Recent attemptsYuan et al. (2026); Zhang et al. (2026a) explore to formulate the task as code programming process, yet their main dynamics are still in-place motion(e.g., leaves sway on the tree). In contrast, our approach generates dynamics directly within an explicit 3D world, where global trajectory and local articulation are separately planned, executed, and verified, enabling high controllablity over its state evolution.

Agentic Visual Generation. Recent works begin to frame visual generation as an agentic control process in which a controller plans, invokes tools, inspects intermediate outputs, and revises failures Huang et al. (2026b). However, existing systems operate predominantly in image and video spaceZhang et al. (2026b); Zhou et al. (2026a); Bian et al. (2026); Zhou et al. (2026b). In 3D/4D generation, LLM-based agentsChi et al. (2026); Luo et al. (2026); Kim et al. (2026) have been introduced mainly for static scene construction and asset authoring, where the 3D world is treated as a fixed target to be built rather than a dynamic space to be controlled. SceneWeaver instead focuses on agentic control over the state evolution of the constructed world, aiming to produce a coherent 4D scene in which the subject moves through the environment with a controllable, articulated motion.

3 Method

3.1 Task Formulation

Given a single image $\mathcal{I}$ and a prompt $\mathcal{P}$, our goal is to generate a temporally evolving 4D scene in which the subject specified by $\mathcal{P}$ performs intended action within the 3D world. In particular, the output consists of a temporally evolving 3D subject together with the background world, such that the generated dynamics remain spatially coherent and consistent with the described behavior in $\mathcal{P}$.

3.2 Method Overview

The key challenge of 3D-native 4D generation is that explicit 3D representations provide strong structural constraints, but do not inherently provide the motion intelligence needed to translate semantic instructions into dynamics. As shown in Fig.1, SceneWeaver therefore follows a *Coordinator–Executor–Verifier* framework to bridge language-level motion intent and structured 3D dynamics. Given an image and a prompt, we first construct separate 3D representations of the prompt-specified subject and its surrounding environment. The Coordinator then interprets the prompt into global movement, local articulated motion, and environmental conditions, and routes them to two specialized Executors: one generates the world-space trajectory and the other generates local motion. Their outputs, together with the composed 4D scene, are passed to a Verifier, which evaluates these intermediate 3D states and returns structured diagnostics for targeted repair, performing in a close-loop with generation, verification and target refinement.

3.3 Explicit 3D Scene Construction

We first identify the *subject specified by the prompt* as the foreground dynamic entity whose motion is to be generated. Given the input image $\mathcal{I}$, we segment the prompt-specified subject from the

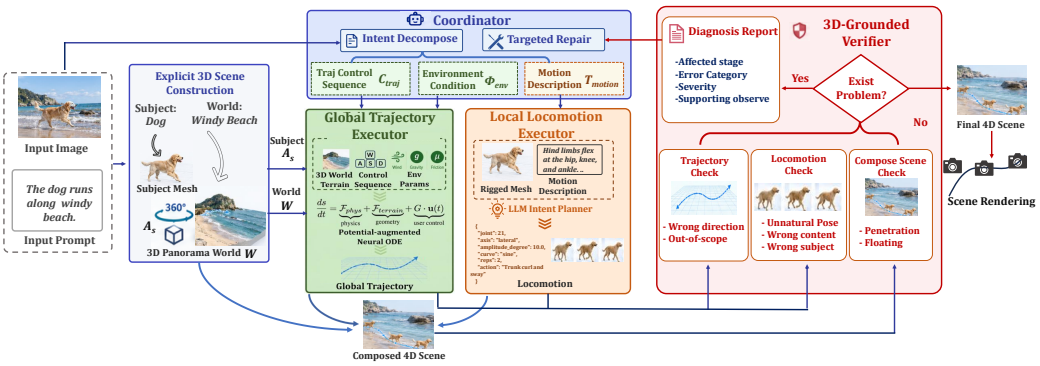


Figure 1: **Overview of SceneWeaver.** Given an input image and prompt, SceneWeaver first constructs an explicit 3D scene, then uses a Coordinator to translate prompt into global trajectory, local motion, and environmental conditions. Specialized Executors generate the corresponding dynamics, which are composed into a 4D scene. A verifier then examines these intermediate representation in 3D space. Detected failures are localized and repaired by re-invoking the responsible module.

image to obtain an independent foreground image $\mathcal{I}_{\text{fg}}$, then inpaint the remained part to obtain a complete background image $\mathcal{I}_{\text{bg}}$. We then process them independently into 3D. $\mathcal{I}_{\text{fg}}$ is used to generate an articulated subject $\mathcal{A}_s = (\mathcal{M}_s, \mathcal{K}_s, \mathcal{W}_s)$, where $\mathcal{M}_s$, $\mathcal{K}_s$, and $\mathcal{W}_s$ denote the subject mesh, articulated skeleton, and skinning weights. $\mathcal{I}_{\text{bg}}$ is first extrapolated into a 360° panorama and subsequently lifted into a 3D environment mesh $\mathcal{M}_b$. We denote world as $\mathcal{W} = (\mathcal{M}_b, \phi_b)$, where ϕ_b is its signed distance field. This decomposition separates the prompt-specified dynamic subject from the static environment and establishes a common explicit 3D space in which following global trajectory, local articulated motion can be generated and evaluated.

3.4 Intent Analysis and Motion Coordination

A motion should specify both *where* the subject moves and *how* it acts locally. We therefore decompose the dynamics into a global trajectory $X_{1:T} = \{\mathbf{x}_t\}_{t=1}^{T}$, where $\mathbf{x}_t \in \mathbb{R}^3$ denotes the subject's world-space position at time step t, and a local articulated pose sequence $Q_{1:T} = \{\mathbf{q}_t\}_{t=1}^{T}$, where $\mathbf{q}_t$ denotes the pose at time step t. The former depends on movement intent and the environment, while the latter depends on the semantic action and articulated structure. To translate coarse-grained prompt intent into these structured dynamics, we employ a Coordinator that parses the prompt into the conditioning signals each generation step requires: a trajectory-control sequence $C_{\text{traj}} = \{u_t\}_{t=1}^{T-1}$, where each $u_t \in \mathbb{R}^4$ denotes the encoding of WASD keys specifying the intended movement direction at time step t; a fine-grained motion description $\mathcal{T}_{\text{motion}}$, denoting a part-level textual specification of the local joint motion, e.g., which part should move and in what manner; and environmental conditions $\Phi_{\text{env}} \in \mathbb{R}^3$, which in principle model how the environment affects the subject's motion. In this work, we consider three such effects, namely gravity, friction, and wind, and represent their aggregated influence as a three-dimensional acceleration, $\Phi_{\text{env}} = \{a_{\text{gravity}}, a_{\text{friction}}, a_{\text{wind}}\}$. The Coordinator further routes each task to the required Executors, activating global trajectory generation and local motion generation according to the described behavior, which provides a unified interface between semantic motion intent and the two complementary forms of 3D dynamics.

3.5 Global Trajectory Executor

Global trajectory executor determines *where* the subject moves within the 3D world. With an explicit 3D scene, the environment's influence on motion can be characterized more directly; we therefore build upon the trajectory-control sequence C_{traj} and introduce two terms that inject physical and geometric conditions from environment, leading to a potential-field augmented Neural ODE:

$$\frac{d\mathbf{s}}{dt} = F_{\text{phys}}(\mathbf{s}(t), \Phi_{\text{env}}) + F_{\text{terrain}}(\mathbf{x}(t), \phi_b) + G \cdot \mathbf{u}(t), \tag{1}$$

where $\mathbf{s} = [\mathbf{x}, \mathbf{v}, \mathbf{a}]^\top$ denotes the position, velocity, and acceleration state of the subject, and G is the fixed matrix that maps the control signal to the corresponding directional movement. Specifically,

$\mathcal{F}_{phys}$ models the environmental influence summarized by Φ_{env} (gravity, friction, and wind), modulated by an MLP-based Neural ODE (see training details in the appendix). $\mathcal{F}_{terrain}$ incorporates the scene geometry through an SDF-based potential field:

$$\mathcal{F}_{terrain}(\mathbf{x}) = -k_{rep}\nabla_{\mathbf{x}}\phi_b(\mathbf{x})\exp\left(-\frac{\phi_b(x)}{\sigma}\right). \tag{2}$$

$\nabla_{\mathbf{x}}\phi_b$ gives the direction steers subject away from penetrating background, k_{rep} is weight scalar, while $\exp\left(-\frac{\phi_b(\mathbf{x})}{\sigma}\right)$ modulates the force according to subject's distance from the terrain: for ground locomotion, where $\phi_b(x)$ remains small, the force stays active and shapes the path along local terrain geometry, whereas for aerial locomotion in open regions such as the sky, where $\phi_b(x)$ grows large, it decays to negligible and the remaining dynamics dominate. Integrating the above dynamics over user intention, physical and geometric constraints yields a world-space trajectory $\mathbf{X} = \{\mathbf{x}_t\}_{t=1}^{T}$ with its velocity and acceleration profiles that is both geometry- and physics-aware.

3.6 Local Locomotion Executor

The Local Locomotion Executor determines *how* the subject performs the described action. Given the articulated subject representation $\mathcal{A}_s$ and motion description $\mathcal{T}_{\text{motion}}$, an LLM-based intent planner decomposes the behavior into meaningful sub-actions and translates them into structured joint-level edits. Specifically, each edit is represented as $e_i = (j_i, a_i, A_i, c_i, r_i, d_i)$,, where j_i denotes the target joint, a_i denotes its rotation axis, A_i the motion amplitude, c_i the temporal curve, r_i the repetition count, and d_i the corresponding semantic description. The joint-level edits are applied to the articulated skeleton K_s and propagated to the mesh through the skinning weights W_s, resulting in the local pose sequence $Q_{1:T}$. Since this process models local articulation independently of global displacement, the generated local motion can be combined with different world-space trajectories. The final dynamic subject is obtained by $M_s^t = \text{Comp}(x_t, q_t, \mathcal{A}_s), t = 1, \ldots, T$, and composing the dynamic subject with the background world at each timestep yields the 4D scene $\mathcal{S}_{4D} = \text{Comp}\left(\mathcal{M}_{bg}, \{M_s^t\}_{t=1}^{T}\right)$, where $\text{Comp}(\cdot)$ denotes the composition operation, $\mathcal{M}_{bg}$ is the background mesh and M_s^t is the dynamic subject at timestep t. This composed 4D scene, along with separate trajectory and locomotions, serve as the target of the verification and repair stage.

