ActionParty: Multi-Subject Action Binding in Generative Video Games

Alexander Pondaven[1,2] *, Ziyi Wu[3], Igor Gilitschenski[3], Philip Torr[2], Sergey Tulyakov[1], Fabio Pizzati[4], and Aliaksandr Siarohin[1]

[1] Snap Research, USA
[2] University of Oxford, UK
[3] University of Toronto, Canada
[4] MBZUAI, UAE

https://action-party.github.io/

Abstract. Recent advances in video diffusion have enabled the development of "world models" capable of simulating interactive environments. However, these models are largely restricted to single-agent settings, failing to control multiple agents simultaneously in a scene. In this work, we tackle a fundamental issue of action binding in existing video diffusion models, which struggle to associate specific actions with their corresponding subjects. For this purpose, we propose ActionParty, an action controllable multi-subject world model for generative video games. It introduces *subject state* tokens, *i.e.* latent variables that persistently capture the state of each subject in the scene. By jointly modeling state tokens and video latents with a spatial biasing mechanism, we disentangle global video frame rendering from individual action-controlled subject updates. We evaluate ActionParty on the Melting Pot benchmark, demonstrating the first video world model capable of controlling up to seven players simultaneously across 46 diverse 2D game environments. Our results show significant improvements in action-following accuracy and identity consistency, while enabling robust autoregressive tracking of subjects through complex interactions.

Keywords: Video generation · World models · Action binding

1 Introduction

The emergence of video diffusion models [6, 7, 30–32, 70] has shifted the focus of generative AI from static content creation to the development of "world models", simulators capable of predicting future observations conditioned on interactive actions [8, 27, 33]. By learning the underlying dynamics of environments directly from pixels, recent breakthroughs such as Genie [3] and GameNGen [69] have demonstrated the potential for generating high-fidelity, interactive video games.

* Work done during internship at Snap Inc. Corr: pondaven@robots.ox.ac.uk

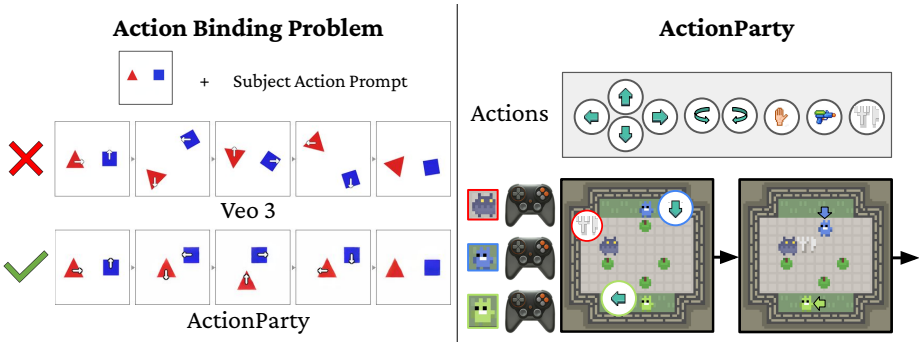


Fig. 1: Left: text-to-video models such as Veo 3 [63] fail to bind actions to the correct subject over a multi-step command sequence (full prompt in the supplementary material). Right: ActionParty enables action control of multiple subjects in a scene.

Despite these advances, existing world models are often restricted to single-agent settings with an egocentric output view [3]. They accept as input a single stream of control signals and can only control a single subject in the scene. Consequently, these generated "worlds" are unable to simulate the multi-agent dynamics that define complex social environments such as in self-driving [34, 66] and robotics [4, 13] applications.

In this work, we aim to explore multi-subject control in video world models. Since video diffusion models can act as world models, a natural adaptation of existing control mechanisms is to use textual instructions describing the actions of all actors as input to the model. However, this approach leads to a catastrophic failure in action-subject association: the model fails to apply an action to the correct target subject. To diagnose this, we analyze a simplified environment consisting of two primitive shapes moving on a solid background in Fig. 1. Surprisingly, even state-of-the-art video generators like Veo 3 [63] struggle to execute a sequence of basic commands such as "triangle moves down while square moves left". Failures are even more significant when several actions are described in sequences, as we show in the figure. This failure relates to a fundamental limitation of attribute binding in existing diffusion models. Indeed, prior work shows that when given multiple conditioning signals, diffusion models often ignore some of them or merge multiple signals together [42, 43, 76], preventing them from achieving per-subject control required by multi-agent simulation.

As a stepping stone for solving the action-binding issue, we propose ActionParty, a multi-subject world model focusing on multi-player game environments, where each player is a controllable subject in a video generation fashion. Inspired by recent works that provide explicit spatio-temporal grounding as model input [42, 43, 76], we introduce *subject state* tokens, a set of latent variables that serve as a persistent implicit state for each subject in the scene. Ultimately, ActionParty operates analagously to a generative game engine, jointly modeling subject state tokens and video latents. Subject state tokens are integrated us-

ing an attention mask that enforces action-subject correspondence. Moreover, to ensure actions are applied to the correct subjects in the video, we leverage 3D Rotary Position Embeddings (RoPE) [65] to bias subject tokens to the current spatial locations of subjects within the video.

We evaluate ActionParty on video outputs from the Melting Pot benchmark [1], a challenging suite of 46 2D multi-agent games. We learn a unified action space that applies to all games, and our model maintains precise per-subject control over up to seven players. Thanks to the separate modeling of subject states, we significantly outperform text-only baselines in action-following accuracy and identity consistency. Beyond pure video generation, we show that the co-generated subject states accurately follow the moving subjects in the video, even through complex interactions and changes in character states. In summary, we make the following contributions:

- We propose ActionParty, a novel autoregressive video generator for generative game simulation. We jointly model video frames and the state of each subject and introduce attention masking to enforce action-subject correspondence to enable consistent multi-subject action control.
- We introduce an action binding mechanism based on RoPE biasing that anchors subject state to specific spatial coordinates within the video, enabling reliable subject localization.
- We demonstrate the first video world model capable of controlling up to seven subjects simultaneously across 46 distinct 2D game environments, achieving superior performance in action-following and subject consistency.

2 Related work

Video diffusion models. Since the rise of diffusion models [30, 64], text-to-video and image-to-video generation have achieved tremendous progress by training on Internet-scale data [6,31,32,41,49,62,79]. While earlier approaches often adopt U-Net [60] as the denoising network [5, 11, 12, 24], recent models have switched to the diffusion transformer (DiT) [55] architecture due to their better scalability in generating high-resolution and complex videos [7,25,52,57,70,81]. Conventional video DiTs typically denoise all video frames jointly. To achieve real-time interactive video generation, later works study distilling a pre-trained bi-directional DiT to a *causal* autoregressive DiT [29,33,86,87]. This often involves Diffusion Forcing [10] that trains the model with an independent noise level per frame, followed by Self-Forcing [36] that fine-tunes the model on self-generated rollout with a distribution matching loss [84,85]. In this work, we follow similar techniques to train an autoregressive action-conditioned video DiT.

Video world models. World models [26, 27] are generative models that are trained to predict the next state (e.g., visual observation) given an action. Actions may be defined as any conditioning signal, such as discrete arrow keys in a video game or text descriptions of changes in a scene. The playable video line of work [50,51] was one of the first in this direction. They try to learn latent actions from raw videos via inverse dynamics. Another line of work simulates first-person

view games like Doom [69], CSGO [2], Minecraft [15, 77], and more [2, 29, 87, 91] where actions are recorded during gameplay. The recent Genie [3, 8] line of research further scales training to real-world videos with both text descriptions and keyboard inputs as actions.

