LIVEACT: TOWARDS REAL-TIME MULTI-AGENT WORLD MODELS WITH HETEROGENEOUS ACTION INTERFACES

Anonymous authors
Paper under double-blind review

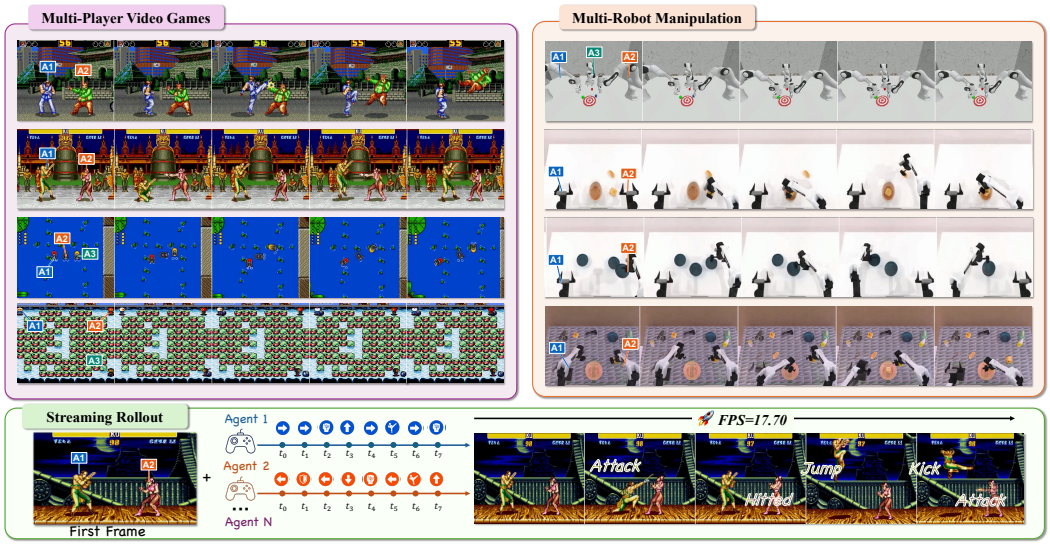

Figure 1: **Agent-specific control across heterogeneous domains.** Given an initial frame and synchronized action streams, LiveAct generates shared visual futures for multi-agent games (top left) and multi-robot manipulation (top right) while preserving stream-to-agent correspondence (bottom).

ABSTRACT

Interactive video world models predict how an environment evolves under user actions, but most existing systems assume one controlled agent and one task-specific action interface. We study multi-agent video world modeling, where one model generates a joint visual future for multiple controlled agents while interpreting heterogeneous controls ranging from binary game buttons to continuous robot commands. This task presents two complementary challenges. First, every agent-indexed action stream must control its intended visual entity independently. Second, the model must adapt the meaning of each control variable across different tasks, where the same tensor dimension may denote an attack, steering command, or robot command. We introduce LiveAct, centered on a Heterogeneous Action Interface (HAI) whose two components mirror this decomposition. To support independent control, Agent-Aware Spatial Modulation uses language-driven spatial gates and agent-specific attention to bind each action stream to its intended visual agent. To accommodate heterogeneous controls, Schema-Guided Action Adaptation describes each control variable in language and modulates its semantic token with the synchronized numerical value. We further transfer the complete interface to a blockwise causal generator through teacher forcing and rollout-aware distribution matching, enabling efficient streaming generation while retaining agent-specific control and action semantics. We instantiate the formulation on six multi-agent sources spanning games and multi-robot manipulation, and evaluate visual quality, action following, agent attribution, and latency. Across games and robot manipulation, LiveAct reaches 0.898 and 0.942 agent-attribution accuracy without agent locations or identity crops, and the causal generator attains 17.70 FPS at 512×448.

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.

1 INTRODUCTION

Video world models (Team et al., 2025; Zhang et al., 2026; Hong et al., 2025; Bi et al., 2026; Li et al., 2025) turn interaction into conditional generation: given visual history and actions, they synthesize the observations that follow. Such models can provide scalable experience for game simulation, robot learning, planning, and interactive content creation (Hafner et al., 2023; Yang et al., 2023). Existing systems increasingly support persistent and low-latency rollout, but most still expose a single aggregate player or camera-control stream (Valevski et al., 2025; Gao et al., 2026). Such a generation paradigm is insufficient for inherently multi-agent environments, where independently controlled entities act concurrently and jointly determine the visual future. We therefore study *multi-agent video world modeling*: given an initial frame, a scene description, and a set of frame-level action streams for multiple agents, the model generates a shared visual future in which each stream controls its intended agent while all agents remain consistent with one another and the environment.

Existing interactive world models cover this setting only in part, and the shortfall lies in the action interface rather than the video backbone. Multi-agent control has relied on numerical action vectors, which keep exact values but typically reach the right entity through spatial priors supplied at inference time and leave control semantics implicit in task-specific dimensions (Wu et al., 2026; Pondaven et al., 2026; Hu et al., 2026). Language-based actions transfer semantics across entities and tasks but give up the synchronized numerical values that continuous control requires (Zhu et al., 2026b). Tab. 1 summarizes the landscape: no current interface is at once multi-agent, real-time, semantically grounded, and numerically exact. We ask whether one interface can satisfy all four requirements without inference-time masks, boxes, poses, or reference images.

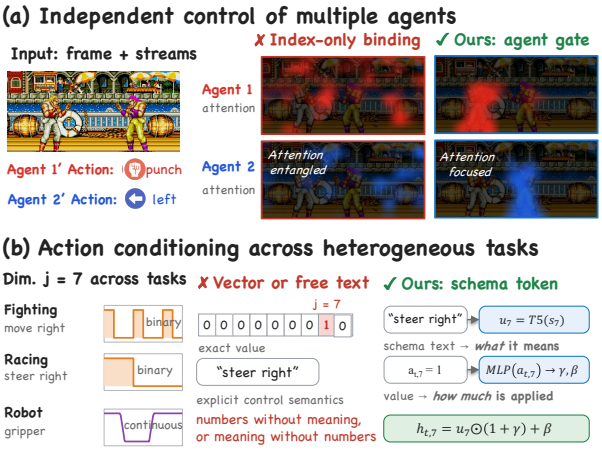

Figure 2: **Two challenges for multi-agent action interfaces.** (a) Streams that differ only by input order entangle fighters under index-only binding; LiveAct's agent gate keeps each on its fighter. (b) Dimension $j{=}7$ changes meaning across tasks. A raw vector keeps the value but not the meaning, free text the reverse; LiveAct modulates each schema token by its value.

However, a unified action interface must still address two challenges: *who* each action stream should control in a shared visual scene, and *what* each control dimension means across heterogeneous tasks. These requirements lead to two key challenges.

(1) ***maintaining independent control over multiple interacting agents.*** A model must keep each agent responsive to its own action stream while capturing interactions. The action tensors of each agent are injected sequentially at input, but do not ensure that their effects remain correctly attributed in the shared visual feature space. Commands may be assigned to the wrong agent or inadvertently drive multiple entities (Fig. 2a), especially when agents overlap, look similar, or jointly manipulate an object. Achieving independent control therefore requires agent-specific conditioning that separates command attribution.

(2) ***unifying action conditioning across heterogeneous control spaces.*** Action vectors do not share a universal semantic coordinate system: their dimensions, ordering, and meanings are defined by task-specific action spaces. As a result, the same action dimension or pattern may encode different behaviors (Fig. 2b). Directly conditioning on raw vectors therefore leads to action semantics being tied to task-specific dimensions and ordering. Language can expose what each control dimension means, but does not preserve the exact frame-synchronized binary or continuous values needed for execution. Unified action conditioning thus requires separating *what a control means* from *what value it takes*, and representing both in a shared action interface.

LiveAct addresses these challenges with a *Heterogeneous Action Interface* (HAI), which factorizes action conditioning into semantic adaptation and agent-specific visual injection. For *independent*

Table 1: **Native capability comparison.** ✓: supported; ×: unsupported.