3.7 3D-grounded Verification and Targeted Repair

Since SceneWeaver maintains explicit 3D representations throughout generation, intermediate motion states can be actively inspected against structured geometric and physical constraints, rather than being judged only from the final rendered video. We therefore introduce a verifier that separately examines the generated global trajectory, local articulated motion, and their composed 3D scene, and reports failures in a form that the Coordinator can act upon.

3.7.1 Trajectory Verification

We verify whether the generated trajectory follows the intended movement direction and remains within the valid region of the reconstructed environment. Let $\Delta X = \mathbf{x}_T - \mathbf{x}_1$ be the generated displacement and ΔX_{cmd} the displacement implied by C_{traj}; the directional discrepancy and the out-of-region ratio are measured by

$$E_{\text{dir}} = 1 - \frac{\Delta X^{\top}\Delta X_{\text{cmd}}}{\|\Delta X\|_2 \, \|\Delta X_{\text{cmd}}\|_2}, \qquad E_{\text{bd}} = \frac{1}{T}\sum_{t=1}^{T}\mathbb{I}\left[\mathbf{x}_t \notin \Omega_{\mathcal{W}}\right], \tag{3}$$

where $\Omega_{\mathcal{W}}$ denotes the valid region of the environment, i.e., the interior of the water-tight background mesh $\mathcal{M}_{bg}$. The trajectory is flagged as a direction error when $E_{\text{dir}} > \tau_{\text{dir}}$, or as a boundary violation when $E_{\text{bd}} > \tau_{\text{bd}}$.

3.7.2 Local Motion Verification

Failures in local motion may arise either from incorrect subject segmentation, skeleton binding or locomotion planning that yield unnatural poses. Since these are visually manifested, we render the generated motion into image sequences and employ verifier to inspect whether the subject remains

structurally intact and whether the generated joint edits produce natural articulation. The observations are converted into structured failures associated with the corresponding local-motion stage.

3.7.3 Composed Scene Verification

After the trajectory and local motion are composed, we verify their spatial relationship with the environment, specifically whether penetration or floating is occurred. Let $\mathcal{P}_t$ be the set of sampled points on the subject surface at timestep t, and $\mathcal{P}_t^{\mathrm{c}} \subseteq \mathcal{P}_t$ the contact points expected to touch the environment; with ϕ_b denotes the background SDF and $d_{\mathcal{W}}(p)$ denotes the distance from p to the nearest supporting surface, the average penetration depth and floating distance are

$$E_{\mathrm{pen}} = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{|\mathcal{P}_t|}\sum_{p\in\mathcal{P}_t}\max\left(0,-\phi_b(p)\right), \qquad E_{\mathrm{float}} = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{|\mathcal{P}_t^{\mathrm{c}}|}\sum_{p\in\mathcal{P}_t^{\mathrm{c}}} d_{\mathcal{W}}(p). \tag{4}$$

A penetration error is reported when $E_{\mathrm{pen}} > \tau_{\mathrm{pen}}$, and a floating error when $E_{\mathrm{float}} > \tau_{\mathrm{float}}$. Note that these checks are invoked only when applicable. For instance, for subjects that are not expected to maintain ground contact, like flying in the sky, the floating check is skipped.

3.7.4 Diagnostic Report and Targeted Repair

The results from the three stages are aggregated into a structured diagnostic report $\mathcal{R} = \{r_k\}_{k=1}^{N}$ with $r_k = (\ell_k, c_k, s_k, e_k)$, where ℓ_k is the affected stage, c_k the error category (e.g., direction deviation, boundary violation, unnatural pose, penetration, or floating), s_k the severity, and e_k the supporting measurement or observation. Based on $\mathcal{R}$, the Coordinator identifies the affected stage and re-invokes the corresponding module with updated conditions. For instance, for a local-motion error, it may refine the motion description or re-segment and re-generated the rigged subject mesh to provide updated condition, then re-invoke the Local Locomotion Executor. Such process thus forms a closed loop, Generation $\rightarrow$ Verification $\rightarrow$ Targeted Repair with Root-cause Localization, so that a detected failure is repaired at its responsible stage without regenerating the entire 4D scene.

4 Experiments

4.1 Experimental Settings

Implementation Details. For scene decomposition, we use Hunyuan3D Lai et al. (2025) to construct 3D scene and AniGenHuang et al. (2026a) to construct rigged mesh. For Agent coordinator, motion planner and verifier, we employ GPT 5.6 solOpenAI (2026). Trajectory planner is implemented with a MLP-based Neural ODE network. See its architecture and training details in the appendix.

Benchmark. We curate a benchmark adapted from WorldScore Duan et al. (2025) with 60 samples, covering diversified subjects, environments, terrains, and motion patterns. Each sample consists of an input image and a text prompt describing the desired subject behavior.

Compared Methods. We compare against two lines of methods: *image-to-4D generation*, including Free4D Liu et al. (2025), DeepVerse Chen et al. (2025a), 4DNeX Chen et al. (2025b); *video generation*, including Wan2.2-I2V-14B Wan et al. (2025) and CogVideoX-I2V-5B Yang et al. (2024).

Evaluation Metrics. We follow WorldScore Duan et al. (2025) to evaluate *Quality*, *Dynamics*, and *Controllability*. Quality includes 3D consistency and subjective quality; Dynamics includes motion smoothness and motion magnitude; and Controllability includes object control and content alignment. We additionally evaluate *physical* behavior with Trajectory Quality, which measures whether the generated global movement follows instruction, and *Environmental Awareness* measures whether the dynamics responds appropriately to environmental conditions. Details in appendix.

4.2 Comparison with State-of-the-Art

Quantitative Comparison. Table 1 compares SceneWeaver with representative video generation and image-to-4D methods. SceneWeaver achieves the strongest overall performance on motion-related and 3D-aware metrics, while maintaining competitive visual quality. These results support our central formulation of generating dynamics in an explicit 3D world: (1) Explicit 3D scene construction

Table 1: Quantitative comparison with image-to-4D and video generation methods. All metrics are higher-is-better. Inference Speed are tested on NVIDIA RTX 4090 GPUs for a 49-frame video.

Method	Infer Speed (min.) ↓	Property			Controllability		Quality		Dynamics		Physical	
		2D Video model-free	Fore/Background Separatable	4D Output	Object Ctrl.↑	Content Align.↑	3D Const.↑	Subjective Quality↑	Motion Smooth↑	Motion Mag.↑	Traj. Quality↑	Env. Awareness↑
Video Generation												
Wan2.2-I2V-14B Wan et al. (2025)	∼2	✗	✗	✗	95.28	60.93	64.97	**57.04**	54.52	21.98	58.30	51.70
CogVideoX-I2V-5B Yang et al. (2024)	∼2	✗	✗	✗	93.64	58.85	57.41	54.13	52.17	20.24	45.92	39.85
Image-to-4D Generation												
DeepVerse Chen et al. (2025a)	∼1	✗	✗	✓	91.85	53.02	77.44	46.21	67.35	16.80	44.70	37.10
Free4D Liu et al. (2025)	82	✗	✗	✓	97.46	58.58	82.73	47.15	76.28	17.64	60.05	51.32
4DNeX Chen et al. (2025b)	52	✗	✗	✓	97.12	**61.55**	81.02	46.88	68.62	17.15	59.40	53.60
SceneWeaver	∼10	✓	✓	✓	**98.27**	60.97	**84.76**	47.58	**89.73**	**24.39**	**78.64**	**73.17**

preserves the subject as a dedicated articulated asset and provides a consistent spatial representation for motion generation and verification, leading to stronger object controllability and geometric consistency. (2) Decoupling global trajectory from local locomotion enables independent control over *where* the subject moves and *how* it moves, allowing substantially larger spatial displacement while preserving smooth articulated motion, yielding the best dynamics-related performance on motion smoothness and magnitude. (3) The 3D environment provides explicit geometry and physical conditions, allowing terrain and environmental effects to directly influence the generated trajectory, yielding higher physical-related metrics for the trajectory reasonability and environmental awareness. (4) For inference speed, SceneWeaver runs at a moderate cost that is substantially faster than Free4DLiu et al. (2025) and 4DNeXChen et al. (2025b) while delivering higher generation quality. This indicates that our explicit 3D-based generation does not trade efficiency for control, but rather achieves a favorable balance between speed, controllability, and motion quality. We note video generation methods show higher subjective quality compared with all image-to-4D generation methods, since video generation methods benefit from large-scale training with higher perceptual quality. However, their dynamics are formed implicitly in latent space with weaker motion control and environmental condition awareness. Overall, it demonstrate that structured 3D-native paradigm is valid for controllable, coherent, and physically grounded dynamics generation.

Qualitative Comparison. Figure 2 presents comparisons across diverse subjects, motion instructions, and environments. Video generation methods generally produce visually appealing motion, but may fail to faithfully execute spatial or physical instructions. For example, in the third case (Lines 5-6), Wan2.2 does not follow the instruction to knock the cup over, while CogVideoX produces a motion that violates the expected physical process. In contrast, SceneWeaver explicitly plans the cup to fall after being knocked over, and models the trajectory in aware of physical constraints(influenced by gravity) and geometry constraints(fall to the ground without penetration) within the explicit 3D environment. Similar benefits are observed in the second case, where SceneWeaver generates the intended leftward turning trajectory of the bird. Compared with image-to-4D methods, which often exhibit relatively small-range or in-place motion, SceneWeaver produces larger and more structured global displacement while retaining local articulated motion, as illustrated by the bird flying (Case 2) and the boat sailing(Case 4). These examples highlight the benefit of separating global trajectory from local locomotion and executing both within an explicit 3D space.