While existing world models excel at single-actor control, there is little work extending them to multiplayer settings, where separate actions must be applied to multiple subjects within a scene. Multiverse [17] and Solaris [61] generate aligned views for each subject in a 3D world. They model each actor's view individually, making action binding trivial. However, separate actor modeling scales the number of video tokens linearly with the number of actors, which may explain why they only test on two-player settings in a single game. MultiGen [56] is concurrent work that also predicts explicit game state (including a minimap and player states) for aligning multiplayer views. Existing work apply one action to individual player video streams. In contrast, our method generates one video containing all actors, necessitating action binding. We further scale our model to up to seven actors and train one model that generalizes across 46 games [1] with a unified action space.

Some model-based multi-agent reinforcement learning methods also build a multi-actor world model for policy learning [72,73,80]. However, these approaches often train small-scale models from scratch [28]. In contrast, our approach is applicable to large-scale pre-trained video DiTs.

Motion control in video diffusion models. Several works have studied object trajectory control [21, 53, 71, 74, 83, 92] in generated videos. They often provide auxiliary control signals such as object masks and bounding boxes along the object trajectory. Another line of work transfers the motion of a reference video to a newly synthesized one [22, 37, 54, 58, 78, 82, 93]. Human motion generation [38,67,89] focuses on generating realistic motion sequences or poses from textual descriptions. Some works model sequences of multiple actions within a single motion trajectory [14, 39]. However, these approaches mainly generate motion for a single subject at a time and operate in structured pose spaces rather than controlling multiple agents within a visual scene. Game settings involve more abstract actions than pure motion, *e.g.*, interacting with items, firing a beam. These actions depend on the current state of the subject, such as its current orientation. Therefore, existing motion control techniques are not suitable for our game environment settings.

Attribute binding in generative models. Attribute binding [23, 59] refers to the challenge of associating specific attributes like text descriptions with corresponding entities within the generated scene. Prior work has studied compositional generation [9, 35, 46, 75, 88] to bind appearance descriptions or spatial relations to different subjects without mixing properties, often implemented with some spatial masking mechanism. Scene graph guided generation [19, 20, 48] condition on a structured map of entities, attributes, and spatial relations. Spatio-temporal binding of attributes in video is an even more complex problem that is unsolved [43, 44, 76], especially when multiple entities are present in the video.

3 Method

We propose ActionParty for multi-subject action-conditioned video generation (Sec. 3.1). To associate subjects with corresponding actions, we run video diffusion models to jointly denoise video frames and latent tokens capturing the state of each subject (Sec. 3.2). We design an update-and-render paradigm with attention masking (Sec. 3.3), and train our model to achieve autoregressive video generation (Sec. 3.4). The overall architecture of ActionParty is shown in Fig. 2.

3.1 Problem Setup

In this paper, we study controllable video generation for gaming and world-modeling scenarios where multiple entities, referred to as subjects, act within a shared observation space (the same video in this paper). The primary challenge, largely unaddressed in prior single-subject world models, is *action binding*. When multiple controllable subjects are present in a video, the model must correctly associate each control signal (typically an action) with its corresponding subject.

Formally, we define a video as a sequence of frames $x_{0:T}$. We consider a set of different environments $\mathcal{G} = \{g_1, g_2, \ldots, g_M\}$. Within a specific environment g, we observe N_g controllable subjects $\{S_1, \ldots, S_{N_g}\}$. Our objective is to generate future video frames conditioned on a global text description of the game, several initial context frames, and a sequence of action inputs for each subject.

For simplicity, in this paper we only consider the control spaces with discrete actions. Specifically, at every timestep $t \in \{0, \ldots, T-1\}$, we apply one discrete action per subject, $a_t^i \in \mathcal{A}$, where the action space is fixed across all environments: $\mathcal{A} = \{\mathcal{A}_0, \ldots, \mathcal{A}_{K-1}\}$. Note that the actions here are treated as abstract "buttons": the same $\mathcal{A}_k$ can have different effects depending on the environment g and the subject's current state. For example, an "interact" command might cause a subject to pick up an object in one game or open a door in another, requiring the model to learn context-dependent dynamics.

Let $a_t \coloneqq \{a_t^i\}_{i=1}^{N_g}$ denote the set of actions applied to each subject at time t. A standard world model f_θ generates the next frame by sampling from a conditional distribution:

$$x_{t+1} \sim f_\theta(x_{t+1} \mid x_{0:t},\, a_{0:t})\,. \tag{1}$$

By rolling this process forward for $t = \{t+1, \ldots, T-1\}$, the model produces a controlled trajectory. In practice, f_θ is implemented as a conditional video diffusion model. We assume it can utilize not only the current action a_t, but also all previous actions $a_{0:t-1}$ as context. Therefore, the key question here is: how can the model associate external action signals with subjects that are only implicitly defined within the pixel space? Beyond their visual appearance, there is no explicit representation of the subjects to which actions can be anchored. To address this, we propose to augment the world model with additional latent variables representing the subject state.

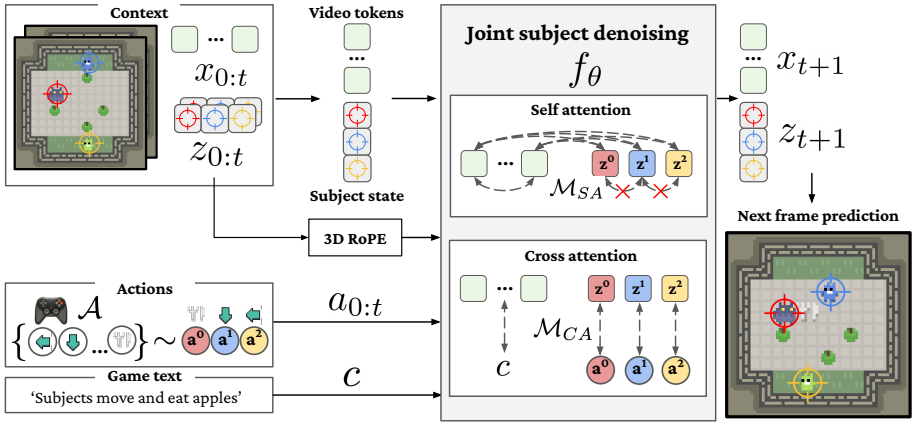


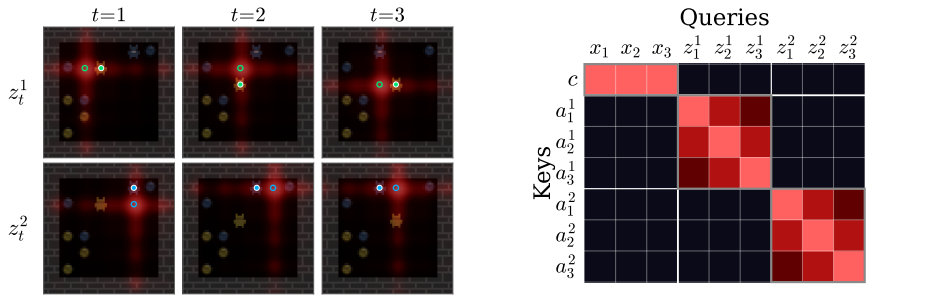
Fig. 2: ActionParty pipeline. Given initial video frames $x_{0:t}$ and subject states $z_{0:t}$ as context, we aim to generate the next video frame x_{t+1} conditioned on action inputs $a_{0:t}$ and a text description of the game c. We concatenate the video and subject state tokens along the sequence dimension and feed them into a diffusion transformer (DiT) for joint denoising. Each DiT block first runs self-attention with an attention mask $\mathcal{M}_{SA}$ and 3D RoPE biasing to render subject states to pixels in the video. It then runs cross-attention with another attention mask $\mathcal{M}_{CA}$ with explicit subject-action binding to update subject states using action inputs.