Method	Domain	Multi-Agent	Real-Time	Semantic	Numerical
LingBot-World 2.0 (Gao et al., 2026)	General visual worlds	×	✓	✓	✓
Solaris (Savva et al., 2026)	Single (*Minecraft*)	✓	×	×	✓
MultiWorld (Wu et al., 2026)	Single (*It Takes Two*)	✓	×	×	✓
ActionParty (Pondaven et al., 2026)	Single (*Melting Pot*)	✓	×	×	✓
Incantation (Zhu et al., 2026b)	Single (*Elden Ring*)	✓	✓	✓	×
WanToFight (Hu et al., 2026)	Single (KOF '97)	✓	✓	×	✓
LiveAct (Ours)	**Diverse**	✓	✓	✓	✓

control, *Agent-Aware Spatial Modulation* derives an agent-specific spatial gate from the agent description, so that each branch selectively updates the visual tokens associated with its intended entity while attending only to the corresponding action stream. For *heterogeneous action conditioning*, *Schema-Guided Action Adaptation* converts each control dimension into a semantic token defined by its schema and injects the synchronized numerical value through affine modulation, decoupling control meaning from task-specific vector layouts. This provides a shared semantic interface for heterogeneous control spaces. Together, these two designs provide a common action representation while preserving agent-specific effects in the shared visual space. The resulting residuals are injected through zero-initialized projections, preserving the pretrained visual prior at initialization.

Beyond accurate control, interaction requires efficient incremental rollout without repeatedly regenerating the visual history. We therefore retain the complete HAI when transferring a bidirectional video teacher to a blockwise causal generator through Teacher Forcing for autoregressive (AR) Tuning and Distribution Matching Distillation (DMD) for Streaming Video (Yin et al., 2024; Huang et al., 2026). Causal temporal alignment maps raw actions to compressed video latents, while a global latent offset advances action windows together with the cached rollout. Causal generation thus accelerates a fixed control interface rather than redefining who acts or what each action means.

We evaluate LiveAct across six multi-agent environments spanning interactive games and robot manipulation, covering both discrete and continuous control spaces. Experiments assess visual fidelity, action adherence, agent attribution, and real-time efficiency. Our main contributions are:

- We formulate real-time multi-agent video world modeling under heterogeneous action spaces and identify two coupled challenges overlooked by existing action interfaces: binding each indexed action stream to its intended visual agent and jointly representing task-specific control semantics and synchronized numerical values in a shared interface.
- We propose LiveAct, a real-time multi-agent video world model built around a Heterogeneous Action Interface. HAI combines Agent-Aware Spatial Modulation for language-driven stream-to-agent binding with Schema-Guided Action Adaptation for value-preserving semantic control, and is retained during teacher-forced causal adaptation and rollout-aware distribution matching for real-time streaming generation.
- We build a dataset of 17,951 multi-agent trajectories, totaling 119.54 hours of raw video, and a 108-clip benchmark spanning four games and two robot-manipulation sources. LiveAct reaches 0.898 and 0.942 agent-attribution accuracy on games and robots without locations, masks, or identity crops, and generates at 17.70 FPS at 512×448.

2 Related Work

Multi-agent video world models. Multi-agent video world models jointly predict an environment under multiple agents' observations or controls, and must coordinate shared dynamics while preserving the identity of each condition stream. Solaris jointly models synchronized observations from multiple Minecraft agents (Savva et al., 2026); MultiWorld combines a multi-agent condition module with a global state encoder across game and robot manipulation (Wu et al., 2026); and Gamma-World uses permutation-equivalent agent encoding and sparse hub attention (Liu et al., 2026). Complementary systems focus on binding independent controls to visual entities: ActionParty jointly predicts persistent subject-state tokens initialized from subject locations (Pondaven et al., 2026), WanToFight grounds locally causal keyboard streams with first-frame character references (Hu et al., 2026), and Incantation binds language actions to individual entities (Zhu et al., 2026b). Across these designs, the central control problem is to preserve per-agent condition provenance and ensure that each action affects its intended entity. The Agent-Aware Spatial Modulation

component of our HAI addresses this problem by conditioning the visual queries on each agent's language description. without masks, boxes, coordinates, or reference images at inference time.

Action-controllable world models across domains. World models learn environment dynamics for imagined interaction (Hafner et al., 2023). Genie learns latent actions from unlabeled videos (Bruce et al., 2024), DIAMOND uses diffusion for Atari and Counter-Strike dynamics (Alonso et al., 2024), and GameNGen predicts DOOM frames from past observations and actions (Valevski et al., 2025). LingBot-World 2.0 couples camera poses and time-varying semantic prompts for open-ended interaction (Gao et al., 2026). Real-world simulators extend action-conditioned prediction to diverse interactions (Yang et al., 2023) and continuous robot trajectories (Zhu et al., 2025), while Incantation expresses per-entity actions in natural language to support semantic transfer (Zhu et al., 2026b). Most interfaces choose between numerically precise but task-specific vectors and language interfaces that make control semantics explicit. Within our HAI, Schema-Guided Action Adaptation combines the two: language identifies the task-dependent meaning of each control variable, while synchronized binary or continuous values retain frame-level control.

3 METHOD

Given an initial frame I_0, a scene description c, and ordered action streams $\mathcal{A} = \{A^{(i)}\}_{i=1}^{N_{\text{ag}}}$ for N_{ag} agents, where $A^{(i)} = [a_{\tau,j}^{(i)}] \in \mathbb{R}^{T\times C}$ gives the value of control dimension j at raw frame $\tau \in \{1,\dots,T\}$ (raw frame 0 is I_0), paired with agent descriptions $G = \{g_i\}$ and a control schema $S = \{s_j\}_{j=1}^{C}$, multi-agent world modeling aims to synthesize the shared visual future in which each stream drives its own agent while all agents remain consistent with one another and with the scene. We write $\mathcal{C} = (c, \mathcal{A}, S, G)$ for the full condition. In this section, we first discuss the motivation and two failure modes of existing action interfaces in Sec. 3.1. We then introduce Agent-Aware Spatial Modulation in Sec. 3.2, which decides *who* responds to each stream, and Schema-Guided Action Adaptation in Sec. 3.3, *what* each control dimension means and *what value* it takes; Sec. 3.4 assembles them into the Heterogeneous Action Interface. Finally, Sec. 3.5 carries the same action-conditioning mechanism into causal streaming rollout. Fig. 3 summarizes this three-stage design.

3.1 MOTIVATION

Although the indexed action tensor preserves the numerical trajectory of each stream, it does not establish the correspondences required to condition a shared visual representation. The stream index i separates $A^{(i)}$ from the other streams but provides no visual evidence associating it with the tokens of its intended agent. Likewise, the control index j identifies a coordinate only within a task-specific action space and therefore lacks a consistent interpretation across tasks.

Cross-Agent Misattribution. Keeping the action streams separate preserves their provenance in action space, but does not establish their recipients in visual space. An agent-index embedding can distinguish $A^{(i)}$ from $A^{(i')}$, yet contains no visual evidence linking index i to the appearance of agent i. Consequently, the command from one stream may be attributed to the wrong visual agent or inadvertently affect multiple agents. The missing correspondence must therefore be established on the query side of action conditioning: the visual features that read $A^{(i)}$ should be grounded in the description g_i of its corresponding agent.

Control-Semantics Collapse. An action vector contains C control dimensions, where the j-th coordinate provides the numerical value of one control at a given time step. However, the index j itself carries no explicit semantic meaning beyond the definition of its task-specific action space. Compressing all C dimensions into a single action embedding further obscures the identity of individual controls, making it difficult for the model to distinguish which control produces a particular effect. The action representation should therefore keep each control dimension distinct while jointly encoding its schema-defined semantics and frame-level numerical value.

These two requirements are complementary: expressive action tokens cannot resolve agent binding, while agent-aware queries cannot recover missing control semantics. HAI therefore separates the two, using Agent-Aware Spatial Modulation for query-side binding and Schema-Guided Action Adaptation for key–value-side control representation, coupled through cross-attention.