4.3 Analysis of the Coordinator–Executor–Verifier Framework

We evaluate the Coordinator-Executor-Verifier framework from following dimensions. (1) Effectiveness of the *agentic coordination and recovery process*, measuring whether explicit motion decomposition, intermediate verification, and targeted repair improve generation quality and efficiency. (2) Effectiveness of *explicit 3D-grounded modeling*, evaluating contributions of introducing physical, geometric, and semantic conditions within 3D space to the generated dynamics. (3) Analysis of *verifier dimensions*, including the trajectory, locomotion and composed scene checks.

Table 2: Ablation of Coordinator-Executor-Verifier framework and explicit 3D-grounded modeling. "Module Calls" denote the number of modules(coordinator,executor,verifier) are employed.

(a) Agentic Coordination and Recovery

Strategy	Module Calls↓	Time (min.)↓	Object Ctrl.↑	Traj. Quality↑	Motion Smooth↑
w/o Decomposition	12.6	23.5	89.44	58.01	70.43
w/o Verification	3.0	4.5	95.84	68.92	79.64
Full Re-generation	11.5	17.8	98.75	76.14	87.55
Ours	7.8	10.3	98.27	**78.64**	**89.73**

(b) Explicit 3D-Grounded Modeling

Strategy	Traj. Quality↑	Env. Awareness↑	Geo. Validity↑	Content Align.↑
w/o F_{phys}	66.42	54.81	81.35	58.74
w/o F_{terrain}	69.87	67.53	72.64	59.36
w/o $\mathcal{T}_{\text{motion}}$	74.21	70.86	82.17	54.82
Full Model	**78.64**	**73.17**	**84.76**	**60.97**

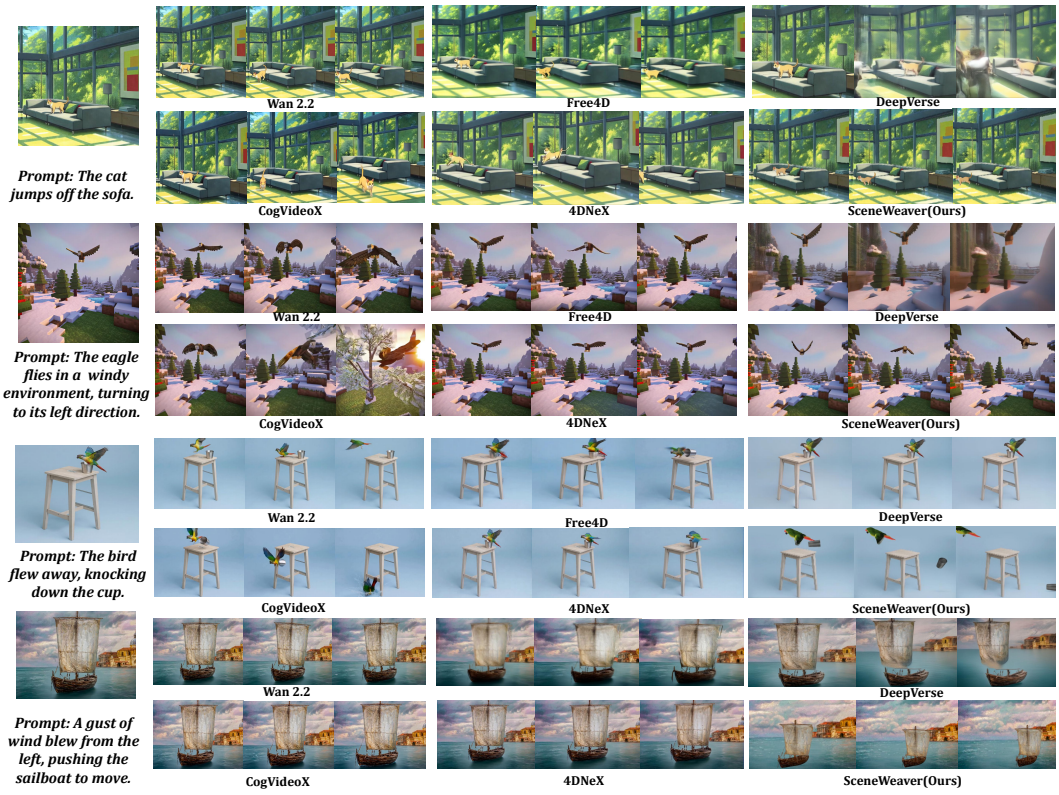


Figure 2: **Qualitative comparison.** We compare with image-to-4D and video generation methods.

Agentic Coordination and Recovery. We analyze the agentic coordination and recovery process by progressively removing explicit motion decomposition, verification, and targeted repair process. Specifically, *w/o Decomposition* directly provides the original prompt to the motion Executors for overall motion generation, without decomposing into trajectory and local motion for separate generation. *w/o Verification* removes the verification and recovery stage, such that the initial generation is directly used as the final result. *Full Re-generation* retains verification but regenerates the complete motion sequence whenever a failure is detected, rather than repairing only the responsible component. As shown in Table 2(a), (1) Removing motion decomposition leads to degraded controllability and motion quality, while increasing the number of module calls and inference time, as the motion executor can not simultaneously satisfy the heterogeneous requirements of global trajectory, local articulated motion, and environmental condition. As a result, failures are more likely to emerge after the generated motion is composed into the scene, requiring repeated refinement to resolve conflicts between different motion requirements. (2) Removing verification also shows a quality drop, indicating that intermediate validation is important for identifying failures. Although Full Re-generation can recover from these failures, it incurs substantially more module calls and inference time by re-run the whole process. In contrast, our targeted repair strategy achieves the highest trajectory quality and motion smoothness while requiring substantially fewer module calls and less inference time, demonstrating the efficiency and effectiveness of explicit decomposition, stage-level verification, and targeted recovery.

Explicit 3D-Grounded Modeling. We analyze explicit 3D-grounded modeling by individually removing the physical, geometric, and semantic conditions from the motion generation process. Specifically, we ablate the physical condition F_{phys}, the geometric condition F_{terrain}, and the semantic motion description $\mathcal{T}_{\text{motion}}$. As shown in Table 2(b), (1) Removing F_{phys} causes a clear degradation in trajectory quality and environmental awareness, indicating that explicit physical conditions help produce more plausible motion dynamics. (2) Removing F_{terrain} primarily affects environmental awareness and geometric validity, highlighting the importance of grounding trajectory generation in the reconstructed 3D geometry. (3) Without $\mathcal{T}_{\text{motion}}$, the generated motion exhibits reduced content alignment, showing it's crucial to translate intent into fine-grained articulated motion.

Verify Content	Content Align.↑	Geo. Validity↑	Motion Smooth↑
w/o Trajectory	57.24	80.13	87.45
w/o Local Motion	55.86	82.31	71.62
w/o Scene	58.43	69.87	88.24
Ours	**60.97**	**84.76**	**89.73**

Table 3: Ablation study of verifier components.

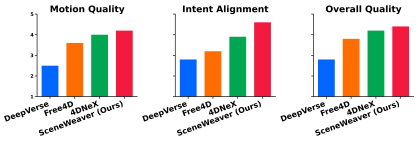


Figure 3: User study on Image-to-4D methods.

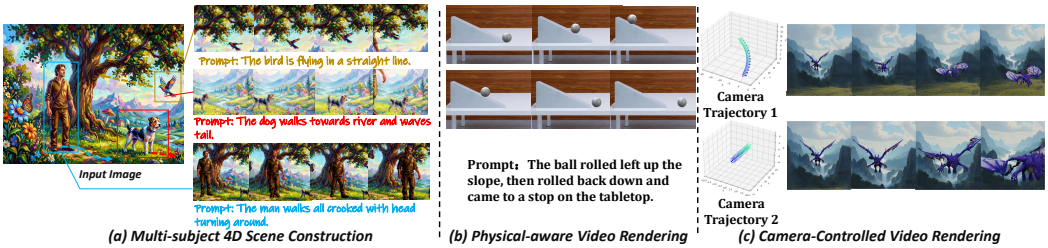


Figure 4: Diverse applications of our SceneWeaver.

Verification Ablation. We ablate on verification components by removing checks of different intermediate representations, ranging from trajectory, local motion and composed scene. As shown in Table 3, removing any verification component leads to degradation in the corresponding motion properties, for instance, removing local motion verification degrades motion smoothness and object controllability. It demonstrates that fine-grained verification at different stages is important for diagnosing heterogeneous failures and enabling effective targeted repair.

User Study. We conduct a user study to evaluate the perceptual quality of the generated 4D scenes. Participants are asked to rate each generated result on a 5-point scale(1 the worst, 5 the best) across three dimensions: *Motion Quality*, *Intent Alignment*, *Overall Quality*. Motion Quality assesses the naturalness and coherence of the subject motion, and Intent Alignment evaluates whether the scene appropriately responds to user prompt, overall Quality measures the general perceptual quality of the generated 4D scene. As shown in Fig. 3, our method achieves consistently high ratings across all three dimensions, indicating that the explicit 3D-grounded generation process produces perceptually coherent dynamics that faithfully expressing user intent.

4.4 Applications, Limitations and Future Work Discussion

The explicit 3D formulation of SceneWeaver naturally supports a range of extensions beyond single-subject 4D scene generation. Since both the subject dynamics and environmental constraints are represented in a shared 3D space, our framework can accommodate *multi-subject control* within the same scene (Fig. 4(a)). The shared spatial representation as well as physical and geometry-aware global trajectory planning executor also enables *base-physics-aware video rendering* (Fig. 4(b)), and *simple interactions between subjects*, allowing their motions to be jointly grounded in the same scene geometry (Case 3 in Fig. 2). Moreover, the 4D nature allows *multiple viewpoints rendering* (Fig. 4(c)). It suggest that 3D-native generation provides a flexible foundation for image-to-4D. However, our current method struggle to handle more complex interactions and physically grounded behaviors, which remains a promising directions for future work.