3.2 Subject State

The main motivation behind our approach is to enable the model f_θ to disambiguate between different subjects via explicit grounding. To this end, we introduce an explicit variable z_t^i representing the state of subject S_i at time t. To simplify notation, we let $z_t \coloneqq \{z_t^i\}_{i=1}^{N_g}$ denote the set of states for all subjects at a given timestep t, and let $z \coloneqq \{z_t\}_{t=0}^{T}$ represent the state trajectory over the entire sequence. Consequently, the world model objective in Eq. (1) can be reformulated as a joint prediction task over x and z:

$$x_{t+1}, z_{t+1} \sim f_\theta(x_{t+1}, z_{t+1} \mid x_{0:t}, a_{0:t}, z_{0:t})\,. \tag{2}$$

To implement this, we extend the standard video DiT architecture (e.g., Wan [70]) to jointly denoise x and z, similar to MMDiT [18]. Specifically, we concatenate the flattened video tokens x and subject state tokens z along the sequence dimension, and run DiT on the combined sequence of tokens. To train such a model, we require a concrete definition for the state z. Ideally, this representation should be minimal yet sufficient to disambiguate subjects during complex interactions. For most game environments, we find that the spatial position of a subject suffices, since two subjects cannot occupy the same position. Therefore, throughout this paper, we define a subject state as its 2D coordinates $z_t^i = (h_t^i, w_t^i)$.



(a) Self-attention subject-video RoPE bias **(b)** Cross-attention bias between a_t^i and z_t^i

Fig. 3: Attention mechanisms in ActionParty DiT. (a) In self-attention, we use RoPE to link a subject in a video frame to its state token z_i^t. We encode the state token with the subject's coordinates in the *previous* timestep, biasing it to attend to video tokens close to the subject. (b) In cross-attention, subject i's state tokens z^i is only allowed to attend to its own actions a^i, ensuring correct subject-action binding. We also allow the text embedding of the environment description c to attend to video tokens x.

3.3 Generative Game Engines

To build an architecture that effectively leverages z, we draw an analogy to standard game engines, which decompose the generation of a single frame into two distinct stages: *state update* and *rendering*. During state update, game engines use actions corresponding to a specific subject to update its state; during rendering, they use the updated subject states to synthesize the video frame. We implement this paradigm within our video diffusion framework by controlling information flow through masked attention as shown in Fig. 3.

State update in cross-attention. We modify the DiT cross-attention layers to process action inputs and update subject states as shown in Fig. 3b. Each discrete action a_t^i is first projected to embeddings that match the latent dimension of the model. We then concatenate the text embedding of the environment description c and all action embeddings to run cross-attention with video and subject state tokens. We apply a mask $\mathcal{M}_{CA}$ that enforces strict subject-action binding: subject i's state tokens z^i are only allowed to attend to its corresponding action embeddings a^i, blocking all other communication between unmatched actions and subject tokens. To preserve the capacity of the pre-trained model, we restrict the text embedding c to attending solely to the video tokens x.

Rendering in self-attention. To render video frames, we apply a mask $\mathcal{M}_{SA}$ in DiT self-attention layers that encourages subject isolation while enabling visual integration. This mask allows all subject tokens to attend to the video tokens, enabling them to read information from the environment. Yet, we block subject-to-subject communication to avoid mixing the state between different subjects. On the other hand, the video tokens attend to all subject tokens, allowing the model to use the state of all subjects for frame synthesis.

Subject-binding with RoPE bias. While the self-attention mask $\mathcal{M}_{SA}$ prevents subject mixing, it does not explicitly link a subject in the video to its state tokens. Thanks to the use of spatial coordinates as subject states, we can leverage the 3D Rotary Position Embeddings (RoPE) to disambiguate this mapping. We design a RoPE biasing mechanism within the self-attention layers, where subject tokens are biased to attend to video tokens at the same spatial coordinates.

Specifically, a video token at position $p = (h, w)$ in frame x_t receives a rotation $\mathcal{R}(t, h, w)$. Given the subject positions from the previous timestep $z^i_{t-1} = (h^i_{t-1}, w^i_{t-1})$, we apply a rotation $\mathcal{R}(t, h^i_{t-1}, w^i_{t-1})$ to the subject token z^i_t. We utilize the previous spatial position because the current position z^i_t is being denoised and thus unknown at the current stage. Still, z^i_{t-1} should be close to the true z^i_t, ensuring that video tokens close to the subject are biased to attend to the corresponding subject state token. Consequently, the model's task is greatly simplified from global disambiguation to local refinement within a single action's radius, as illustrated in Fig. 3a.

3.4 Training and Inference

We train the model on sequences of video tokens and subject state tokens, $(x_{0:t+1}, z_{0:t+1})$, where the context $(x_{0:t}, z_{0:t})$ consists of fully clean ground-truth data, while the targets x_{t+1} and z_{t+1} are noisy. In other words, we train the model f_θ to denoise x_{t+1} and z_{t+1} conditioned on the clean context $(x_{0:t}, z_{0:t})$. To support variable-length contexts, we use sequences up to length T and pad the positions from $t + 2$ to T with fully noisy frames.

During inference, we assume that the initial frame x_0 and the initial subject positions $\{z^i_0\}_{i=1}^{N_g}$ are provided. Note that without this initialization, distinguishing between subjects would be impossible in many environments where entities share an identical appearance. We then perform an autoregressive rollout: given the previously generated context $(x_0||\hat{x}_{1:t}, z^i_0||\hat{z}^i_{1:t})$, where $\hat{x}$ and $\hat{z}$ are previous model predictions, we predict the subsequent video frame x_{t+1} and state variables z_{t+1}. When $t > T$, we drop the oldest frames from the context, ensuring the context window size never exceeds $T - 1$.

4 Experiments

4.1 Setup

Datasets and action space. We train ActionParty on gameplay videos of 46 different 2D games from the Melting Pot benchmark [1]. These games feature character sprites moving on a tile grid with different environments and static objects. Some characters may look identical, making action binding more challenging. ActionParty is trained with an action set $\mathcal{A}$ including 25 different actions. Notably, some actions are only present in specific games. There are 7 base actions shared by all games, which we categorize in 4 different groups: *Idle action* (staying still), *Move actions*: {forward, backward, strafe left, strafe

right}, *Turn actions*: {turn left, turn right} and *Interact*. The "Interact" action involves game-specific interaction with the environment, such as highlighting tiles in front of it to fire a beam. Importantly, subject movements are relative to the subject's orientation. For example, the "forward" action moves a sprite one tile forward in the direction it is facing. This is particularly interesting in our setup as subject orientations can only be inferred from raw video frames; hence, the model has to comprehend visual information to get the correct action outcome, rather than learning absolute directional instructions like "move up". We generate game-specific text descriptions using an LLM from the Melting Pot paper and videos of each game (full details in Supplementary). For training, we gather 2,000 videos per game at a resolution of 512×512. We generate diverse rollouts for each game by executing either random actions or pretrained policies at each step t. For evaluation, we collect 230 rollouts (5 per game). As completely random actions result in players staying close to their initial positions, we enforced $a_t^i \sim \{Move, Turn\}$ for $t < 2$ to leave its initial state, and at least one subject is assigned an "Interact" action for $t \geq 2$.