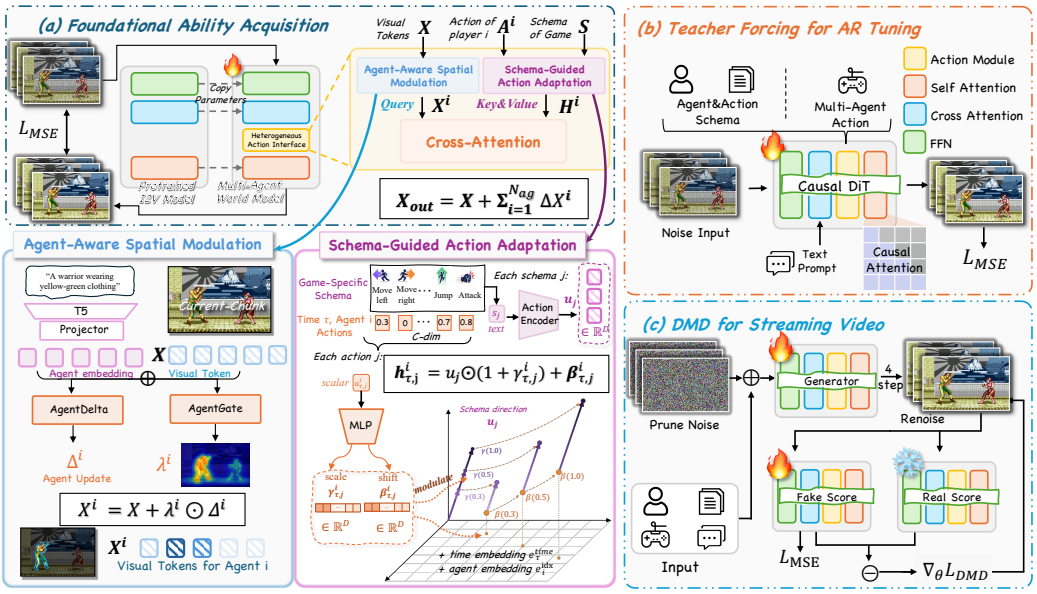

Figure 3: **Overview of LiveAct.** (a) Foundational Ability Acquisition trains a bidirectional teacher with a Heterogeneous Action Interface, which integrates Agent-Aware Spatial Modulation and Schema-Guided Action Adaptation. (b) Teacher Forcing for AR Tuning transfers the same HAI to causal attention. (c) DMD for Streaming Video distills few-step streaming rollouts.

3.2 Agent-Aware Spatial Modulation

To prevent cross-agent misattribution, we condition the visual queries on each agent's description g_i, encouraging each action stream to affect its corresponding visual agent without requiring explicit spatial supervision or inference-time spatial inputs.

Agent semantic representations. By construction, each action sequence $A^{(i)}$ is paired with an agent description g_i. We encode this description with the frozen umT5 (Raffel et al., 2020) text encoder and project it to an agent embedding:

$$e_i^G = \mathrm{Pool}(\mathrm{T5}(g_i)) \in \mathbb{R}^{4096}, \qquad q_i = \mathrm{Proj}_G(e_i^G) \in \mathbb{R}^D. \tag{1}$$

Here Pool denotes a pooling layer and D is the backbone width. The resulting q_i provides the semantic identity associated with action stream $A^{(i)}$.

Spatial query modulation. A global identity vector does not by itself indicate where the corresponding agent appears. For latent frame k, let $X_k = [x_{k,1}, \ldots, x_{k,N}]^\top \in \mathbb{R}^{N\times D}$ contain the N patchified visual tokens. We compute a token-dependent gate from the visual feature, the agent representation, and their element-wise interaction:

$$\lambda_{k,n}^{(i)} = \mathrm{sigmoid}(\mathrm{AgentGate}([x_{k,n}, q_i, x_{k,n} \odot q_i])), \quad x_{k,n}^{(i)} = x_{k,n} + \lambda_{k,n}^{(i)} \mathrm{AgentDelta}(q_i). \tag{2}$$

Here, $\mathrm{AgentDelta}(q_i) \in \mathbb{R}^D$ supplies an identity-dependent feature direction, while the scalar $\lambda_{k,n}^{(i)}$ controls its strength at each spatial token. Because $x_{k,n}$ varies across space and time, the same agent description can induce different modulation strengths throughout the video. The gate is a soft, learned modulation rather than a segmentation mask or an explicitly supervised object assignment.

Stacking the modulated tokens gives $X_k^{(i)} = [x_{k,1}^{(i)}, \ldots, x_{k,N}^{(i)}]^\top$. Each $X_k^{(i)}$ is computed from the same unmodified state X_k and only preconditions the queries for agent i; it is not written into the backbone as a separate residual. The agent-conditioned states remain separate until cross-attention forms the action residuals.

3.3 Schema-Guided Action Adaptation

To prevent control-semantics collapse, we construct a separate action token for each control dimension j at each time step τ. Each token combines the schema description s_j, which specifies *what*

the control means, with its synchronized numerical value $a^{(i)}_{\tau,j}$, which specifies its current state. The resulting tokens are used as keys and values in cross-attention.

Semantic control representations. Raw dimension indices cannot express task-specific control semantics. We therefore encode description s_j of every control variable with the same text encoder:

$$e^S_j = \mathrm{Pool}(\mathrm{T5}(s_j)) \in \mathbb{R}^{4096}, \qquad u_j = \mathrm{Proj}_S(e^S_j) \in \mathbb{R}^D. \tag{3}$$

The resulting token turns coordinate j into a named control such as an attack, steering command, or robot command. The schema description and numerical value are paired by control dimension, allowing a shared representation to accommodate heterogeneous control layouts within a single model.

Value-modulated control tokens. Language alone expresses action meaning but does not preserve the synchronized binary or continuous trajectory. For agent i, raw-frame index τ, and control dimension j, a value encoder shared across control dimensions maps the scalar $a^{(i)}_{\tau,j}$ to affine vectors:

$$(\gamma^{(i)}_{\tau,j}, \beta^{(i)}_{\tau,j}) = \mathrm{MLP}_{\mathrm{mod}}(a^{(i)}_{\tau,j}), \qquad h^{(i)}_{\tau,j} = u_j \odot (1 + \gamma^{(i)}_{\tau,j}) + \beta^{(i)}_{\tau,j}. \tag{4}$$

Thus, u_j specifies *what* a control variable represents, while the synchronized value determines its current modulation. Unlike early aggregation, this construction retains C tokens per time step, allowing visual queries to select individual controls. We add learned time and agent embeddings,

$$\widetilde{h}^{(i)}_{\tau,j} = h^{(i)}_{\tau,j} + e^{\mathrm{time}}_{\tau} + e^{\mathrm{idx}}_{i}, \tag{5}$$

and collect these tokens into $\widetilde{H}^{(i)} \in \mathbb{R}^{T\times C\times D}$, retaining a separate token for every control dimension at every time step before temporal windowing.

3.4 Assembling the Heterogeneous Action Interface

After constructing the agent-conditioned visual queries and schema-guided action tokens, we assemble the HAI through per-agent cross-attention. For latent frame k and agent i, $X^{(i)}_k \in \mathbb{R}^{N\times D}$ queries the aligned action window $H^{(i)}_k \in \mathbb{R}^{M\times D}$ (Eq. 8), where r is the temporal compression ratio of the video VAE and $M = rC$ in the one-latent window used by our main model:

$$\Delta X^{(i)}_k = \mathrm{CrossAttn}(X^{(i)}_k, H^{(i)}_k). \tag{6}$$

The visual tokens serve as queries and the action tokens as keys and values, producing one control residual per agent. We insert this layer after text cross-attention and before the feed-forward sublayer in each transformer block. The shared visual state X_k of latent frame k is then updated as

$$\widehat{X}_k = X_k + \sum\nolimits_{i=1}^{N_{\mathrm{ag}}} \Delta X^{(i)}_k. \tag{7}$$

All agents read from the same shared visual state X_k, while their control residuals are computed independently before aggregation. We zero-initialize each cross-attention output projection, so that the augmented model initially preserves the pretrained backbone behavior.

3.5 Efficient Causal Rollout

To enable streaming rollouts, we convert the bidirectional diffusion teacher into a causal generator while addressing action alignment and exposure to self-generated history. Teacher Forcing for AR Tuning first adapts the model to causal attention, followed by DMD on autoregressive rollouts.

Causally aligned action conditioning. Each video latent should read only the controls of its own physical interval. Since the VAE encodes I_0 separately and compresses every r raw frames into one latent, future latent $k \geq 1$ covers raw frames $\mathcal{I}_k = \{r(k-1)+1, \ldots, rk\}$, and the action memory of agent i stacks the control tokens in this interval:

$$H^{(i)}_k = \big[\widetilde{h}^{(i)}_{\tau,j}\big]_{\tau\in\mathcal{I}_k,\ 1\leq j\leq C} \in \mathbb{R}^{rC\times D}. \tag{8}$$

In the causal generator, k is the global latent index rather than the position within the current chunk, so every latent keeps its physical controls as the KV cache advances.