5 Conclusion

We presented SceneWeaver, a 3D-native framework for generating 4D scenes directly in an explicit 3D world from a single image and a prompt. Instead of generating videos first and subsequently reconstructing their 3D structure, SceneWeaver operates in explicit 3D space, separates global trajectory and local articulated motion, and translates high-level behavioral intent into structured motion conditions for subsequent independent control. Such 3D-based framework also facilitates 3D-grounded verification that identifies failures at intermediate stages and enables targeted repair. Experiments demonstrate that SceneWeaver achieves promising controllability, motion quality and plausibility in generated 4D scenes. Overall, SceneWeaver highlights the potential of treating 4D scene generation as explicit dynamic synthesis in a structured 3D world, providing a controllable and interpretable alternative to video-first generation.

AI USE STATEMENT

In this work, we used generative AI tools to assist with paper formatting, such as table formatting and typesetting. We have not used generative AI tools for method design, experimental design, result analysis, or technical writing. We have reviewed all AI-assisted work and take responsibility for the final content of this work.

ETHICS STATEMENT

This work studies image-to-4D task that carries significantly broader impact potential from both positive and negative sides. By generating 4D content, this work contributes to diverse applications like immersive virtual environment creation. However, it also raises ethical concerns related to privacy and misinformation, which must be carefully con sidered and addressed to ensure responsible deployment and minimize potential negative consequences.

REPRODUCIBILITY STATEMENT

For reproducibility, we provide the details needed to reproduce our results in the appendix. Method implementation and prompt details are given in Appendix A, including the models and tools used by each module, the motion planner prompt, the trajectory planner architecture and training details in Appendix A.2, and the scene composition procedure in Appendix A.3. Evaluation details are given in Appendix C, including all metrics, normalization bounds, and LLM judge prompts. Complete video results are included in the supplementary material via an HTML viewer.

REFERENCES

Jan Ackermann, Shengqu Cai, Boyang Deng, Zhengfei Kuang, Songyou Peng, and Gordon Wetzstein. Geoflow: Enforcing implicit geometric consistency in video generation. *arXiv preprint arXiv:2605.18365*, 2026.

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. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Feifei Bian, Zhimin Zheng, Wei Deng, Daiguo Zhou, and Jian Luan. Rs-gen: A multi-stage agentic framework for reasoning and search-augmented image generation, 2026. URL https://arxiv.org/abs/2606.23221.

Steve Bryson. Virtual reality: A definition history - a personal essay, 2013. URL https://arxiv.org/abs/1312.4322.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts, 2026. URL https://arxiv.org/abs/2511.16719.

Junyi Chen, Haoyi Zhu, Xianglong He, Yifan Wang, Jianjun Zhou, Wenzheng Chang, Yang Zhou, Zizun Li, Zhoujie Fu, Jiangmiao Pang, et al. Deepverse: 4d autoregressive video generation as a world model. *arXiv preprint arXiv:2506.01103*, 2025a.

Zhaoxi Chen, Tianqi Liu, Long Zhuo, Jiawei Ren, Zeng Tao, He Zhu, Fangzhou Hong, Liang Pan, and Ziwei Liu. 4dnex: Feed-forward 4d generative modeling made easy, 2025b. URL https://arxiv.org/abs/2508.13154.

Yankuan Chi, Xiang Li, Zixuan Huang, and James M. Rehg. Vinedresser3d: Agentic text-guided 3d editing, 2026. URL https://arxiv.org/abs/2602.19542.

Ioannis Deliyannis. *Interactive Multimedia*. IntechOpen, London, Mar 2012. doi: 10.5772/2394. URL https://doi.org/10.5772/2394.

Haoyi Duan, Hong-Xing Yu, Sirui Chen, Li Fei-Fei, and Jiajun Wu. Worldscore: A unified evaluation benchmark for world generation. *arXiv preprint arXiv:2504.00983*, 2025.

Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T Barron, and Ben Poole. Cat3d: Create anything in 3d with multi-view diffusion models. *arXiv preprint arXiv:2405.10314*, 2024.

Vinayak Gupta, Yunze Man, and Yuxiong Wang. Paintscene4d: Consistent 4d scene generation from text prompts. *https://arxiv.org/abs/2412.04471*, 2024.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: a reference-free evaluation metric for image captioning. In *EMNLP*, 2021.

Yi-Hua Huang, Zi-Xin Zou, Yuting He, Chirui Chang, Cheng-Feng Pu, Ziyi Yang, Yuan-Chen Guo, Yan-Pei Cao, and Xiaojuan Qi. Anigen: Unified s^3 fields for animatable 3d asset generation. *arXiv preprint arXiv:2604.08746*, 2026a.

Yinming Huang, Shuyuan Tu, Xi Yan, Jiahao Zhan, Zihan Yang, Zhen Xing, Hui Zhang, Tiehua Zhang, Yu-Gang Jiang, and Zuxuan Wu. Agentic visual generation: From generative models to agentic control, 2026b. URL https://arxiv.org/abs/2609.06758.

Jeonghwan Kim, Yushi Lan, Yongwei Chen, Hieu Trung Nguyen, Chuanyu Pan, and Xingang Pan. Sceneconductor: 3d scene generation from a single image with multi-agent orchestration, 2026. URL https://arxiv.org/abs/2606.08402.

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.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

Tianqi Liu, Zihao Huang, Zhaoxi Chen, Guangcong Wang, Shoukang Hu, Liao Shen, Huiqiang Sun, Zhiguo Cao, Wei Li, and Ziwei Liu. Free4d: Tuning-free 4d scene generation with spatial-temporal consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 25571–25582, October 2025.

Jun Luo, Jiaxiang Tang, Ruijie Lu, and Gang Zeng. Sceneassistant: A visual feedback agent for open-vocabulary 3d scene generation, 2026. URL https://arxiv.org/abs/2603.12238.

OpenAI. Gpt-5.6 sol. https://developers.openai.com/api/docs/models/gpt-5.6-sol, 2026. Accessed: 2026-09-23.

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.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

Uriel Singer, Shelly Sheynin, Adam Polyak, Oron Ashual, Iurii Makarov, Filippos Kokkinos, Naman Goyal, Andrea Vedaldi, Devi Parikh, Justin Johnson, and Yaniv Taigman. Text-to-4d dynamic scene generation, 2023. URL https://arxiv.org/abs/2301.11280.

Wenqiang Sun, Shuo Chen, Fangfu Liu, Zilong Chen, Yueqi Duan, Jun Zhang, and Yikai Wang. Dimensionx: Create any 3d and 4d scenes from a single image with controllable video diffusion. *arXiv preprint arXiv:2411.04928*, 2024.

Zachary Teed and Jia Deng. DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras. *Advances in neural information processing systems*, 2021.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. Exploring clip for assessing the look and feel of images. In *AAAI*, 2023.

Xingrui Wang, Xin Li, Yaosi Hu, Hanxin Zhu, Chen Hou, Cuiling Lan, and Zhibo Chen. Tiv-diffusion: Towards object-centric movement for text-driven image to video generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 7988–7996, 2025.

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, 2024. URL https://arxiv.org/abs/2411.18613.

Ying Yang, Zhengyao Lv, Yujia Zeng, Tianlin Pan, Haofan Wang, Yueming Lyu, Binxin Yang, Hubery Yin, Chen Li, Jing Lyu, Ziwei Liu, and Chenyang Si. Stableworld: Towards stable and consistent long interactive video generation. *arXiv preprint arXiv:2601.15281*, 2026.

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.

Jinyan Yuan, Bangbang Yang, Keke Wang, Panwang Pan, Lin Ma, Xuehai Zhang, Xiao Liu, Zhaopeng Cui, and Yuewen Ma. Immersegen: Agent-guided immersive world generation with alpha-textured proxies, 2026. URL https://arxiv.org/abs/2506.14315.

Guozhen Zhang, Chunxu Liu, Yutao Cui, Xiaotong Zhao, Kai Ma, and Limin Wang. Vfimamba: Video frame interpolation with state space models, 2024a. URL https://arxiv.org/abs/2407.02315.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arxiv:2410.03825*, 2024b.

Yi Zhang, Yunshuang Wang, Zeyu Zhang, and Hao Tang. Code2worlds: Empowering coding llms for 4d world generation, 2026a. URL https://arxiv.org/abs/2602.11757.

Zekai Zhang, Jiahao Li, Jie Zhang, Kaiyuan Gao, Kun Yan, Lihan Jiang, Ningyuan Tang, Shengming Yin, Tianhe Wu, Xiaoyue Chen, Xiao Xu, Yan Shu, Yanran Zhang, Yixian Xu, Yuxiang Chen, Zhendong Wang, Zihao Liu, Zikai Zhou, Huishuai Zhang, Dongyan Zhao, and Chenfei Wu. Qwen-image-agent: Bridging the context gap in real-world image generation, 2026b. URL https://arxiv.org/abs/2606.26907.

Yuyang Zhao, Zhiwen Yan, Enze Xie, Lanqing Hong, Zhenguo Li, and Gim Hee Lee. Animate124: Animating one image to 4d dynamic scene. *arXiv preprint arXiv:2311.14603*, 2023.

Yuyang Zhao, Chung-Ching Lin, Kevin Lin, Zhiwen Yan, Linjie Li, Zhengyuan Yang, Jianfeng Wang, Gim Hee Lee, and Lijuan Wang. Genxd: Generating any 3d and 4d scenes. In *ICLR*, 2025.

Dewei Zhou, Xinyu Huang, Xun Wang, Ji Xie, Yabo Zhang, Liang Li, Kunchang Li, Zongxin Yang, and Yi Yang. Metapoint: Unlocking precise spatial control in agentic visual generation, 2026a. URL https://arxiv.org/abs/2606.05031.

Hengji Zhou, Lingxuan Huang, Jian Wang, Bing Zhou, Si Wu, Lianghao Xia, and Chao Huang. Videoagent: All-in-one framework for video understanding and editing, 2026b. URL https://arxiv.org/abs/2606.23327.

APPENDIX FOR INFUSING MOTION INTELLIGENCE INTO STRUCTURED 3D WORLDS FOR 4D SCENE GENERATION

This appendix provides extra information organized as follows.