Implementation details. We fine-tune Wan2.1-1.3B [70], an open-source text-to-video (T2V) DiT with the training setup outlined in Sec. 3.4. To speed up model convergence, we first pretrain the model on raw game videos to adapt to the autoregressive setup. For this stage, we only provide text conditioning, which is a simple description of the game, without action control, subject state, or any architecture modification. We train on all games ($|\mathcal{G}| = 46$) for 22.5k steps with a batch size of 64. Then, we add action control and fine-tune the full ActionParty model for 65k steps with the same batch size. We train our method with $T = 5$ timesteps, corresponding to 4 unique actions per rollout. We use Adam [40] to optimize the flow matching loss [45, 47] computed on x_{t+1}^i and z_{t+1}^i. We consider a maximum of 7 subjects in a scene and use zeros for padded subject states. At inference time, we sample for 20 steps with a timestep shift of 5.0 [18]. We implement attention masking with FlexAttention [16], only adding $N \times T$ subject tokens per subject, which brings minimal overhead compared to the number of video tokens. In practice, we only add a 6% overhead for 7 players for $T = 5$ (40 extra tokens). Note that this is significantly more efficient than other methods [17,61] that generate multiple views of the scene for each subject.

Baselines. Since there is no open-source method capable of controlling more than two subjects, we design several baselines. To demonstrate the challenge of action binding from text prompts, we propose a naive *Zero-shot I2V* baseline where we simply prompt a video model with a textual description of the initial subject positions and the actions to execute. We use the Wan2.1-14B I2V model [70], which is much larger than the 1.3B model ActionParty fine-tunes from. We also study such *Text-Action* control on our autoregressive setup, using the same backbone as ActionParty, but trained only using textual conditioning. Inspired by view-based action control methods [56, 61], we add a baseline that generates the global view together with subject-centered 3×3 grid tile subject views to which actions are applied. Finally, we take the autoregressive model

Table 1: Baseline comparison on action binding metrics and visual fidelity. ActionParty is the only method able to associate actions to the correct characters (MA), preserve the subject appearance (SP), and execute rollouts with correct subject trajectories (DR). We also compute fidelity metrics, LPIPS, PSNR, and FVD, which show that videos generated by ActionParty are consistent with ground-truth.

Method	Action Binding			Visual Quality		
	MA ↑	**SP** ↑	**DR** ↑	**LPIPS** ↓	**PSNR** ↑	**FVD** ↓
Pretrained AR	0.065	0.571	0.330	0.0494	26.98	59.30
Zero-shot I2V	0.027	0.422	0.440	0.0906	23.17	281.55
View-based	0.013	0.237	0.588	0.0503	27.38	75.1
Text-Action	0.158	0.668	0.433	0.0353	29.14	56.74
Ours (ActionParty)	**0.779**	**0.903**	**0.886**	**0.0102**	**36.35**	**17.16**

pretrained on game videos without action control (termed *Pretrained AR*) as a random action lower bound.

Evaluation. The effect of an action on a subject depends on its state and the environment, e.g., position/orientation and occlusion. To evaluate action binding, direct visual comparison with ground-truth videos is unreliable as the correct action may be performed even if the subject's state has drifted. We propose a Movement Accuracy metric (MA) that extracts each subject's movement (left, right, up, down, or still) from consecutive video frames and compares it with the input action. To do so, we train simple subject detectors that achieve near 100% accuracy on raw video frames. To measure the effect of context-dependent "Interact" actions, we introduce an Effect Accuracy metric (EA) that extracts a 3×3 tile patch centered on the ground truth player position and evaluate dissimilarity with the previous frame with SSIM. We empirically set a threshold patch SSIM ≥ 0.85 to detect executed actions. We also introduce two simple metrics to monitor subjects: Subject Preservation (SP) and Detection Rate (DR). SP measures how many subjects are retained at the end of the generation. DR measures the percentage of steps where the subject's position aligns with ground-truth. Please refer to Supplementary for more details on metric implementations. For completeness, we also report standard visual quality metrics between ground-truth and generated videos, including PSNR, LPIPS [90], and FVD [68]. Finally, as our model also predicts subject coordinates, we compute z_t error as the L2 distance between ground-truth and predicted positions.

4.2 Action binding evaluation

Quantitative evaluation. We compare action binding performance for ActionParty and baselines in Tab. 1. First, we notice that baselines dramatically fail in movement accuracy, for which the best baseline (Text-Action) achieves 0.158. ActionParty instead yields **0.779**, showcasing that our contributions are fundamental to correctly bind actions to specific subjects. Interestingly, we noticed

Table 2: We evaluate the Effect Accuracy of our generated videos across different action types. ActionParty achieves the highest accuracy on "Interact" actions compared to other methods, indicating that it most often creates the desired visual effect on neighboring tiles across different games.

Method	Effect Accuracy ↑				
	Idle	Move	Turn	Interact	Overall
Pretrained AR	0.399	0.414	0.462	0.346	0.412
Zero-shot I2V	0.000	0.101	0.150	0.000	0.126
Text-Action	0.357	0.420	0.553	0.326	0.433
Ours (ActionParty)	**0.899**	**0.867**	**0.914**	**0.774**	**0.861**

that baselines tend to remove the subject from the scene, as highlighted from our improved SP (**0.903**) with respect to the closest baseline (0.668). Finally, we report improvements in DR, showing that subjects are rendered in the correct position most of the time (**0.886**). The trained methods demonstrate consistency with ground-truth, with ours achieving the highest alignment due to improved action following as shown by the visual fidelity metrics. The view-based baseline has difficulty binding actions to separate subject views without explicit position binding. We also break down the effect accuracy in each category in Tab. 2. ActionParty works the best across all types of actions. Especially on the context-dependent "Interact" action, our model more than doubles the success rate of existing baselines.

Autoregressive stability. In Fig. 5, we measure the stability of our autoregressive inference. We plot, for different autoregressive steps, the movement accuracy of ActionParty and baselines. While some baselines, such as Text-Action, yield suboptimal yet acceptable results on the first step, it quickly degrades in subsequent steps. Conversely, thanks to the joint modeling of subject states, ActionParty allows for a relatively stable autoregressive inference, showing that our action binding performance remains consistent over time.