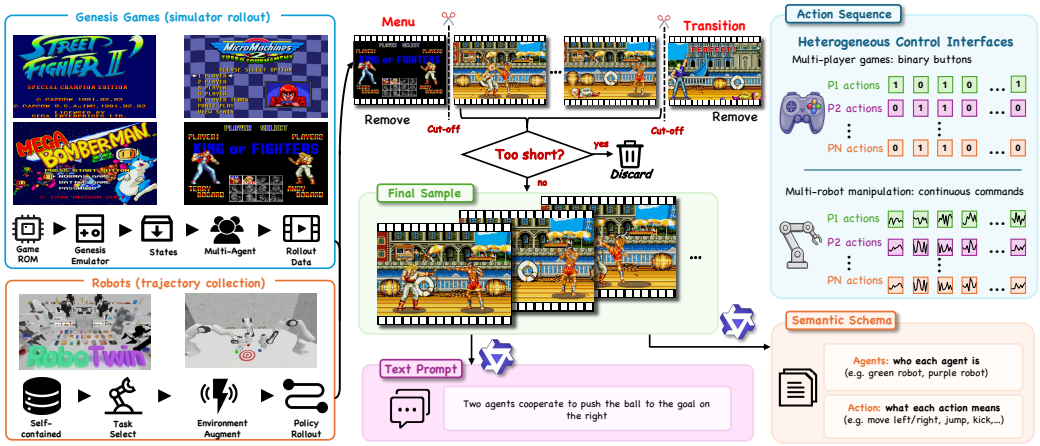

Figure 4: **Data curation pipeline.** Game and robotic episodes are curated into aligned samples containing a scene prompt, per-agent action streams, agent descriptions, and task-specific schemas.

Teacher Forcing for AR Tuning. After obtaining a bidirectional model, this stage then transfers its parameters to a blockwise causal generator (Zhu et al., 2026a). Each target chunk is denoised using only clean preceding chunks, with a causal mask. The HAI remains at the same transformer locations and receives the same aligned conditions, so this stage isolates the change in temporal information flow from errors caused by self-generated history. At inference, preceding chunks are retained in the KV cache and reused when generating the next one.

DMD for Streaming Video. Teacher forcing still trains on clean history, leaving an exposure gap at deployment. DMD addresses this gap by optimizing autoregressive rollouts (Yin et al., 2024; Huang et al., 2026). Starting from the initial latent z_0, the VAE encoding of I_0, the generator produces $\hat{z} = \mathcal{G}_\theta(z_0, \mathcal{C})$ while reusing its self-generated KV cache. The frozen bidirectional teacher serves as the real score, while a trainable bidirectional fake score initialized from the same teacher tracks the generator distribution. At diffusion timestep t, their denoised predictions from the same noised sample $\hat{z}_t$ and condition $\mathcal{C}$ define the unnormalized distribution-matching gradient $\delta_{\text{DMD}} = \mu_{\text{fake},\phi}(\hat{z}_t, t, \mathcal{C}) - \mu_{\text{real}}(\hat{z}_t, t, \mathcal{C})$. Backpropagating the normalized gradient through $\mathcal{G}_\theta$ exposes the causal model to its own rollout history while retaining few-step chunk-wise generation.

4 Experiments

4.1 Data Collection

As shown in Fig. 4, we assemble a six-source dataset of 17,951 independent multi-agent trajectories from four competitive multiplayer games and collaborative multi-robotic sources, totaling 119.54 hours of raw video. Each sample contains an initial frame, a scene description, synchronized per-agent action streams, agent descriptions, and a task-specific control schema, with binary controls for games and continuous controls for robots. For evaluation, we build a benchmark of 108 clips drawn from trajectories held out from training, with one 81-frame clip per trajectory; all comparisons and ablations in this section are conducted on this benchmark. Details are provided in Appendix.

4.2 Implementation Details

We build LiveAct on Wan2.2-TI2V-5B (Wan et al., 2025), which contains 30 transformer blocks with a hidden width of $D = 3072$ and 24 attention heads; its VAE compresses every $r = 4$ raw frames into one latent. We insert the HAI after text cross-attention and before the feed-forward sublayer in every block and fine-tune the full DiT together with the HAI, while the umT5 encoder, which encodes agent and schema descriptions, and the VAE remain frozen. Each agent is represented by $C = 12$ schema-aligned control dimensions. Stage 1 trains the bidirectional teacher for 10k steps with a learning rate of 10^{-4} and a global batch size of 16. Stage 2 then trains the causal generator for 6k steps with a learning rate of 2×10^{-6}, and Stage 3 runs 700 iterations with learning rates of 2×10^{-6} for the generator and 4×10^{-7} for the fake score; both causal stages use a global batch size

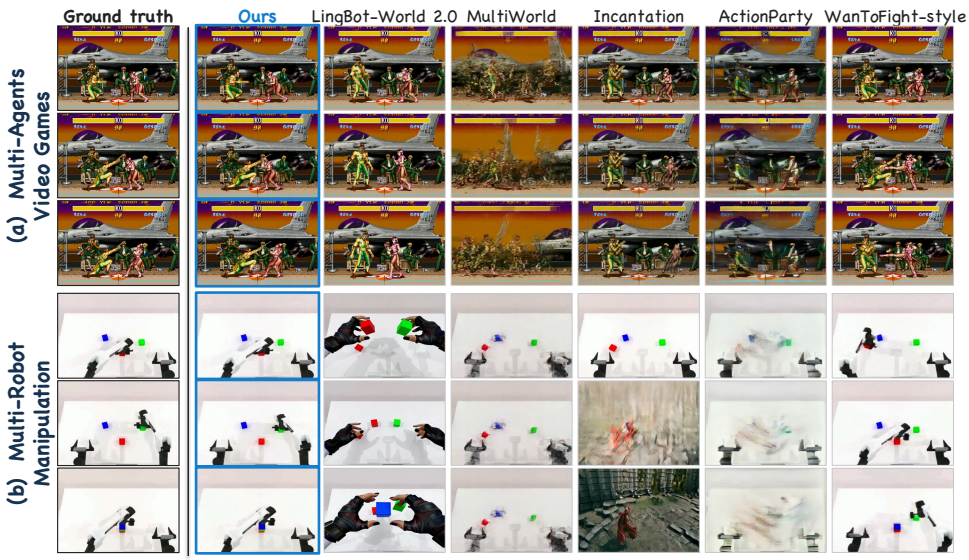

Figure 5: **Qualitative comparison on shared initial frames and action streams.** Columns show ground truth, the five baselines, and LiveAct; Rows are frames sampled at different durations.

of 24 with FSDP. At inference, the teacher uses 50 denoising steps, and the causal generator predicts four latent frames per chunk with four denoising evaluations at timesteps $[1000, 750, 500, 250]$.

4.3 Comparison with Baselines

Baselines. We evaluate LiveAct on two complementary tracks—*Multi-Agent Video Games* and *Multi-Robot Manipulation*—under the same protocol for visual quality, action following, and agent attribution. We compare against representative interactive and multi-agent video world models covering three control paradigms: language-serialized actions (LingBot-World 2.0 (Gao et al., 2026), Incantation (Zhu et al., 2026b)), per-agent numerical controls (MultiWorld (Wu et al., 2026), ActionParty (Pondaven et al., 2026)), and reference-guided method (WanToFight-style (Hu et al., 2026)).

Metrics. Following MultiWorld, we report FVD, LPIPS, PSNR, and SSIM for reconstruction quality. Because frame-paired metrics can favor static or averaged predictions, we complement them with Motion Magnitude Error (MME) and an Aesthetic score using the LAION (Schuhmann et al., 2022) improved predictor. For a more comprehensive control evaluation, we trained inverse dynamics models only on real clips: they recover the executed controls from a generated video, yielding Act-F1 over discrete game events and a normalized error (NRMSE) on continuous robot commands. To evaluate agent attribution, Attr-Acc measures whether the recovered per-agent controls match the commanded streams better under the intended agent assignment than under pairwise agent swaps; higher is better, with a chance level of 0.5. See appendix for more details.

Quantitative Results. Shown in Tab. 2, among the four non-reference-conditioned methods, LingBot-World 2.0 gives the best game FVD and Aesthetic score but follows actions weakly; MultiWorld retains competitive robot PSNR and SSIM but has low agent attribution; Incantation reaches comparable game Act-F1 yet suffers severe visual drift on robots; and ActionParty provides the best robot reconstruction but the weakest action following. WanToFight-style is the strongest baseline overall, combining strong reconstruction with 0.886 robot Attr-Acc using identity crops, although its game Attr-Acc remains 0.552. LiveAct obtains the best results among the evaluated methods on both tracks, with FVDs of 338.5 and 270.2, game Act-F1 of 0.594, robot NRMSE of 0.558, and Attr-Acc of 0.898 and 0.942. It also achieves the best MME and obtains the highest Aesthetic score.