- Implementation Details:
  - Models/tools/prompts used by each module of SceneWeaver and detailed implementation process.(in Sec.A.1)
  - Trajectory Planning Model details regarding its Architecture, Data construction and Training. (in Sec.A.2)
  - Details about the Scene Composition Process. (in Sec.A.3)
- More Related Work Discussion. (in Sec.B)
- Metric Implementation Details. (in Sec.C)
- Additional Visualizations of complete video frames. (in Sec.D). For video visualization, please open the HTML viewer in the supplementary material.

A IMPLEMENTATION DETAILS

A.1 MODELS DETAILS

We summarize the concrete models and tools used by each module of SceneWeaver below.

Scene Decomposition. Given the input image $\mathcal{I}$, we first use SAM3Carion et al. (2026) to segment the subject to be controlled, remove the subject with Stable Diffusion InpaintRombach et al. (2021) tool to obtain a clean background $\mathcal{I}_{bg}$, then extrapolate it into a 360° panorama $\mathcal{I}_{pano}$ and lift the panorama into a 3D environment mesh $\mathcal{M}_{bg}$ with model HunyuanWorld Lai et al. (2025).

3D subject reconstruction and rigging. The static subject mesh $\mathcal{M}_s$ is generated from subject image by AniGenHuang et al. (2026a), which also rigged it with an articulated skeleton $\mathcal{K}$ and the skinning weight $\mathcal{W}$.

LLM for motion intent planning. The Motion Generation Agent uses an LLM as a high-level intent planner. The detailed prompt is provided in Table.4.

A.2 TRAJECTORY PLANNING: ARCHITECTURE, TRAINING, AND DATA

Principle. The Trajectory Executor plans the subject's world-space motion as a continuous dynamical system rather than regressing a sequence of waypoints. The state evolution is decomposed into three additive terms:

$$\frac{d\mathbf{s}}{dt} = \mathcal{F}_{\text{phys}}(\mathbf{s}, \Phi_{\text{env}}) + \mathcal{F}_{\text{terrain}}(\mathbf{x}, \phi_b) + G \cdot \mathbf{u}(t), \tag{5}$$

where $\mathbf{s}$ denotes the subject state, Φ_{env} the environmental condition, ϕ_b the signed distance field (SDF) of the background world, and $\mathbf{u}(t)$ the user control signal. The three terms have distinct roles: $\mathcal{F}_{\text{phys}}$ is a learnable field that captures the physical effects; $\mathcal{F}_{\text{terrain}}$ is an analytic potential-field term derived from the scene geometry, which provides obstacle repulsion and ground contact; and $G \cdot \mathbf{u}(t)$ is a fixed linear mapping that injects explicit user intent. This decomposition keeps the model interpretable and allows the terrain, environmental condition, and control signal to be swapped at inference time without retraining.

State, control, and environment representation. The subject state is

$$\mathbf{s} = [\mathbf{x}, \mathbf{v}, \mathbf{a}] \in \mathbb{R}^9, \tag{6}$$

where $\mathbf{x} = (x, y, z)$ is the world-space position, $\mathbf{v} = (v_x, v_y, v_z)$ is the linear velocity, and $\mathbf{a} = (a_x, a_y, a_z)$ is the acceleration. Including acceleration in the state allows a first-order response model to constrain acceleration continuity and avoid unrealistic instantaneous jumps. The control input is

$$\mathbf{u} = [u_W, u_A, u_S, u_D] \in \mathbb{R}^4, \tag{7}$$

Table 4: Summary of the system prompt used by the Motion Planner. The planner takes a rigged mesh and a fine-grained part-level motion description as input, and generates a structured JSON motion program.

Aspect	Description
Role	The planner acts as a character animator. Given the skeleton of a rigged 3D asset in its canonical frame and a fine-grained, part-level description of the desired motion, it generates a JSON *motion program* that realizes that description.
Input	The input consists of (i) a rigged mesh with its articulated skeleton in the canonical frame, and (ii) a fine-grained prompt that specifies the intended motion for each body part.
Canonical frame	The canonical frame is right-handed with $+Z$ = up. The asset's horizontal heading is not fixed; body-forward, lateral, and up axes are provided and used as rotation axes. "Side" is measured along the lateral axis: L = positive lateral, R = negative lateral, C = center.
Motion representation	Motion is authored as per-joint *delta rotations* applied on top of the fitted pose. Each joint edit rotates that joint and its whole subtree about a chosen global canonical axis, by the right-hand rule (positive angle = counter-clockwise looking down the $+$axis).
Joint selection	Rotate the joint that is the *root* of the subtree to be moved (e.g., a shoulder to move the whole arm, a hip to move the whole leg). For a limb, pick the joint out along the limb whose child direction points away from the torso, not the small clavicle/pelvis stub near the spine. For symmetric both-sides motion, pick a true mirror pair so the two sides match, and use opposite signs for the left and right joint.
Sign and axis choice	Choose the axis and sign so that each part moves as the part-level description implies; use the child direction to get the sign right. For bilateral (symmetric) motions, use opposite signs on the left and right joints.
Locomotion semantics	For ordinary walk/stride/trot/gait requests, the preview is an in-place articulation study: do not add root translation and do not move the whole torso forward or backward. "Walk forward" describes the leg cycle unless the request explicitly says to travel across the scene. A natural four-legged walk uses four distinct footfall phases; same-side legs should not move together, and a single rigid-body translation should not be used to fake locomotion. Add root translation only for an explicit hop, flight, swim travel, or explicit scene travel, and keep it small.
Cyclic consistency	Every motion must return to or cycle through the start, so that frame 0 equals the original pose.
Output format	The planner returns only a JSON object with a short description, a list of joint edits, and an optional root translation. Each joint edit specifies the joint index, the axis (forward/lateral/up), a signed amplitude in degrees, a curve type, optional repetition count, and a comment describing the intended body part or intent. Amplitudes should be physically plausible (e.g., an arm raised fully overhead from the side is about 140–160°). Every joint needed for the motion should be included, with both sides for symmetric motion.

where each component indicates whether the corresponding key is active. The control term is mapped to a world-space acceleration by

$$\mathbf{a}_{\text{control}} = a_c \begin{bmatrix} u_W - u_S \\ -u_A + u_D \\ 0 \end{bmatrix}, \tag{8}$$

where a_c is a fixed control gain. When multiple keys are pressed, the direction vector is summed and the resulting control acceleration is clipped to a maximum magnitude. Thus W maps to $+x$ (forward), S to $-x$, A to $-y$ (left), and D to $+y$ (right).

The environmental condition is not represented as three separate, independently oriented quantities. Gravity, friction, and wind are integrated into a single combined environmental acceleration,

$$\Phi_{\text{env}} = \mathbf{a}_{\text{grav}} + \mathbf{a}_{\text{fric}} + \mathbf{a}_{\text{wind}} \in \mathbb{R}^3, \tag{9}$$

where $\mathbf{a}_{\text{grav}}$ is the gravitational acceleration, and $\mathbf{a}_{\text{fric}}$ and $\mathbf{a}_{\text{wind}}$ are the accelerations contributed by friction and wind. During training data construction, $\mathbf{g}$, the friction coefficient, and the wind vector

are sampled independently and then mapped to Φ_{env}, so the model is exposed to a wide range of environmental conditions.

Model architecture and usage. The learnable physics field $\mathcal{F}_{\text{phys}}$ is parameterized by a MLP network that predicts a residual acceleration from the current state and the environmental condition. The input to the network is the concatenation of the normalized state and the normalized environmental condition,

$$\mathbf{z} = [\tilde{\mathbf{s}}, \tilde{\Phi}_{\text{env}}] \in \mathbb{R}^{12}, \tag{10}$$

where

$$\tilde{\mathbf{s}} = \frac{\mathbf{s} - \mu_s}{\sigma_s}, \qquad \tilde{\Phi}_{\text{env}} = \frac{\Phi_{\text{env}} - \mu_\Phi}{\sigma_\Phi}, \tag{11}$$

and $\mu_s, \sigma_s, \mu_\Phi, \sigma_\Phi$ are statistics over its past state. The network consists of an input linear layer, a stack of residual blocks, and an output linear layer. Each residual block contains two linear layers with SiLU activation and a skip connection. The output is a three-dimensional residual acceleration,

$$\mathbf{F}_{\text{phys}} = \mathcal{F}_{\text{phys}}(\mathbf{s}, \Phi_{\text{env}}) \in \mathbb{R}^3, \tag{12}$$

which represents the portion of the acceleration that is not explained by gravity, the user control, or the terrain potential. This residual formulation is advantageous because it lets the network focus its capacity on the complex physical effects that are not captured by the analytic terms, while the well-understood components: force influence, user control, and the terrain potential, are handled exactly by their closed-form expressions. As a result, the learned field only needs to model the remaining dynamics, which improves both the interpretability of the decomposition and the fidelity of the simulated physical effects.

The terrain term is computed analytically from the SDF of the background world. Following the potential-field formulation used in the main paper, we write

$$\mathcal{F}_{\text{terrain}}(\mathbf{x}, \phi_b) = k_{\text{rep}}\, \nabla\phi_b(\mathbf{x})\, \exp\!\left(-\frac{\text{clip}(\phi_b(\mathbf{x}), -1, 5)}{\sigma}\right), \tag{13}$$

where ϕ_b is the background SDF, $\nabla\phi_b$ is its gradient, and k_{rep} and σ control the strength and decay of the potential field. This term pushes the subject away from terrain obstacles and toward free space, and it also provides the ground-contact response through the same SDF gradient. SDF queries use trilinear interpolation on a 3D grid and are batched for efficiency. The control term is a fixed linear mapping from the WASD signal to a world-space acceleration.