Qualitative comparison. In Fig. 4, we compare ActionParty with baselines on a sample from the Paintball game (more results in Supplementary). In this game, subjects can move around, turn, and shoot a colored beam with the "Interact" action. In the figure, we use small arrows to denote the ground-truth position and orientation of all subjects at a given step. Compared to baselines, ActionParty is the only method that successfully binds all actions to subjects. Our rendered frames are similar to the ground-truth, with arrows overlaid on subjects, and we successfully bind the "Interact" action in the last step, as a red beam appears in front of the subject. Conversely, the Text-Action baseline fails whenever subjects move away from their initial positions, as it is unable to differentiate distinct subjects. The zero-shot I2V approach fails drastically at assigning separate actions to different subjects and applies a continuous downwards movement to detected sprites. It also fails to maintain the appearance of

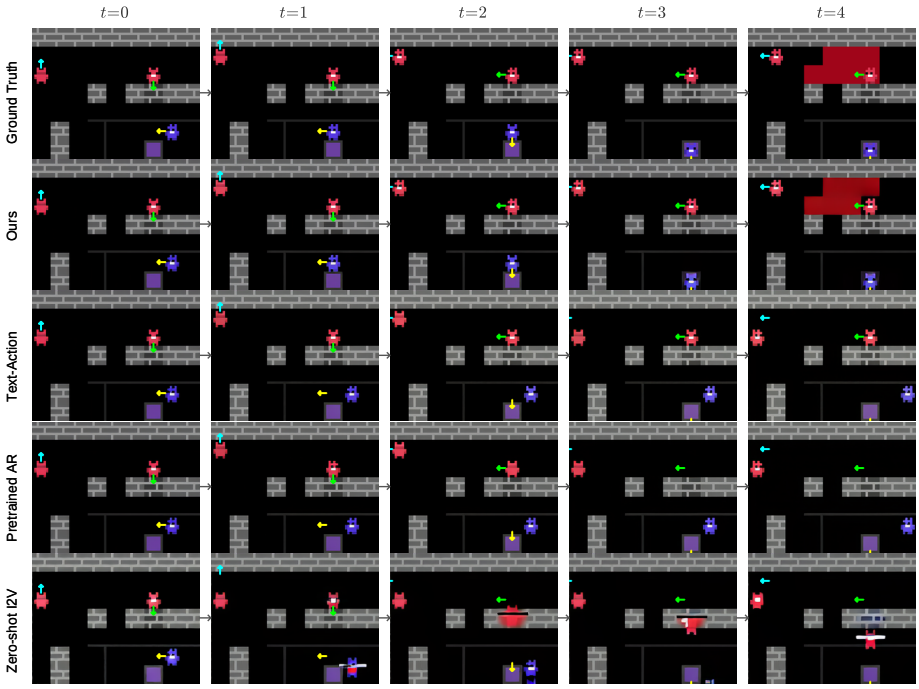


Fig. 4: Qualitative comparison with baselines. We display ground truth subject positions and orientations at each step with an arrow for each subject, which reflects the ground truth subjects (notice that the arrow is consistent in all methods). Our method is the only one able to follow the ground truth actions with appropriate action binding.

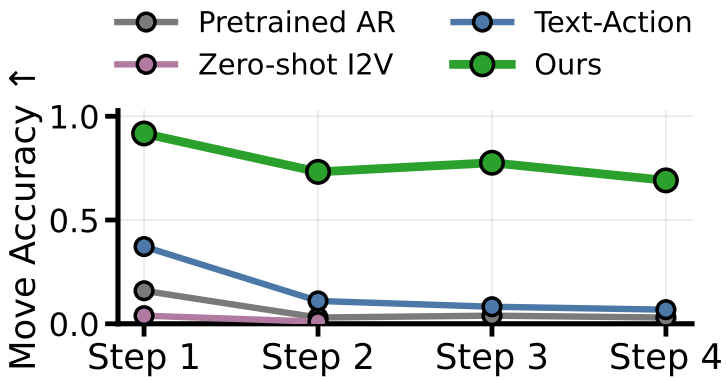


Fig. 5: Movement accuracy (MA) over autoregressive steps. ActionParty maintains stable action binding across multiple rollout steps, whereas baselines degrade over time and get close to 0.

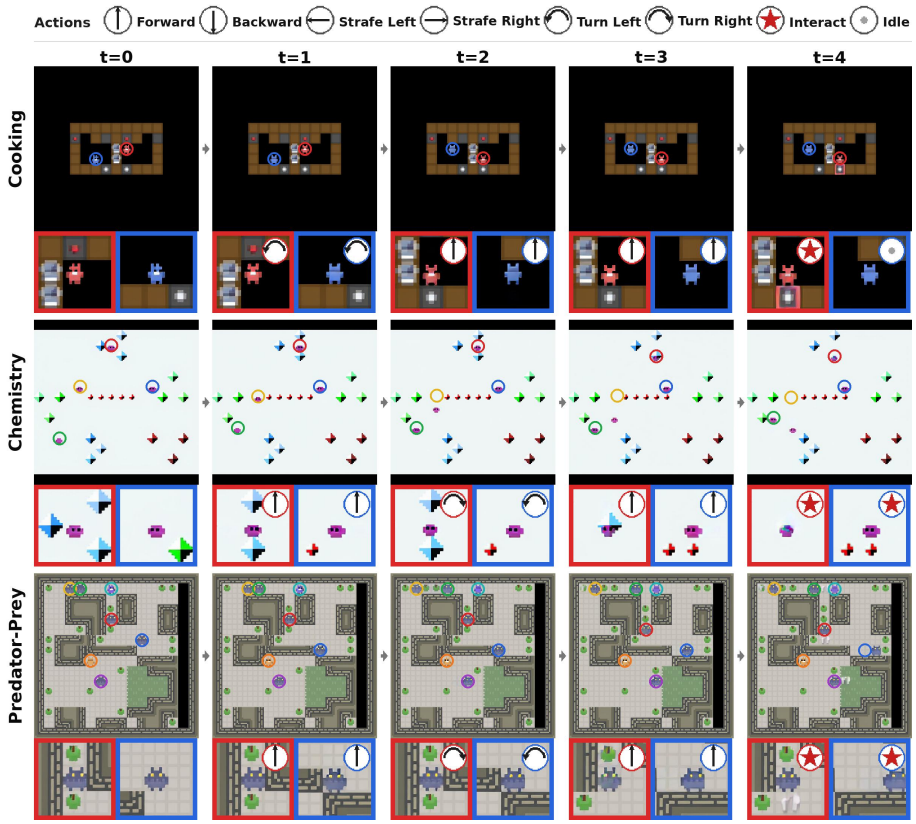


Fig. 6: Results of ActionParty with x_t and z_t (circle) predictions for 2-, 4-, and 7-player games with a diversity of subjects. The full frames are generated and we show a centred crop around 2 subjects for visualization purposes with action annotations on the *resulting* frame. Note that actions are *relative to orientation*, so the subject moving forward is in the direction of where it is facing.

subjects. This aligns with our reported metrics, and showcases the importance of our problem setup as a common failure mode of video diffusion models.

Qualitative results. In Fig. 6, we test our method on a range of games including Cooking, Chemistry, and Predator-Prey, which have 2, 4, and 7 subjects being controlled, respectively. We plot a zoomed-in view of two subjects in each frame to aid visibility, however all subjects are indeed being controlled in a single scene. Coordinate predictions z_t are plotted as colored circles. Our position control can successfully control identical-looking subjects as in the Chemistry game. Our single learned action space can also apply the same actions to diverse appearances in unique backgrounds as seen in the bottom two rows. Even if the coordinate prediction is slightly off, action binding may still be successful if it is enough to spatially disambiguate different subjects.