Qualitative Results. As shown in Fig. 5, the language-serialized models lose the robot scene (LingBot-World 2.0, Incantation). On the game track, they keep the stage but not the commanded attacks. MultiWorld and ActionParty keep the coarse layout but blur the agents: the fighters and arms blur into faint, duplicated silhouettes. WanToFight-style preserves both scenes and agents and executes the kick, but misses the second agent's command. LiveAct reproduces the kick and the opponent's response at the same frames as the ground truth and carries the arm along the ground-truth path, showing that each stream is both executed and bound to the intended agent.

Table 2: **Main comparison across multi-agent video games and multi-robot manipulation.** Best results are bolded. [a]Native MultiWorld uses synchronized multi-view observations; we have set it to a fixed same perspective. [b]The code is not released; this is our reproduction.

	Reconstruction				*Control*		*Diagnostics*	
Method	FVD ↓	LPIPS ↓	PSNR ↑	SSIM ↑	Act-F1 ↑ / NRMSE ↓	Attr-Acc ↑	MME ↓	Aesthetic ↑
Multi-Agent Video Games								
LingBot-World 2.0 (Gao et al., 2026)	1256.6	0.322	13.25	0.490	0.146	0.553	0.57	5.093
MultiWorld (1-view)[a] (Wu et al., 2026)	2661.4	0.719	10.40	0.125	0.163	0.548	2.67	4.059
Incantation (Zhu et al., 2026b)	1989.5	0.593	10.45	0.226	0.162	0.534	8.77	4.539
ActionParty (Pondaven et al., 2026)	1672.5	0.272	14.56	0.467	0.086	0.547	0.47	4.974
WanToFight-style[b] (Hu et al., 2026)	563.6	0.136	17.67	0.671	0.241	0.552	0.23	5.192
LiveAct	**338.5**	**0.086**	**20.17**	**0.743**	**0.594**	**0.898**	**0.15**	**5.207**
Multi-Robot Manipulation								
LingBot-World 2.0 (Gao et al., 2026)	2732.5	0.603	7.13	0.464	1.106	0.665	4.20	3.570
MultiWorld (1-view)[a] (Wu et al., 2026)	1748.9	0.276	15.36	0.797	1.099	0.357	1.90	4.054
Incantation (Zhu et al., 2026b)	3053.3	0.820	3.93	0.185	1.119	0.459	17.37	3.710
ActionParty (Pondaven et al., 2026)	1661.7	0.267	15.42	0.816	1.142	0.504	1.90	4.035
WanToFight-style[b] (Hu et al., 2026)	402.9	0.188	17.32	0.831	0.758	0.886	0.42	4.037
LiveAct	**270.2**	**0.106**	**20.51**	**0.871**	**0.558**	**0.942**	**0.42**	**4.085**

4.4 Ablation Studies

Effectiveness of Agent-Aware Spatial Modulation (AASM). Removing it keeps the action tokens intact but makes the queries identical for both agents, and control collapses in both domains (Tab. 3): game Act-F1 drops from 0.594 to 0.172 and Attr-Acc from 0.898 to 0.592, while robot NRMSE rises from 0.558 to 0.931. In Fig. 6, the two streams no longer stay on their own agents, which is the cross-agent misattribution of Sec. 3.1: binding cannot be recovered from the action side alone.

Effectiveness of Schema-Guided Action Adaptation (SGAA). Replacing it with a conventional flat action encoder, an MLP that maps each frame's full action vector to a single token, keeps agent modulation but reintroduces control-semantics collapse in visual, and performance again drops in both domains: game Act-F1 falls to 0.225 and Attr-Acc to 0.603, while robot NRMSE rises to 1.120 and Attr-Acc falls to 0.484. Attribution degrades even though agent modulation is retained, indicating that the queries can only route controls they can interpret; the two components are therefore complementary. Finer-grained SGAA ablations are provided in the Appendix.

Table 3: **Quantitative ablation study.**

Variant	Game Act-F1 ↑	Game Attr-Acc ↑	Robot NRMSE ↓	Robot Attr-Acc ↑
Full LiveAct	**0.594**	**0.898**	**0.558**	**0.942**
w/o AASM	0.172	0.592	0.931	0.810
w/o SGAA	0.225	0.603	1.120	0.484

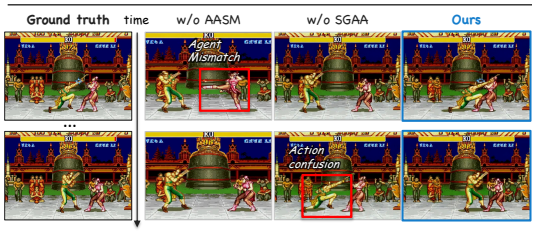

Figure 6: **Qualitative ablation study.**

Effectiveness of real-time distillation. Stage 1 denoises the full sequence and cannot stream. Stage 2 enables blockwise causal generation but remains slow at 1.66 FPS. Stage 3 reduces each chunk to four denoising steps and reaches 17.70 FPS, 10.7× faster than Stage 2. On games, Act-F1 rises from 0.539 to 0.594 and Attr-Acc from 0.844 to 0.898. Autoregressive generation is better suited to these discrete actions: each frame is conditioned only on past context and the controls aligned to that frame, rather than denoising the full sequence jointly. On robots, NRMSE rises from 0.430 to 0.558 and Attr-Acc falls from 0.969 to 0.942. Per-stage settings and results are in the Appendix.

5 Conclusion

We presented LiveAct, a multi-agent video world model for heterogeneous control across games and robot manipulation. Its Heterogeneous Action Interface combines Agent-Aware Spatial Modulation, which binds each indexed action stream to its visual agent through language-conditioned queries, with Schema-Guided Action Adaptation, which couples each control's language-defined semantics with its synchronized numerical value. Per-agent cross-attention integrates the two within a shared visual representation. Teacher Forcing for AR Tuning and DMD for Streaming Video transfer the interface unchanged to few-step causal rollout. Across six sources, LiveAct achieves best reconstruction and control and reaches 0.898 and 0.942 agent-attribution accuracy without location, mask, or identity input, while its causal generator runs at 17.70 FPS at 512×448.

AI USE STATEMENT

Generative AI tools were used to assist with language polishing, structural revision, and LaTeX formatting. All AI-assisted outputs were reviewed and verified by the authors, who take responsibility for the final content, including its text, claims, and artifacts.

REPRODUCIBILITY STATEMENT

The architecture of the Heterogeneous Action Interface and its placement in the backbone are specified in Sec. 3, and the three-stage training procedure, learning rates, batch sizes, step counts, and sampling schedules are given in Sec. 4 and in the Appendix (Backbone and Conditioner Placement; Training and Inference Configuration), together with the frame–latent–action alignment and the causal training objectives. The six data sources, their statistics, and the construction of the 108-clip benchmark are described in Sec. 4.1 and the Appendix (Dataset Statistics). The evaluation protocol, the definitions of all metrics including the inverse-dynamics evaluator and the swap-based attribution test, and the configuration of every baseline, including our reproduction of WanToFight-style, are documented in the Appendix. All methods are evaluated on the same clips with a fixed seed, and throughput is measured at 512×448 on a single NVIDIA A800. We will release data preprocessing and split definitions, schema and agent-description metadata, model configurations, checkpoint-conversion utilities, evaluation seeds, and evaluation scripts. Data and derived recordings will be distributed only where source licenses permit.

ETHICS STATEMENT

This work uses only emulator recordings of retro video games and simulated robot manipulation; no human subjects, personal data, or real-world footage are involved. The game sources contain stylized cartoon violence typical of fighting and action games, and we use them solely as controlled multi-agent environments. The robot data come from simulators, so the model has not been validated on physical hardware and should not be used as a substitute for physical safety testing. As with other video world models, the generator could in principle be adapted to synthesize misleading media; the narrow, game- and simulation-specific training distribution limits this risk, and any released model will carry usage terms that prohibit impersonation and deceptive content. We do not redistribute game ROMs, and derived recordings will be released only where licenses permit. We are not aware of other negative societal impacts specific to this work.

REFERENCES

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. *Advances in Neural Information Processing Systems*, 37:58757–58791, 2024.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, et al. Motus: A unified latent action world model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35101–35113, 2026.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first international conference on machine learning*, 2024.