Given the three terms, the full state derivative used by the model is

$$\dot{\mathbf{x}} = \mathbf{v}, \qquad \dot{\mathbf{v}} = \mathcal{F}_{\text{phys}}(\mathbf{s}, \Phi_{\text{env}}) + \mathcal{F}_{\text{terrain}}(\mathbf{x}, \phi_b) + \mathbf{a}_{\text{control}}, \tag{14}$$

$$\dot{\mathbf{a}} = \frac{\mathbf{a}_{\text{target}} - \mathbf{a}}{\tau}, \tag{15}$$

where the acceleration state follows a first-order response toward the target acceleration with time constant τ. The model is a Neural ODE in the sense that it defines a continuous-time vector field $f_\theta(\mathbf{s}, t)$ and produces the trajectory by integrating this field, rather than by applying a discrete transition function. Concretely, the ODE structure is realized in three places: (i) the state derivative is assembled from the kinematic relation $\dot{\mathbf{x}} = \mathbf{v}$, the learned residual $\mathcal{F}_{\text{phys}}$, the state-dependent terrain potential $\mathcal{F}_{\text{terrain}}(\mathbf{x}, \phi_b)$, and the time-varying control term $G\mathbf{u}(t)$; (ii) the acceleration state is evolved by a first-order response $\dot{\mathbf{a}} = (\mathbf{a}_{\text{target}} - \mathbf{a})/\tau$, which keeps the ODE first-order while enforcing acceleration continuity; and (iii) the trajectory is obtained by a fixed-step RK4 integrator that re-evaluates the vector field at every substep, so the terrain and learned residual act at every integration stage. Because the terrain potential depends on the current position and the control term depends on time, the system is a non-autonomous, state-dependent ODE, and the resulting trajectory is shaped jointly by the analytic physics, the learned residual, and the user control signal.

Input and output format. The Trajectory Executor receives:

- the initial state $\mathbf{s}_0 = [\mathbf{x}_0, \mathbf{v}_0, \mathbf{a}_0]$;
- the environmental condition $\Phi_{\text{env}} \in \mathbb{R}^3$;
- the SDF ϕ_b and its precomputed gradient of the background world;

- the control sequence $\mathcal{U} = \{(\tau_k, \mathbf{u}_k)\}$, where τ_k is the time at which the control signal $\mathbf{u}_k$ becomes active.

It outputs the continuous world-space trajectory

$$\mathbf{s}(t) = [\mathbf{x}(t), \mathbf{v}(t), \mathbf{a}(t)]^\top, \quad t \in [0, T], \tag{16}$$

from which the subject's position $\mathbf{x}(t)$ is used for composition with the local articulated motion. The overall procedure is summarized in Algorithm A.2.

[t] [1] Initial state $\mathbf{s}_0 = [\mathbf{x}_0, \mathbf{v}_0, \mathbf{a}_0]$, environmental condition Φ_{env}, background SDF ϕ_b, control sequence $\mathcal{U} = \{(\tau_k, \mathbf{u}_k)\}$, horizon T, step size Δt Continuous trajectory $\mathbf{s}(t) = [\mathbf{x}(t), \mathbf{v}(t), \mathbf{a}(t)]^\top$ over $[0, T]$ Initialize $\mathbf{s} \leftarrow \mathbf{s}_0$ Define the learnable physics field $\mathcal{F}_{\text{phys}}$ as a residual MLP $t < T$ Query the SDF and its gradient at $\mathbf{x}$: $\phi \leftarrow \phi_b(\mathbf{x})$, $\nabla\phi \leftarrow \nabla\phi_b(\mathbf{x})$ Interpolate the control signal at time t: $\mathbf{u}(t) \leftarrow \text{Interpolate}(\mathcal{U}, t)$ Compute the terrain potential force:

$$\mathcal{F}_{\text{terrain}} = -k_{\text{rep}}\, \nabla\phi(x)\, \exp\left(-\frac{\text{clip}(\phi, -1, 5)}{\sigma}\right)$$

Compute the learned physical residual:

$$\mathbf{F}_{\text{phys}} = \mathcal{F}_{\text{phys}}(\mathbf{s}, \Phi_{\text{env}})$$

Form the state derivative:

$$\dot{\mathbf{s}} = [\mathbf{v},\ \mathbf{g} + \mathbf{F}_{\text{phys}} + \mathcal{F}_{\text{terrain}} + \mathbf{a}_{\text{control}},\ (\mathbf{a}_{\text{target}} - \mathbf{a})/\tau]^\top$$

Advance the state with one RK4 step of size Δt, recomputing the terrain force and the learned residual at every substep $t \leftarrow t + \Delta t$ $\mathbf{s}(t)$ over $[0, T]$

Data construction. Training trajectories are procedurally generated by sampling initial states, environmental parameters, and control sequences, and then simulating the resulting motion under the corresponding physical laws. The dataset covers a diverse range of terrain types, including flat ground, slopes, rock clusters, canyon terrain, and indoor obstacle layouts. Beyond discrete obstacles, the terrain itself contains height variations that simulate the uneven ground and environmental conditions encountered in more realistic worlds, so that the learned dynamics must account for both local surface geometry and global terrain shape. Terrain is generated by analytic height functions, and obstacles are represented by sphere, box, and cylinder SDFs. The scene SDF is the minimum of all primitive SDFs:

$$\phi_b(\mathbf{p}) = \min\left(\phi_{\text{ground}}(\mathbf{p}), \phi_1(\mathbf{p}), \ldots, \phi_K(\mathbf{p})\right), \tag{17}$$

with the convention $\phi_b > 0$ in free space, $\phi_b = 0$ on the surface, and $\phi_b < 0$ inside solids. SDF gradients are precomputed by central differences on a voxel grid and stored with the SDF.

To sample a trajectory, we first sample an initial state $\mathbf{s}_0 = [\mathbf{x}_0, \mathbf{v}_0, \mathbf{a}_0]$ on or near the terrain surface, together with an environmental condition Φ_{env} and a control sequence $\mathcal{U} = \{(\tau_k, \mathbf{u}_k)\}$. The initial position is drawn from a valid free-space region above the terrain, and the control sequence is sampled so that the resulting motion covers a range of directions, speeds, and turning patterns. The simulation dynamics are

$$\dot{\mathbf{x}} = \mathbf{v}, \qquad \dot{\mathbf{v}} = \mathbf{g} + \mathbf{w} - \mu\mathbf{v} + \mathbf{a}_{\text{control}} + \mathcal{F}_{\text{terrain}}, \tag{18}$$

with first-order acceleration response

$$\dot{\mathbf{a}} = \frac{\mathbf{a}_{\text{target}} - \mathbf{a}}{\tau}. \tag{19}$$

Training. We train our Trajectory Executor with construction data. The main loss is a weighted state loss,

$$\mathcal{L}_{\text{state}} = \lambda_x \mathcal{L}_x + \lambda_v \mathcal{L}_v + \lambda_a \mathcal{L}_a, \tag{20}$$

which measures the difference between predicted and ground-truth states. We also include two auxiliary terms. The dynamics consistency loss compares predicted and ground-truth velocity increments and mechanical energy changes:

$$\mathcal{L}_{\text{momentum}} = \|(\mathbf{v}_{t+1} - \mathbf{v}_t)_{\text{pred}} - (\mathbf{v}_{t+1} - \mathbf{v}_t)_{\text{gt}}\|_2^2, \tag{21}$$

$$E_t = \frac{1}{2}\|\mathbf{v}_t\|^2 + \|\mathbf{g}\|z_t, \qquad \mathcal{L}_{\text{inv}} = \mathcal{L}_{\text{momentum}} + \lambda_E \mathcal{L}_{\text{energy}}. \tag{22}$$

The acceleration smoothness loss penalizes abrupt acceleration changes:

$$\mathcal{L}_{\text{reg}} = \frac{1}{T}\sum_t \|\mathbf{a}_{t+1} - \mathbf{a}_t\|_2^2. \tag{23}$$

The final loss is

$$\mathcal{L} = \mathcal{L}_{\text{state}} + \lambda_{\text{inv}}\mathcal{L}_{\text{inv}} + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}}. \tag{24}$$

A.3 Details about the Scene Composition .

For animated subject and background world composition, we first need to placed it back into the background world at the initial position and scale it occupied in the input image, then controlling it move along the trajectory. Let $\mathcal{M}_{src}$ be the source mask marking the subject in the input image, and $\mathcal{M}_{cond}$ the condition mask marking the region the source image occupies on the background panorama. Mapping $\mathcal{M}_{src}$ into the region delimited by $\mathcal{M}_{cond}$ yields a target bounding box $\mathcal{B}$ on the panorama, from which we read the corresponding background vertices and take their median as the 3D anchor $\mathbf{p}$ at depth $d = \|\mathbf{p}\|$. The apparent size in the panorama is converted to world units by

$$w = 2\,d\tan\left(\frac{\Delta\theta_w}{2}\right), \qquad h = 2\,d\tan\left(\frac{\Delta\theta_h}{2}\right), \tag{25}$$

where $\Delta\theta_w, \Delta\theta_h$ are the angular extents of $\mathcal{B}$. The subject mesh is scaled by the ratio between this target size and its own extents, and its projected bounding box is centered on $\mathcal{B}$. A light geometric refinement then settles the subject onto the terrain, aligning its up-axis with the local surface normal and applying an upward correction where the footprint penetrates the ground.

B More Related Work Discussion

In the main paper we compare against two lines of methods: *image-to-4D generation*, including Free4D Liu et al. (2025), DeepVerse Chen et al. (2025a), 4DNeX Chen et al. (2025b); *video generation*, including Wan2.2-I2V-14B Wan et al. (2025) and CogVideoX-I2V-5B Yang et al. (2024). Here we discuss more Image-to-4D methods are not directly comparable for the reasons detailed below. **Not fully open-resourced:** GenXD Zhao et al. (2025), Dimension-X Sun et al. (2024), and PaintScene4D Gupta et al. (2024) do not release complete code or checkpoints sufficient(for example, only provide the static 3D scene generation part code and checkpoint), so we can not compare with them. **Not supporting scene-level 4D generation:** Animate124 Zhao et al. (2023) focuses on animating a single object rather than generating a scene-level 4D representation. Since our task requires a subject moving within an explicit 3D background world, its formulation does not directly apply. **Target for text-based 4D generation:** More works like Code2WorldZhang et al. (2026a) and ImmerseGenYuan et al. (2026) propose to formulate 4D generation as code programming process. However, they only accept textual input, which can not be fairly compared with us.