Table 3: Ablations on the 4-action Coins game. The full ActionParty model achieves the most accurate localization of players indicated by the highest movement accuracy (MA) and the lowest z_t error. Our $\mathcal{M}_{\text{CA}}$ is important for consistent subject preservation according to the subject appearance (SP) and detection rate (DR).

Method	**MA** ↑	z_t **Error** ↓	**SP** ↑	**DR** ↑
w/o $\mathcal{M}_{SA}$	0.580	0.108	**1.00**	0.716
w/o $\mathcal{M}_{CA}$	0.052	0.215	0.88	0.307
Frame-wise $\mathcal{M}_{CA}$	0.052	0.240	0.80	0.293
No RoPE in SA	0.032	0.291	**1.00**	0.278
Ours (ActionParty)	**0.872**	**0.072**	**1.00**	**0.913**

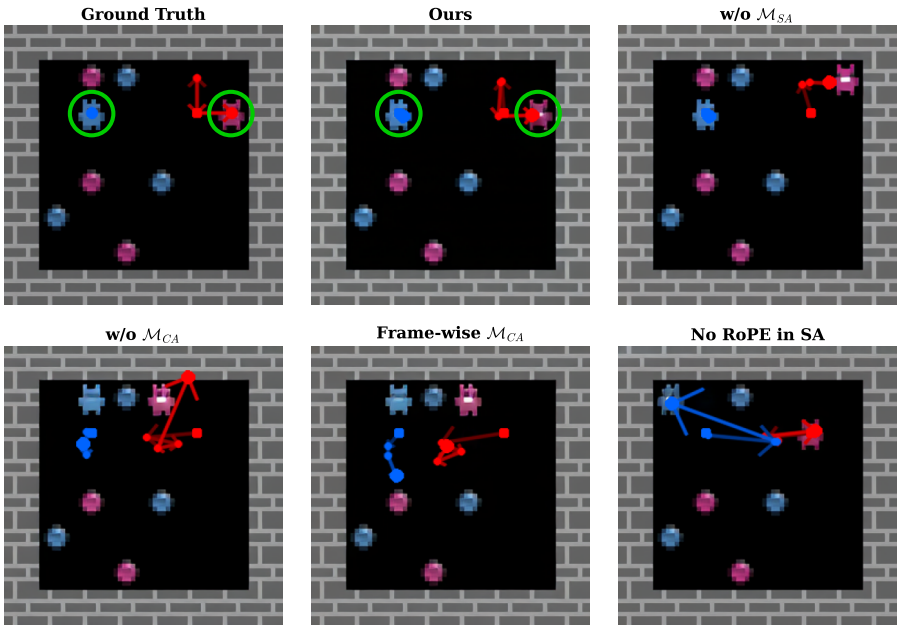


Fig. 7: Qualitative comparison of ablations run on the same initial frame and action trajectory. The blue subject is still, while the purple one moves and returns to its previous position. The z_t trajectory is plotted for each subject. If we remove our introduced components, we lose the possibility to control characters.

4.3 Ablation studies

We conduct ablation studies in a smaller-scale setup due to excessive training costs. Here, we collect 2.5k samples from one of the game environments called Coins at a resolution of 256×256. This game involves two players walking around and picking up coins that are randomly scattered around a map. All ablations are initialized from the same autoregressive pretrained checkpoint and

fine-tuned with action control for 45k steps. In this setup, our model achieves **87.2%** movement accuracy (MA) and maintains a subject detection rate (DR) at ground truth positions of **91.3%**.

We list results in Tab. 3 and qualitatively compare the final generated frame and subject trajectories z_t in Fig. 7. We first study the importance of the attention masks in ActionParty. By removing the self-attention mask $\mathcal{M}_{\text{SA}}$, subject tokens are allowed to attend to each other, causing information leak between subjects and ultimately reducing MA to 58%. We also tried removing the cross-attention mask $\mathcal{M}_{\text{CA}}$, which results in a drop of MA to 5.2%. Evidently, the presence of $\mathcal{M}_{\text{CA}}$ ensures the assignment of actions to the correct subject state tokens, and thus is a fundamental component for correct action binding. Similarly, adding a frame-wise $\mathcal{M}_{\text{CA}}$ that only allows each subject token to attend to the action at its own frame, rather than all frames, reduces performance drastically. Attending to the full action sequence in z_t may improve action following. Finally, removing the RoPE bias in self-attention prevents any action binding as the subject tokens do not attend to the correct video tokens (MA drops to 3.2%). In general, we notice that the z_t error and detection rate closely align with MA, meaning that action binding relies on accurate localization of subjects.

5 Conclusion

In this paper, we present ActionParty, a framework for resolving the action binding challenge in multi-player 2D game simulations. By introducing subject state tokens that are jointly denoised with video latents, we successfully associate unique, controllable identifiers with individual subjects across 46 diverse environments. Through attention masking and spatial RoPE biasing, ActionParty disentangles subject update and frame rendering, enabling precise control of up to seven subjects simultaneously. Our results demonstrate that explicit subject grounding significantly improves action-following accuracy and identity preservation in the autoregressive video diffusion setup. We believe our study addresses a fundamental bottleneck in action-conditioned video generation, and we hope it can inspire future research on robust, multi-subject video world models that extend to 3D, partially observed settings.

Acknowledgements

Alexander Pondaven is generously supported by a Snap studentship and the EPSRC Centre for Doctoral Training in Autonomous Intelligent Machines and Systems [EP/S024050/1].

References

1. Agapiou, J.P., et al.: Melting Pot 2.0. arXiv preprint arXiv:2211.13746 (2022)
2. Alonso, E., Jelley, A., Micheli, V., Kanervisto, A., Storkey, A.J., Pearce, T., Fleuret, F.: Diffusion for world modeling: Visual details matter in atari. NeurIPS (2024)