Remi Cadene, Simon Alibert, Francesco Capuano, Michel Aractingi, Adil Zouitine, Pepijn Kooijmans, Jade Choghari, Martino Russi, Caroline Pascal, Steven Palma, et al. Lerobot: An open-source library for end-to-end robot learning. In *International Conference on Learning Representations*, volume 2026, pp. 122398–122417, 2026.

Zelin Gao, Qiuyu Wang, Jiapeng Zhu, Jingye Chen, Zichen Liu, Qingyan Bai, Jiahao Wang, Yufeng Yuan, Hanlin Wang, Yichong Lu, et al. Infinite worlds with versatile interactions. *arXiv preprint arXiv:2607.07534*, 2026.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in neural information processing systems*, 35:8633–8646, 2022.

Yicong Hong, Yiqun Mei, Chongjian Ge, Yiran Xu, Yang Zhou, Sai Bi, Yannick Hold-Geoffroy, Mike Roberts, Matthew Fisher, Eli Shechtman, et al. Relic: Interactive video world model with long-horizon memory. *arXiv preprint arXiv:2512.04040*, 2025.

Li Hu, Guangyuan Wang, Peng Zhang, and Bang Zhang. Wantofight: Real-time generative game engine for multi-player combat interaction. *arXiv preprint arXiv:2607.12592*, 2026.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *Advances in Neural Information Processing Systems*, 38:167283–167308, 2026.

Jiaqi Li, Junshu Tang, Zhiyong Xu, Longhuang Wu, Yuan Zhou, Shuai Shao, Tianbao Yu, Zhiguo Cao, and Qinglin Lu. Hunyuan-gamecraft: High-dynamic interactive game video generation with hybrid history condition. *arXiv preprint arXiv:2506.17201*, 2(3):6, 2025.

Fangfu Liu, Kai He, Tianchang Shen, Tianshi Cao, Sanja Fidler, Yueqi Duan, Jun Gao, Igor Gilitschenski, Zian Wang, and Xuanchi Ren. Gamma-world: Generative multi-agent world modeling beyond two players. *arXiv preprint arXiv:2605.28816*, 2026.

Yao Mu, Tianxing Chen, Shijia Peng, Zanxin Chen, Zeyu Gao, Yude Zou, Lunkai Lin, Zhiqiang Xie, and Ping Luo. Robotwin: Dual-arm robot benchmark with generative digital twins (early version). *arXiv preprint arXiv:2409.02920*, 2024.

Alex Nichol, Vicki Pfau, Christopher Hesse, Oleg Klimov, and John Schulman. Gotta learn fast: A new benchmark for generalization in rl. *arXiv preprint arXiv:1804.03720*, 2018.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Mathieu Poliquin. Stable retro, a maintained fork of openai's gym-retro. https://github.com/Farama-Foundation/stable-retro, 2026.

Alexander Pondaven, Ziyi Wu, Igor Gilitschenski, Philip Torr, Sergey Tulyakov, Fabio Pizzati, and Aliaksandr Siarohin. Actionparty: Multi-subject action binding in generative video games. In *European Conference on Computer Vision*, pp. 74–92. Springer, 2026.

Yiran Qin, Li Kang, Xiufeng Song, Zhenfei Yin, Xiaohong Liu, Xihui Liu, Ruimao Zhang, and Lei Bai. Robofactory: Exploring embodied agent collaboration with compositional constraints. *arXiv preprint arXiv:2503.16408*, 2025.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Georgy Savva, Oscar Michel, Daohan Lu, Suppakit Waiwitlikhit, Timothy Meehan, Dhairya Mishra, Srivats Poddar, Jack Lu, and Saining Xie. Solaris: Building a multiplayer video world model in minecraft. *arXiv preprint arXiv:2602.22208*, 2026.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022.

HunyuanWorld Team, Zhenwei Wang, Yuhao Liu, Junta Wu, Zixiao Gu, Haoyuan Wang, Xuhui Zuo, Tianyu Huang, Wenhuan Li, Sheng Zhang, et al. Hunyuanworld 1.0: Generating immersive, explorable, and interactive 3d worlds from words or pixels. *arXiv preprint arXiv:2507.21809*, 2025.

Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pp. 402–419. Springer, 2020.

Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. In *International Conference on Learning Representations*, volume 2025, pp. 73754–73776, 2025.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Haoyu Wu, Jiwen Yu, Yingtian Zou, and Xihui Liu. Multiworld: Scalable multi-agent multi-view video world models. *arXiv preprint arXiv:2604.18564*, 2026.

Sherry Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 2023.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6613–6623. IEEE, 2024.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018.

Songchun Zhang, Yaowei Li, Junhao Zhuang, Weiyang Jin, Haoyu Wang, Xin Lu, Yilang Sun, Shiyi Zhang, Haoran Li, Xiaoxiao Ma, et al. Echowm: Open and enterable omnimodal world models. *arXiv preprint arXiv:2608.23189*, 2026.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. Irasim: A fine-grained world model for robot manipulation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9834–9844. IEEE, 2025.

Hongzhou Zhu, Min Zhao, Guande He, Hang Su, Chongxuan Li, and Jun Zhu. Causal forcing: Autoregressive diffusion distillation done right for high-quality real-time interactive video generation. *arXiv preprint arXiv:2602.02214*, 2026a.

Shangwen Zhu, Qianyu Peng, Zhao Pu, Zhilei Shu, Xiangrui Ke, Zhaohu Xing, Zizhao Tong, Zeqing Wang, Xinyu Cui, Zian Zheng, et al. Incantation: Natural language as the action interface for multi-entity video world models. *arXiv preprint arXiv:2605.18601*, 2026b.

APPENDIX

A Additional Related Work **S1**

B Method Details **S2**

B.1 Problem Formulation S2

B.2 Frame–Latent–Action Alignment S2

B.3 Causal Condition Propagation S2

B.4 Causal Training Objectives S3

C Dataset Details **S3**

D Implementation Details **S4**

D.1 Backbone and Conditioner Placement S4

D.2 Training and Inference Configuration S5

E Evaluation Details **S5**

E.1 Evaluation Protocol S5

E.2 Baseline Configurations S6

E.3 Metrics S6

F Additional Results **S7**

F.1 Three-Stage Causal Transfer S7

F.2 Detailed Ablations S7

F.3 Spatial Localization of Agent-Specific Residuals S9

F.4 Additional Qualitative Comparisons S9

F.5 Additional LiveAct Showcases S10

F.6 Long-Horizon Rollout Showcase S12

Supplementary video. Please see the accompanying `demo.mp4` for additional video results.

A ADDITIONAL RELATED WORK

Structured conditioning in video diffusion. Video diffusion models provide temporally coherent generative priors (Ho et al., 2022), while diffusion transformers and open video foundation models offer scalable token backbones (Peebles & Xie, 2023; Wan et al., 2025). Control signals are commonly injected through concatenation, adaptive modulation, or cross-attention. Our distinction is representational rather than merely architectural: a padded action vector is decomposed into per-dimension control tokens, each token carries a task-specific semantic description, and its time-varying scalar modulates rather than replaces that semantic representation. Agent-Aware Spatial Modulation then keeps each stream's provenance intact until cross-attention.

Causal video generation and diffusion distillation. Streaming video generation restricts attention to past chunks and caches their keys and values. Self Forcing reduces exposure bias by training on autoregressive model rollouts (Huang et al., 2026); Causal Forcing motivates causal teacher initialization before few-step distillation (Zhu et al., 2026a); and distribution matching distillation expresses the generator update through real- and fake-distribution scores (Yin et al., 2024). Recent multi-agent systems also combine bidirectional teachers, causal students, and rollout-aware distilla-

tion (Savva et al., 2026; Liu et al., 2026; Hu et al., 2026). We adopt this training paradigm to carry the complete HAI into efficient rollout while keeping physical action intervals aligned.

B Method Details

B.1 Problem Formulation

We follow the notation of Sec. 3: N_{ag} agents with ordered action streams $\mathcal{A} = \{A^{(i)}\}_{i=1}^{N_{\text{ag}}}$, $A^{(i)} = [a_{\tau,j}^{(i)}] \in \mathbb{R}^{T\times C}$, a control schema $S = \{s_j\}_{j=1}^{C}$, and agent descriptions $G = \{g_i\}_{i=1}^{N_{\text{ag}}}$. The interface enforces

$$a_{\tau,j}^{(i)} \longleftrightarrow (g_i, s_j), \qquad i \in \{1,\dots,N_{\text{ag}}\}, \quad j \in \{1,\dots,C\}, \tag{9}$$

so each numerical control dimension is paired with an agent identity and a task-specific semantic description; the controls of every source are padded to a common width of $C = 12$, while S preserves source-specific control semantics.