C Metric Details

C.1 WorldScore Metrics

We introduce the metric details here. Some metrics are adopted from WorldScore, evaluating the generated scene quality from perspectives of controllability, quality, and dynamics. We also include metrics to evaluate the trajectory quality and environment awareness. We provide a complete description of each metric below, including formulation details and implementation specifics.

C.1.1 Controllability

We have two metrics to assess controllability.

Object controllability. We evaluate whether the objects specified in the prompt $\mathcal{P}$ consistently appear in the generated next scene. To this end, we measure the success rate of object detection. Specifically, we leverage a state-of-the-art open-set object detection model (Liu et al., 2023). We extract one or two individual object descriptions from the text prompt $\mathcal{N}$. We compute the success rate by matching the detected objects with the object descriptions. This provides a quantitative measure of how well the generated foreground objects adhere to the world specification.

Content alignment. Besides the objects, we also assess whether the generated scenes are aligned with the entire text $\mathcal{P}$ using CLIPScore Hessel et al. (2021) for video frame-text alignment calculation.

C.1.2 Quality

We use two metrics to assess quality.

3D consistency. We evaluate the 3D consistency in the static world videos. This metric focuses on how the geometry of a scene remains stable across frames, regardless of slight changes in visual textures. To this end, we use DROID-SLAM (Teed & Deng, 2021), a standard SLAM method, to estimate dense pixel-wise depth for each frame. One key advantage of DROID-SLAM is its dense nature, employing a differentiable Dense Bundle Adjustment (DBA) layer that continuously refines camera poses and dense, per-pixel depth estimates.

We calculate the reprojection error between a pair of co-visible pixels in consecutive frames after DBA layer refinement:

$$e_{\text{reproj}} = \frac{1}{|\mathcal{V}|} \sum_{(i,j)\in\mathcal{V}} \left\| \mathbf{p}_{ij}^{*} - \Pi(\mathbf{P}_{ij}) \right\|_2, \tag{26}$$

where $\mathcal{V}$ denotes the valid set of co-visible points, $\mathbf{p}_{ij}^{*}$ is the observed point on the ground truth image, $\mathbf{P}_{ij}$ is the reconstructed 3D point, obtained from refined depth and camera pose, and $\|\cdot\|_2$ calculates the Euclidean distance. Since DROID-SLAM is designed to be robust against appearance changes, this metric effectively measures geometric inconsistency.

Subjective quality. We use automatic metrics to evaluate the human perceptual quality of the generated scenes. To find a combination that best fits human perception, we performed a human study of 400 participants and found that the combination (arithmetic mean) of CLIP-IQA+ Wang et al. (2023) and CLIP Aesthetic best matches human preference.

C.1.3 Dynamics

We have two metrics to assess dynamics capabilities.

Motion magnitude. We measure a world generation model's ability to create large motions by estimating the optical flow between the consecutive frames of the generated video. We calculate $s_{\text{motion-mag}}$, defined as the median value of all the entries of the optical flow magnitude matrix $\mathbf{F}$. The final metric is the average of $s_{\text{motion-mag}}$ across all pairs of consecutive frames.

Motion smoothness. We evaluate the temporal smoothness of generated videos $\mathbf{V}$ by leveraging motion priors from a standard video frame interpolation model (Zhang et al., 2024a). We drop odd-indexed frames from the generated video and use the interpolation model to infer them. Then, we compute the mean squared error, SSIM, and LPIPS between the reconstructed frames and the original dropped frames. After normalization, we average these scores to get the motion smoothness metric.

After computing individual evaluation metrics, we apply a linear normalization and mapping process:

$$s^{\text{norm}} = \begin{cases} \left\langle \frac{s - b^{\min}}{b^{\max} - b^{\min}} \right\rangle, & \text{if higher better,} \\ \left\langle 1 - \frac{s - b^{\min}}{b^{\max} - b^{\min}} \right\rangle, & \text{if lower better,} \end{cases} \tag{27}$$

where s denotes the raw value, $b^{\min}$ and $b^{\max}$ denote the empirical bounds, and $\langle\cdot\rangle$ denotes the clip function to range $[0, 1]$.

Aggregation. The normalized scores are scaled by 100. We compute the arithmetic mean of the dimension scores within control and quality aspects to obtain Static. Additionally, incorporating the three dynamics dimension scores results in Dynamic. For 3D scene generation models that do not support dynamic tasks, we assign 0 to each dynamics metric.

Trajectory Quality. Trajectory Quality measures whether the generated global movement follows the user instruction. Specifically, it evaluates whether the subject moves in the intended direction, whether the overall path is completed as requested, and whether the motion stays within the valid region of the environment. Since these properties are semantic and relational rather than purely numerical, we evaluate them with an LLM-based judge that is given the user instruction, the rendered video of the scene. The judge returns a score together with a short justification, and we average the scores over all evaluated cases. The corresponding prompt is summarized in Table 5.

Environmental Awareness. Environmental Awareness measures whether the generated dynamics responds appropriately to the environmental conditions. It evaluates whether the subject exhibits the expected physical and geometric behavior, such as responding to gravity, friction, wind, and terrain slope, avoiding obstacles, and not exhibiting artifacts like penetration or floating. It also checks whether the generated motion reflects the environmental effects implied by the user instruction, for example wind visibly moving foliage, or friction causing an object to slow down and eventually stop. We again use an LLM-based judge, which receives the user instruction and the rendered video, and returns a score with a short justification. The corresponding prompt is summarized in Table 6.

Table 5: Prompt template for LLM-based Trajectory Quality evaluation. The placeholders in angle brackets are filled with the user instruction, the generated trajectory, and the rendered scene.

Field	Content
Role	You are an evaluator for 4D scene generation. You will be given a user instruction, the generated world-space trajectory of the subject, and a rendering of the generated scene. Your task is to judge whether the generated global movement follows the user instruction.
Input	`User instruction: <instruction>` `Rendered scene: <rendered scene>`
Evaluation criteria	Please consider the following aspects: 1. **Direction alignment**: Does the subject move in the direction implied by the instruction and the intended control signal? 2. **Path completion**: Does the subject complete the intended path, rather than stopping early, drifting away, or moving in an unrelated direction? 3. **Boundary compliance**: Does the subject stay within the valid region of the environment, without leaving the scene or entering invalid areas? 4. **Overall consistency**: Is the global movement consistent with the user instruction as a whole?
Scoring	Give a score from 1 to 100, where 1 means the generated movement clearly violates the instruction and 100 means it follows the instruction faithfully. Provide a short justification for your score.
Output format	Return a JSON object with two fields: `score` and `reason`.

D Additional Visualizations

We provide the complete video results in the supplementary material via a HTML viewer. Below we also provide the complete video frames visualizations from Figure.5 to Figure.13.

Table 6: Prompt template for LLM-based Environmental Awareness evaluation. The placeholders in angle brackets are filled with the user instruction, the generated result, and the environmental conditions.

Field	Content
Role	You are an evaluator for 4D scene generation. You will be given a user instruction, the generated 4D result, and the environmental conditions. Your task is to judge whether the generated dynamics responds appropriately to the environment.
Input	`User instruction: <instruction>` `Generated result: <rendered scene or sampled frames>`
Evaluation criteria	Please consider the following aspects: 1. **Physical plausibility**: Does the result avoid physical and geometric artifacts such as penetration, floating, or implausible deformation? 2. **Environmental response**: Does the subject respond to the environment as expected, e.g., falling under gravity, slowing down or stopping under friction, drifting under wind, and adapting to terrain slope? 3. **Instruction grounding**: Does the result reflect the environmental effects implied by the user instruction, e.g., wind visibly moving foliage, or friction causing an object to slow down and stop? 4. **Overall consistency**: Is the overall dynamics consistent with the environmental conditions and the user instruction?
Scoring	Give a score from 1 to 100, where 1 means the result clearly violates the environment or the instruction and 100 means it responds to the environment faithfully. Provide a short justification for your score.
Output format	Return a JSON object with two fields: `score` and `reason`.



Prompt: the cat jumps down from the sofa

Figure 5: Complete video frames visualizations 1



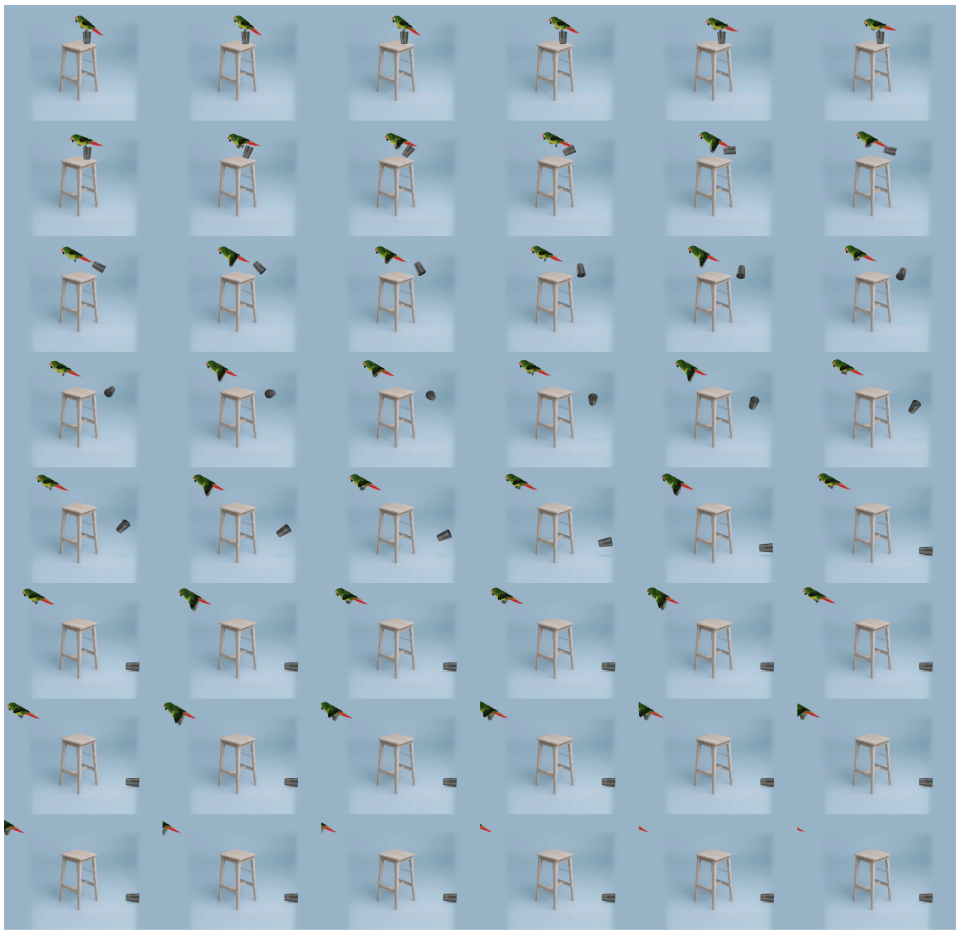
Prompt: The eagle flies in a windy environment, turn to its left direction.