3. Ball, P.J., et al.: Genie 3: A new frontier for world models (2025), https://deepmind.google/blog/genie-3-a-new-frontier-for-world-models/, accessed 30 Jun 2026
4. Bettini, M., Kortvelesy, R., Blumenkamp, J., Prorok, A.: VMAS: A vectorized multi-agent simulator for collective robot learning. In: International Symposium on Distributed Autonomous Robotic Systems (2022)
5. Blattmann, A., Rombach, R., Ling, H., Dockhorn, T., Kim, S.W., Fidler, S., Kreis, K.: Align your Latents: High-resolution video synthesis with latent diffusion models. In: CVPR (2023)
6. Blattmann, A., et al.: Stable video diffusion: Scaling latent video diffusion models to large datasets. arXiv preprint arXiv:2311.15127 (2023)
7. Brooks, T., et al.: Video generation models as world simulators. OpenAI technical reports (2024), https://openai.com/research/video-generation-models-as-world-simulators, accessed 30 Jun 2026
8. Bruce, J., et al.: Genie: Generative interactive environments. In: ICML (2024)
9. Chefer, H., Alaluf, Y., Vinker, Y., Wolf, L., Cohen-Or, D.: Attend-and-Excite: Attention-based semantic guidance for text-to-image diffusion models. TOG (2023)
10. Chen, B., Martí Monsó, D., Du, Y., Simchowitz, M., Tedrake, R., Sitzmann, V.: Diffusion Forcing: Next-token prediction meets full-sequence diffusion. NeurIPS (2024)
11. Chen, H., Xia, M., He, Y., Zhang, Y., Cun, X., Yang, S., Xing, J., Liu, Y., Chen, Q., Wang, X., Weng, C., Shan, Y.: VideoCrafter1: Open diffusion models for high-quality video generation. arXiv preprint arXiv:2310.19512 (2023)
12. Chen, H., Zhang, Y., Cun, X., Xia, M., Wang, X., Weng, C., Shan, Y.: VideoCrafter2: Overcoming data limitations for high-quality video diffusion models. In: CVPR (2024)
13. Chen, T., Chen, Z., Chen, B., Cai, Z., Liu, Y., Li, Z., Liang, Q., Lin, X., Ge, Y., Gu, Z., et al.: RoboTwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. arXiv preprint arXiv:2506.18088 (2025)
14. Chi, S., Chi, H.g., Ma, H., Agarwal, N., Siddiqui, F., Ramani, K., Lee, K.: M2D2M: Multi-motion generation from text with discrete diffusion models. In: ECCV (2024)
15. Decart, E., McIntyre, Q., Campbell, S., Chen, X., Wachen, R.: Oasis: A universe in a transformer (2024), https://oasis-model.github.io/, accessed 30 Jun 2026
16. Dong, J., Feng, B., Guessous, D., Liang, Y., He, H.: Flex Attention: A programming model for generating optimized attention kernels. arXiv preprint arXiv:2412.05496 (2024)
17. Enigma team: Introducing multiverse: The first ai multiplayer world model (2025), https://enigma.inc/blog, accessed 30 Jun 2026
18. Esser, P., et al.: Scaling rectified flow transformers for high-resolution image synthesis. In: ICML (2024)
19. Farshad, A., Yeganeh, Y., Chi, Y., Shen, C., Ommer, B., Navab, N.: Scenegenie: Scene graph guided diffusion models for image synthesis. In: ICCV Workshops (2023)
20. Gao, G., Liu, W., Chen, A., Geiger, A., Schölkopf, B.: Graphdreamer: Compositional 3d scene synthesis from scene graphs. In: CVPR (2024)
21. Geng, D., et al.: Motion Prompting: Controlling video generation with motion trajectories. In: CVPR (2025)
22. Gokmen, A.B., Ekin, Y., Bilecen, B.B., Dundar, A.: RoPECraft: Training-free motion transfer with trajectory-guided rope optimization on diffusion transformers. NeurIPS (2025)

23. Greff, K., Van Steenkiste, S., Schmidhuber, J.: On the binding problem in artificial neural networks. arXiv preprint arXiv:2012.05208 (2020)
24. Guo, Y., Yang, C., Rao, A., Wang, Y., Qiao, Y., Lin, D., Dai, B.: Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. In: ICLR (2024)
25. Gupta, A., Yu, L., Sohn, K., Gu, X., Hahn, M., Fei-Fei, L., Essa, I., Jiang, L., Lezama, J.: Photorealistic video generation with diffusion models. arXiv preprint arXiv:2312.06662 (2023)
26. Ha, D., Schmidhuber, J.: Recurrent world models facilitate policy evolution. NeurIPS (2018)
27. Ha, D., Schmidhuber, J.: World Models. arXiv preprint arXiv:1803.10122 (2018)
28. Hafner, D., Lillicrap, T., Ba, J., Norouzi, M.: Dream to control: Learning behaviors by latent imagination. In: ICLR (2020)
29. He, X., et al.: Matrix-Game 2.0: An open-source real-time and streaming interactive world model. arXiv preprint arXiv:2508.13009 (2025)
30. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: NeurIPS (2020)
31. Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., Fleet, D.J.: Video diffusion models. NeurIPS (2022)
32. Ho, J., et al.: Imagen Video: High definition video generation with diffusion models. arXiv preprint arXiv:2210.02303 (2022)
33. Hong, Y., et al.: RELIC: Interactive video world model with long-horizon memory. arXiv preprint arXiv:2512.04040 (2025)
34. Hu, A., Russell, L., Yeo, H., Murez, Z., Fedoseev, G., Kendall, A., Shotton, J., Corrado, G.: GAIA-1: A generative world model for autonomous driving. arXiv preprint arXiv:2309.17080 (2023)
35. Huang, K., Duan, C., Sun, K., Xie, E., Li, Z., Liu, X.: T2I-CompBench++: An enhanced and comprehensive benchmark for compositional text-to-image generation. PAMI (2025)
36. Huang, X., Li, Z., He, G., Zhou, M., Shechtman, E.: Self Forcing: Bridging the train-test gap in autoregressive video diffusion. NeurIPS (2025)
37. Jeong, H., Park, G.Y., Ye, J.C.: VMC: Video motion customization using temporal attention adaption for text-to-video diffusion models. In: CVPR (2024)
38. Jiang, B., Chen, X., Liu, W., Yu, J., Yu, G., Chen, T.: MotionGPT: Human motion as a foreign language. NeurIPS (2023)
39. Jin, P., Li, H., Cheng, Z., Li, K., Yu, R., Liu, C., Ji, X., Yuan, L., Chen, j.: Local action-guided motion diffusion model for text-to-motion generation. In: ECCV (2024)
40. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
41. Kong, W., et al.: HunyuanVideo: A systematic framework for large video generative models. arXiv preprint arXiv:2412.03603 (2024)
42. Li, Y., Liu, H., Wu, Q., Mu, F., Yang, J., Gao, J., Li, C., Lee, Y.J.: GLIGEN: Open-set grounded text-to-image generation. In: CVPR (2023)
43. Lian, L., Shi, B., Yala, A., Darrell, T., Li, B.: Llm-grounded video diffusion models. In: ICLR (2024)
44. Lin, H., Zala, A., Cho, J., Bansal, M.: VideoDirectorGPT: Consistent multi-scene video generation via llm-guided planning. In: COLM (2024)
45. Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. In: ICLR (2023)