Let z_0 be the VAE encoding of I_0 and $z^{1:F-1}$ the future video latents over F latent frames in total. The DiT backbone patchifies these latents into visual tokens $X \in \mathbb{R}^{FN\times D}$, where each latent frame contributes N spatial tokens of hidden dimension D. The desired conditional distribution is

$$p_\theta(z^{1:F-1} \mid z_0, c, \mathcal{A}, S, G). \tag{10}$$

The bidirectional teacher models all future tokens jointly. The causal generator instead partitions future latents into chunks $\mathcal{B}_1,\dots,\mathcal{B}_K$ and factorizes

$$p_\theta(z^{1:F-1} \mid z_0, \mathcal{C}) = \prod_{b=1}^{K} p_\theta(z^{\mathcal{B}_b} \mid z^{<\mathcal{B}_b}, z_0, \mathcal{C}), \quad \mathcal{C} = (c, \mathcal{A}, S, G). \tag{11}$$

Eq. 9 makes both agent identity and control semantics explicit; the main-text Method realizes this interface and keeps it under causal rollout.

B.2 Frame–Latent–Action Alignment

The Wan VAE uses temporal compression ratio $r = 4$ with a separately encoded first frame. Action sequences are truncated or padded to $T_{\text{need}} = (F-1)r+1$. For a window of L latent frames, latent frame zero repeats the first action rL times, while future latent $k \geq 1$ receives

$$\{a_{\tau,j}^{(i)} :\ \tau_k \leq \tau \leq \tau_k + rL - 1\}, \qquad \tau_k = 1 + r(k-1), \tag{12}$$

with final-value padding at the sequence boundary; for the primary setting $L = 1$ this reduces to the interval $\mathcal{I}_k$ of Eq. 8. The time and control-dimension axes are retained until each local window is formed and then flattened for action attention.

For an 81-frame clip, raw indices 0:80 map to 21 latent frames. Future latent $k \in \{1,\dots,20\}$ uses actions beginning at $1 + 4(k-1)$, and chunk $b \in \{1,\dots,5\}$ contains future latents $4(b-1)+1{:}4b$. During causal teacher forcing, the clean-context/noisy-target shift advances the action interval by one latent. During DMD training, the global latent index is applied not only to the generator's rollout but also to both score models, so all three read the same physical interval. Self-forced score evaluation restores the clean initial latent and excludes it from both DMD and fake-score losses.

B.3 Causal Condition Propagation

In the causal generator, the HAI operates only on the real video tokens of the target chunks and returns zero residuals for sequence-padding tokens. Under the clean-context/noisy-target layout used by teacher forcing, actions are injected only into the noisy target half; the clean history remains an unmodified context. During rollout, the clean initial latent is cached without action injection before subsequent target chunks receive their aligned streams. This separation prevents context tokens from being optimized as future action responses.

Checkpoint transfer is performed only between networks with matching HAI architectures and conditioning dimensions.

B.4 Causal Training Objectives

Foundational Ability Acquisition. For clean future latents $z^{1:F-1}$, Gaussian noise ϵ, and sampled time t, let $z_t^{1:F-1} = \alpha_t z^{1:F-1} + \sigma_t \epsilon$. The bidirectional model predicts the velocity target $v^*(z^{1:F-1}, \epsilon, t)$ under the initial latent, scene, and action conditions:

$$\mathcal{L}_{\text{bi}} = \mathbb{E}\left[w(t)\left\|v_{\theta_b}(z_t^{1:F-1}, z_0, t, \mathcal{C}) - v^*\right\|_2^2\right]. \tag{13}$$

It provides both the action-conditioned prior and initialization for the causal generator and score networks.

Teacher Forcing for AR Tuning. Given the $F = 21$ clean latents $z^{0:F-1}$, the training pair uses $z^{0:F-2}$ as context and $z^{1:F-1}$ as targets. The initial frame is conditioning only and is excluded from the loss. We add noise to the $F - 1 = 20$ targets, with every group of four latents sharing a timestep to match chunks:

$$\mathcal{L}_{\text{TF}} = \mathbb{E}\left[w(t)\left\|v_{\theta_c}(z_t^{1:F-1}, z^{0:F-2}, t, \mathcal{C}_{\text{shift}}) - v^*\right\|_2^2\right]. \tag{14}$$

Here $\mathcal{C}_{\text{shift}}$ applies the four-raw-frame offset induced by the context–target shift.

DMD for Streaming Video. The trainable causal generator $\mathcal{G}_\theta$, which unrolls the causal network v_{θ_c} with the four-step schedule, initializes its KV cache from the clean initial latent and generates five future chunks from noise without reading future ground-truth latents. The real score μ_{real} and the fake score $\mu_{\text{fake},\phi}$ of the main paper receive the same initial frame, scene text, control schema, agent descriptions, and cropped action window as the generator. For $\hat{z} = \mathcal{G}_\theta(z_0, \mathcal{C})$ and its noised version $\hat{z}_t$, we normalize the unnormalized distribution-matching gradient δ_{DMD} of the main paper as

$$\bar{\delta}_{\text{DMD}} = \frac{\delta_{\text{DMD}}}{\text{mean}(|\hat{z} - \mu_{\text{real}}(\hat{z}_t, t, \mathcal{C})|) + \varepsilon}, \tag{15}$$

with a small constant ε. We then optimize the generator through stop-gradient regression,

$$\mathcal{L}_{\text{DMD}} = \frac{1}{2}\left\|\hat{z} - \text{sg}(\hat{z} - \bar{\delta}_{\text{DMD}})\right\|_2^2. \tag{16}$$

The fake score is alternately trained by flow denoising regression on detached samples from the current generator. The default schedule updates it every iteration and updates the generator once every five iterations. We use conditional-only score passes to avoid an additional unconditional forward.

C Dataset Details

Composition. Tab. 1 summarizes the six sources and Fig. 1 shows frames from each of them. The dataset contains 17,951 independent multi-agent trajectories, 119.54 hours of raw video, with two to four concurrently controlled agents: the two fighting games and RoboTwin are two-agent, Mega Bomberman and Micro Machines include scenes with two to four agents, and RoboFactory includes two- and three-arm scenes. Every trajectory pairs a shared visual observation with one synchronized action stream per agent, so the number of streams equals the number of agents in the scene.

Sources. The game portion is collected through the retro game platform (Poliquin, 2026; Nichol et al., 2018). These games cover versus fighting, shared-map competition, and shared-track racing. Each trajectory begins from a curated emulator state and records synchronized visual observations and per-agent controller traces under scripted policies at the native 60 fps. The policy drives each agent's controller independently: it samples one allowed button or a no-op from a game-specific distribution, holds it for a fixed number of frames, and then releases it. We exclude menus, transitions, and incomplete rollouts, yielding 12,860 trajectories of 30 s each.

The robot portion combines RoboTwin (Mu et al., 2024) and RoboFactory (Qin et al., 2025). RoboTwin contributes 2,532 episodes across 50 multi-arm tasks: 280 self-collected simulations of 14 tasks at 30 fps and 2,252 augmented episodes covering all 50 tasks from the randomized LeRobot (Cadene et al., 2026) collection at 50 fps. RoboFactory contributes 2,559 episodes of cooperative cube stacking at 30 fps, retaining both successful and unsuccessful trajectories and including



Figure 1: **Showcase of our dataset.** Frames from the six sources, covering versus fighting, shared-map competition, shared-track racing, and multi-arm manipulation.

Table 1: **Dataset statistics for four multi-agent games and two multi-robot manipulation sources.** Frame counts and durations refer to the raw recordings at their native frame rates (RoboTwin: 30 fps for self-collected and 50 fps for augmented episodes). Training and evaluation use 81-frame clips drawn from these trajectories.

	Street Fighter II	Fatal Fury 2	Mega Bomberman	Micro Machines	RoboTwin	RoboFactory	**Total**
Trajectories	5460	3200	2050	2150	2532	2559	**17951**
Agents	2	2	2–4	2–4	2	2–3	**2–4**
Action type	binary	binary	binary	binary	continuous	continuous	**mixed**
Frame rate (fps)	60	60	60	60	30 / 50	30	–
Raw frames (M)	9.83	5.76	3.69	3.87	0.65	0.91	**24.71**
Raw duration (h)	45.50	26.67	17.08	17.92	3.93	8.44	**119.54**

two- and three-arm scenes. In both sources, concurrent continuous action streams are aligned with a shared visual observation over the full episode, yielding 5,091 robot trajectories.