Figure 6: Complete video frames visualizations 2



Prompt: The ball rolls up the slope and falls.

Figure 7: Complete video frames visualizations 3



Prompt: The bird flew away, knocking down the cup.

Figure 8: Complete video frames visualizations 4



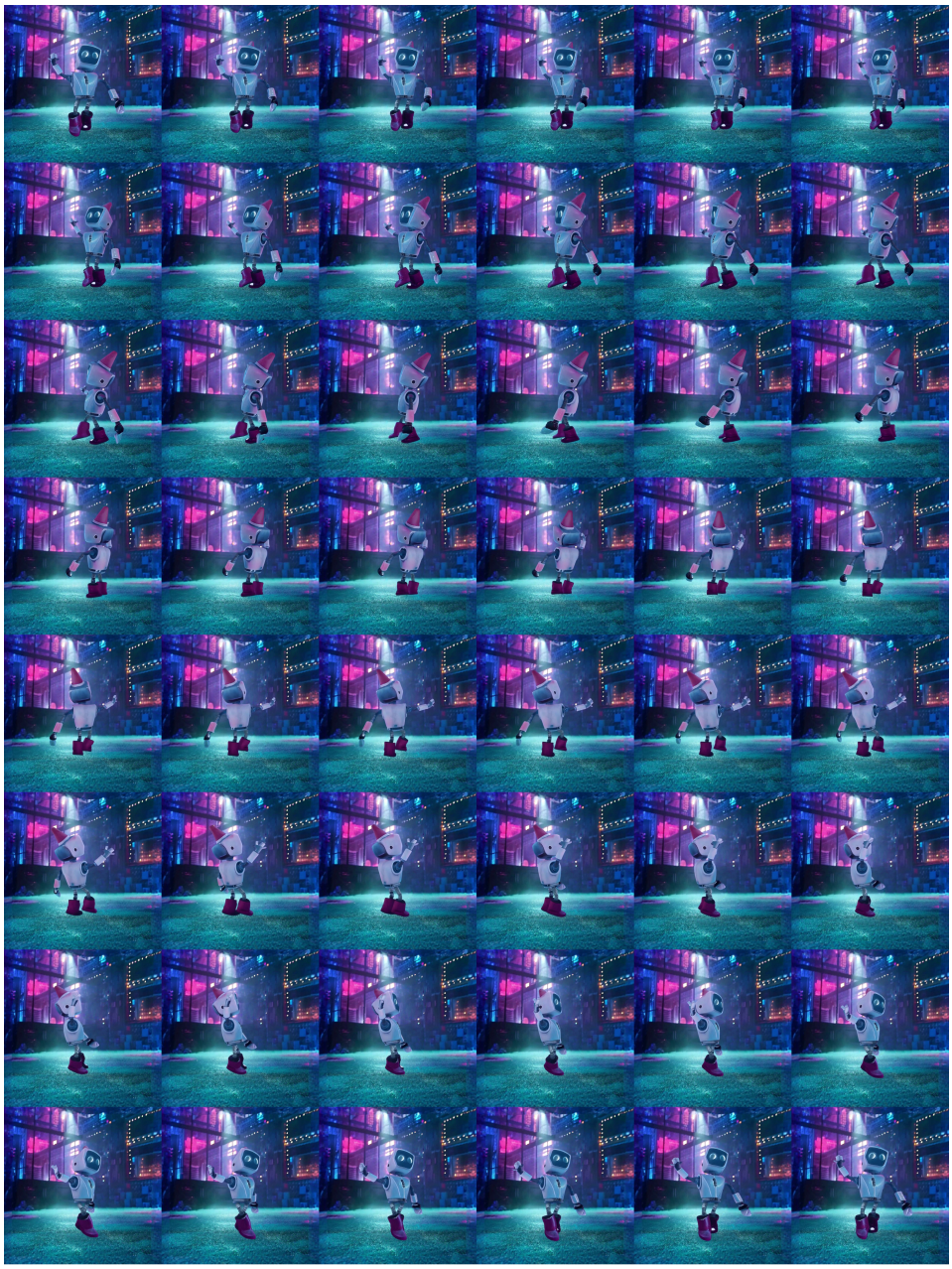
Prompt: Prompt: (1) The dog walks towards river and waves tail.(2) The man walks all crooked with head turning around.(3) The bird is flying in a straight line.

Figure 9: Complete video frames visualizations 5



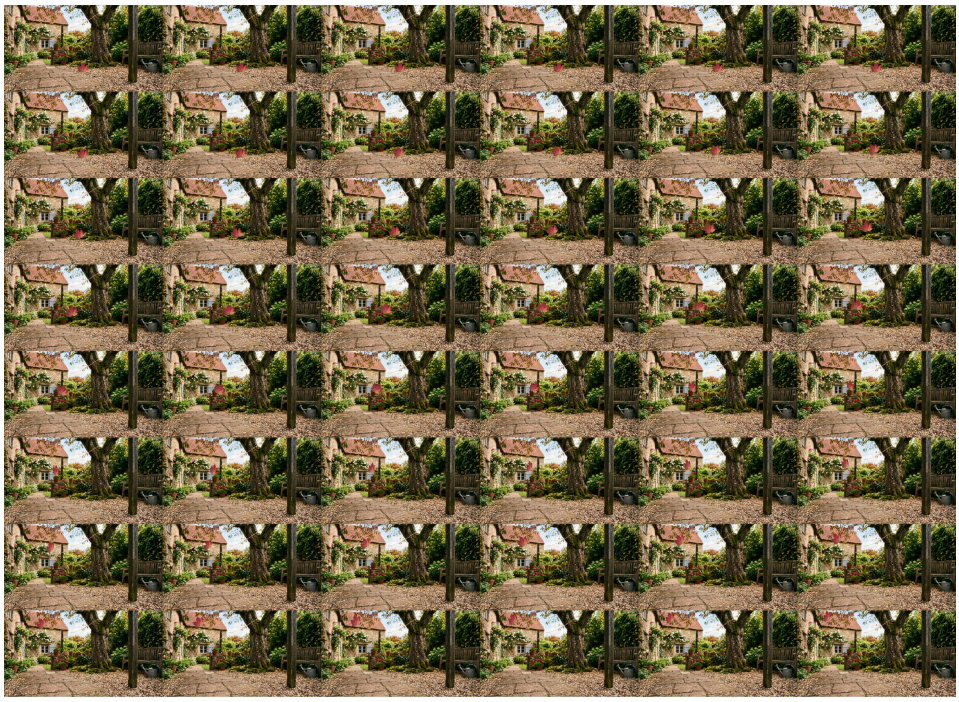
Prompt: The car skidded forward on the rough road.

Figure 10: Complete video frames visualizations 6



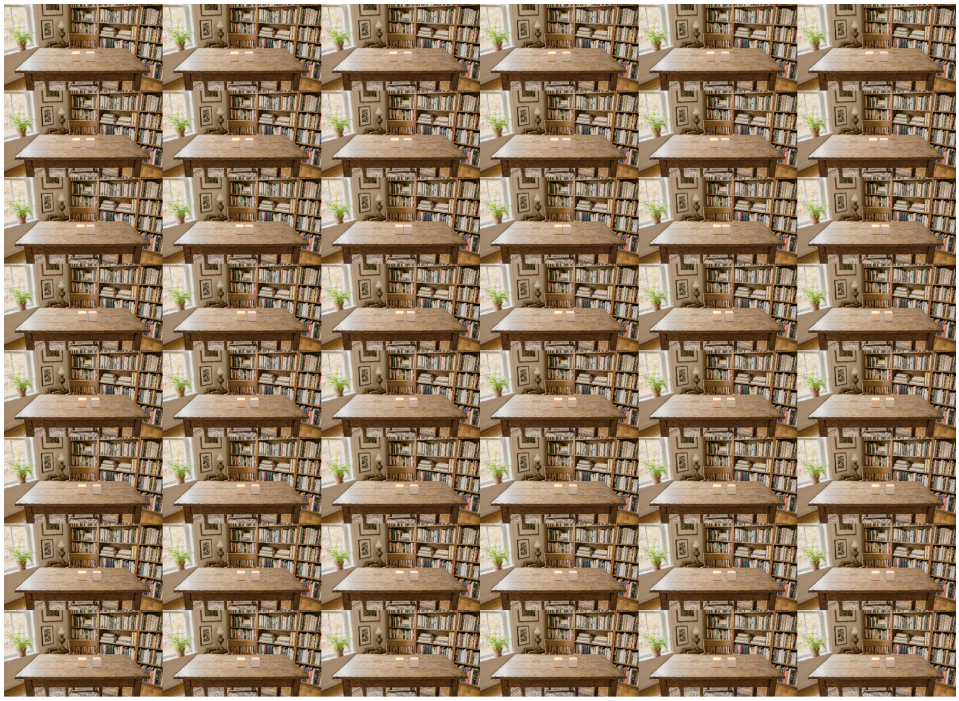
Prompt: The robot spun in place, dancing and waving its arms.

Figure 11: Complete video frames visualizations 7



Prompt: A gust of wind blew from the right, sending the maple leaves up into the sky.

Figure 12: Complete video frames visualizations 8



Prompt: The left block slides to the right and hits the right block.

Figure 13: Complete video frames visualizations 9