46. Liu, N., Li, S., Du, Y., Torralba, A., Tenenbaum, J.B.: Compositional visual generation with composable diffusion models. In: ECCV (2022)
47. Liu, X., et al.: Flow straight and fast: Learning to generate and transfer data with rectified flow. In: ICLR (2023)
48. Liu, Y., Li, X., Zhang, Y., Qi, L., Li, X., Wang, W., Li, C., Li, X., Yang, M.H.: Controllable 3d outdoor scene generation via scene graphs. In: ICCV (2025)
49. Ma, G., et al.: Step-Video-T2V technical report: The practice, challenges, and future of video foundation model. arXiv preprint arXiv:2502.10248 (2025)
50. Menapace, W., Lathuiliere, S., Siarohin, A., Theobalt, C., Tulyakov, S., Golyanik, V., Ricci, E.: Playable Environments: Video manipulation in space and time. In: CVPR (2022)
51. Menapace, W., Lathuiliere, S., Tulyakov, S., Siarohin, A., Ricci, E.: Playable video generation. In: CVPR (2021)
52. Menapace, W., et al.: Snap video: Scaled spatiotemporal transformers for text-to-video synthesis. CVPR (2024)
53. Namekata, K., Bahmani, S., Wu, Z., Kant, Y., Gilitschenski, I., Lindell, D.B.: SG-I2V. In: ICLR (2025)
54. Pardo, A., Pizzati, F., Zhang, T., Pondaven, A., Torr, P., Perez, J.C., Ghanem, B.: Matchdiffusion: Training-free generation of match-cuts. In: ICCV (2025)
55. Peebles, W., Xie, S.: Scalable diffusion models with transformers. ICCV (2023)
56. Po, R., Zhang, D.J., Hertz, A., Wetzstein, G., Wadhwa, N., Ruiz, N.: Multigen: Level-design for editable multiplayer worlds in diffusion game engines. arXiv preprint arXiv:2603.06679 (2026)
57. Polyak, A., et al.: Movie Gen: A cast of media foundation models. arXiv preprint arXiv:2410.13720 (2024)
58. Pondaven, A., Siarohin, A., Tulyakov, S., Torr, P., Pizzati, F.: Video motion transfer with diffusion transformers. In: CVPR (2025)
59. Rassin, R., Hirsch, E., Glickman, D., Ravfogel, S., Goldberg, Y., Chechik, G.: Linguistic Binding in Diffusion Models: Enhancing attribute correspondence through attention map alignment. NeurIPS (2023)
60. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: MICCAI (2015)
61. Savva, G., Michel, O., Lu, D., Waiwitlikhit, S., Meehan, T., Mishra, D., Poddar, S., Lu, J., Xie, S.: Solaris: Building a multiplayer video world model in minecraft. arXiv preprint arXiv:2602.22208 (2026)
62. Seawead, T., Yang, C., et al.: Seaweed-7B: Cost-effective training of video generation foundation model (2025)
63. Sharma, A., et al.: Veo (2024), `https://deepmind.google/technologies/veo/`, accessed 30 Jun 2026
64. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: ICML (2015)
65. Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., Liu, Y.: Roformer: Enhanced transformer with rotary position embedding. Neurocomputing (2024)
66. Suo, S., Regalado, S., Casas, S., Urtasun, R.: TrafficSim: Learning to simulate realistic multi-agent behaviors. In: CVPR (2021)
67. Tevet, G., Raab, S., Gordon, B., Shafir, Y., Cohen-Or, D., Bermano, A.H.: Human motion diffusion model. In: ICLR (2023)
68. Unterthiner, T., van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., Gelly, S.: Towards accurate generative models of video: A new metric & challenges. arXiv preprint arXiv:1812.01717 (2018)

69. Valevski, D., Leviathan, Y., Arar, M., Fruchter, S.: Diffusion models are real-time game engines. In: ICLR (2025)
70. Wang, A., et al.: Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314 (2025)
71. Wang, J., Zhang, Y., Zou, J., Zeng, Y., Wei, G., Yuan, L., Li, H.: Boximator: Generating rich and controllable motions for video synthesis. arXiv preprint arXiv:2402.01566 (2024)
72. Wang, Z., Meger, D.: VDFD: Multi-agent value decomposition framework with disentangled world model. arXiv preprint arXiv:2309.04615 (2023)
73. Wang, Z., Shi, C., Hu, J., Rohling, K., Martín-Martín, R., Zhang, A., Stone, P.: Factored latent action world models. arXiv preprint arXiv:2602.16229 (2026)
74. Wu, W., Li, Z., Gu, Y., Zhao, R., He, Y., Zhang, D.J., Shou, M.Z., Li, Y., Gao, T., Zhang, D.: DragAnything: Motion control for anything using entity representation. In: ECCV (2024)
75. Wu, Z., Rubanova, Y., Kabra, R., Hudson, D.A., Gilitschenski, I., Aytar, Y., Van Steenkiste, S., Allen, K.R., Kipf, T.: Neural Assets: 3d-aware multi-object scene synthesis with image diffusion models. NeurIPS (2024)
76. Wu, Z., Siarohin, A., Menapace, W., Skorokhodov, I., Fang, Y., Chordia, V., Gilitschenski, I., Tulyakov, S.: Mind the Time: Temporally-controlled multi-event video generation. In: CVPR (2025)
77. Xiao, Z., Lan, Y., Zhou, Y., Ouyang, W., Yang, S., Zeng, Y., Pan, X.: WorldMem: Long-term consistent world simulation with memory. NeurIPS (2025)
78. Xiao, Z., Zhou, Y., Yang, S., Pan, X.: Video diffusion models are training-free motion interpreter and controller. NeurIPS (2024)
79. Xing, J., Xia, M., Zhang, Y., Chen, H., Wang, X., Wong, T.T., Shan, Y.: DynamiCrafter: Animating open-domain images with video diffusion priors. In: ECCV (2024)
80. Xue, D., Jiang, J., Zhang, S., Guo, W., Yuan, L., Zhang, Z., Yu, Y.: Learning disentangled multi-agent world model for decentralized control (2025), `https://openreview.net/forum?id=2KTEaY0lLi`, accessed 30 Jun 2026
81. Yang, Z., et al.: CogVideoX: Text-to-video diffusion models with an expert transformer. In: ICLR (2025)
82. Yatim, D., Fridman, R., Bar-Tal, O., Kasten, Y., Dekel, T.: Space-time diffusion features for zero-shot text-driven motion transfer. In: CVPR (2024)
83. Yin, S., Wu, C., Liang, J., Shi, J., Li, H., Ming, G., Duan, N.: DragNUWA: Fine-grained control in video generation by integrating text, image, and trajectory. arXiv preprint arXiv:2308.08089 (2023)
84. Yin, T., Gharbi, M., Park, T., Zhang, R., Shechtman, E., Durand, F., Freeman, B.: Improved distribution matching distillation for fast image synthesis. NeurIPS (2024)
85. Yin, T., Gharbi, M., Zhang, R., Shechtman, E., Durand, F., Freeman, W.T., Park, T.: One-step diffusion with distribution matching distillation. In: CVPR (2024)
86. Yin, T., Zhang, Q., Zhang, R., Freeman, W.T., Durand, F., Shechtman, E., Huang, X.: From slow bidirectional to fast autoregressive video diffusion models. In: CVPR (2025)
87. Yu, J., Qin, Y., Wang, X., Wan, P., Zhang, D., Liu, X.: GameFactory: Creating new games with generative interactive videos. In: ICCV (2025)
88. Zarei, A., Rezaei, K., Basu, S., Saberi, M., Moayeri, M., Kattakinda, P., Feizi, S.: Improving compositional attribute binding in text-to-image generative models via enhanced text embeddings. arXiv preprint arXiv:2406.07844 (2024)

89. Zhang, M., Cai, Z., Pan, L., Hong, F., Guo, X., Yang, L., Liu, Z.: Motiondiffuse: Text-driven human motion generation with diffusion model. PAMI (2024)
90. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018)
91. Zhang, Y., et al.: Matrix-Game: Interactive world foundation model. arXiv preprint arXiv:2506.18701 (2025)
92. Zhang, Z., Liao, J., Li, M., Dai, Z., Qiu, B., Zhu, S., Qin, L., Wang, W.: Tora: Trajectory-oriented diffusion transformer for video generation. In: CVPR (2025)
93. Zhao, R., Gu, Y., Wu, J.Z., Zhang, D.J., Liu, J.W., Wu, W., Keppo, J., Shou, M.Z.: MotionDirector: Motion customization of text-to-video diffusion models. In: ECCV (2024)