Scale and training windows. The raw recordings total 24.71 M frames (games 23.15 M, 107.17 h; robots 1.56 M, 12.37 h), or 119.54 hours at the native frame rates (Tab. 1). Individual trajectories are much longer than a training sample, whereas the model consumes 81-frame windows (Tab. 2). Because generation is conditioned on an initial frame, a window may start at any frame of a trajectory, with its first frame serving as the conditioning image.

Benchmark. The 108-clip benchmark of the main paper is kept separate from the training data: it takes one 81-frame clip from each of 108 trajectories that are never used for training, and contains 78 game and 30 robot clips.

D Implementation Details

D.1 Backbone and Conditioner Placement

Beyond the placement described in the main paper, the HAI leaves the pretrained conditioning pathways of Wan2.2-TI2V-5B (Wan et al., 2025) unchanged. The initial image latent is fused into the 48-channel video representation, and a separated timestep assigns zero to this clean latent and the

sampled diffusion timestep to future latents; scene text retains the pretrained text cross-attention, diffusion time continues to modulate AdaLN, and video positions retain 3D RoPE. The bidirectional teacher, the causal generator, and both score networks share the same HAI placement and structure.

Control-schema and agent-description embeddings are precomputed with the frozen text encoder and reused throughout training; the pooling operator Pool of the main paper averages the encoder outputs over the non-padding tokens of each description.

D.2 Training and Inference Configuration

Tab. 2 collects the backbone, data, and optimization settings of the three stages in Fig. 3, including those stated in the main paper. Beyond them, the primary configuration uses description-based, value-modulated control tokens, and all stages share a flow-matching parameterization with 1000 training timesteps and a timestep shift of 5.0. Foundational Ability Acquisition uses AdamW with $\beta = (0.9, 0.999)$ and a constant learning rate, whereas the two causal stages use $\beta = (0, 0.999)$ and clip gradient norms at 10. DMD for Streaming Video maintains an EMA of the generator with decay 0.99 from iteration 200 onward, and the EMA weights are used at inference.

Table 2: **Training and inference configuration of LiveAct.** Stages 2 and 3 are trained with FSDP, and Stage 3 samples at timesteps $[1000, 750, 500, 250]$ with the EMA weights; "–" marks settings that do not apply.

Model and data			
Backbone	Wan2.2-TI2V-5B; 30 blocks, width 3072, 24 heads; VAE $4 \times 16 \times 16$		
Trainable	Full DiT and HAI; umT5 encoder and VAE frozen		
Clip	81 frames at 512×448, 16 fps; 1 conditioning and 20 future latents		
Frame sampling	Games: consecutive native frames; robots: uniform over the episode		
Actions	12 description-modulated controls per agent; one-latent window		
Conditioner	Text projections $4096 \to 1024 \to 3072$; value/gate/delta bottleneck 768		
Training and inference	**Stage 1** Foundation	**Stage 2** Teacher Forcing	**Stage 3** DMD
Optimizer	AdamW, weight decay 0.01, BF16		
Training steps	10k	6k	700
Global batch size	16	24	24
Learning rate	10^{-4}	2×10^{-6}	2×10^{-6}
Fake-score LR	–	–	4×10^{-7}
AdamW β	$(0.9, 0.999)$	$(0, 0.999)$	$(0, 0.999)$
Gradient clipping	–	10	10
Generator update	every step	every step	every 5 steps
Generator EMA	–	–	0.99 from step 200
Denoising steps	50	50	4
Classifier-free guidance	5.0	5.0	–

E Evaluation Details

E.1 Evaluation Protocol

Both evaluation tracks of Sec. 4.3 ask the same three questions. **First**, does Agent-Aware Spatial Modulation assign each indexed action stream to the correct visual agent in the jointly generated scene? **Second**, does Schema-Guided Action Adaptation interpret heterogeneous controls consistently across games and embodied tasks? **Third**, how much control and visual quality survive causal transfer, and what latency is gained? The first two questions evaluate the complementary components of the HAI.

Every reported method is evaluated on the benchmark of App. C with one 81-frame prediction at seed 0 per condition. Frame-paired metrics exclude the conditioning frame and average 16 uniformly sampled frames from frames 1–80. Pixel metrics use 256^2 inputs, while FVD uses canonical

Kinetics-400 I3D pre-softmax features at 224^2. Since every clip starts from a fixed initial frame and a fixed set of action streams in a deterministic environment, frame-paired errors measure deviation from the commanded reference future; the control metrics separately assess execution and attribution. Causal efficiency is evaluated separately because published throughput values use different backbones, resolutions, hardware, and decoding pipelines. Architecture ablations use the same data, optimization budget, and evaluation seed.

E.2 Baseline Configurations

All five baselines are evaluated on the same clips and protocol. LingBot-World 2.0, MultiWorld, and Incantation do not release training code, so we run their released checkpoints with their original model configurations and conditioning interfaces, without further training. ActionParty does not release trained checkpoints, so we train it with the same budget on our dataset. The WanToFight code is not publicly available, so we train our own reimplementation on the same mixture and denote it WanToFight-style.

- **LingBot-World 2.0** (Gao et al., 2026) serializes all agents into one time-varying prompt stream, including explicit continuous robot values.
- **MultiWorld** (Wu et al., 2026) retains its Multi-Agent Condition Module, while the Global State Encoder receives only the same single initial view available to other methods.
- **Incantation** (Zhu et al., 2026b) uses per-frame, per-entity language actions, with robot values represented explicitly in text.
- **ActionParty** (Pondaven et al., 2026) jointly predicts persistent subject-state tokens and video tokens. It receives initial agent locations as required by its binding mechanism; we extend its discrete action projection to continuous robot values and train the resulting model.
- **WanToFight-style** (Hu et al., 2026) combines per-agent locally causal action injection with first-frame identity crops and reference-conditioned Player Association. The original code is not publicly available; our reported results are obtained by training this reproduction.
- **LiveAct** uses per-agent schema-numerical controls without locations, masks, boxes, or identity crops.

E.3 Metrics

Reconstruction quality. FVD (Unterthiner et al., 2018), LPIPS (Zhang et al., 2018), PSNR, and SSIM follow the sampling protocol of App. E.1: FVD measures distributional similarity using I3D features, whereas the three frame-paired metrics measure perceptual, pixel, and structural agreement with the ground-truth future.

Motion and appearance diagnostics. Motion Magnitude Error (MME) is $|\rho(\mathrm{gen}) - \rho(\mathrm{GT})|$, where $\rho(\cdot)$ is the mean RAFT (Teed & Deng, 2020) flow magnitude over the same 16-frame 256^2 grid; lower values indicate closer aggregate motion energy. Aesthetic uses the LAION improved predictor with CLIP ViT-L/14.

Control and attribution. Following MultiWorld (Wu et al., 2026), the inverse dynamics evaluator of the main paper is trained only on real training clips and never on generated videos; it consists of one frozen convolutional network with about 2.9M parameters per data source. For each action frame, the network reads the surrounding RGB frames resized to 128^2, using a five-frame window with offsets $\{-6, -3, 0, 3, 6\}$ for games and a three-frame window $\{-1, 0, 1\}$ for robots, and jointly predicts the controls of both agents. It receives no agent identity or location, so it can tell the agents apart only from the pixels. Predictions cover frames 1–80 of each clip, excluding the conditioning frame. For games, it predicts movement and interaction events for each agent, and Game Act-F1 is the macro F1 across valid events and agents. For robots, it predicts the continuous control trajectory. Let $\mathcal{V}$ denote the valid, non-padding control dimensions, which are the only ones scored, and let σ_j be the standard deviation of dimension j on the real training set. We compute

$$\mathrm{NRMSE} = \sqrt{\frac{1}{T|\mathcal{V}|}\sum_{\tau=1}^{T}\sum_{j\in\mathcal{V}}\left(\frac{\widehat{a}_{\tau,j} - a_{\tau,j}}{\sigma_j + \varepsilon}\right)^2}, \tag{17}$$

where $\widehat{a}_{\tau,j}$ is the recovered control, ε is a small constant, and lower values indicate more accurate control recovery.

Attr-Acc is a pairwise swap test on the same generated videos and evaluator predictions, so it requires no additional renders and is computed identically for every method in Tab. 2. For a clip with N_{ag} agents, let $\mathbf{u}_\tau^{(i)}$ and $\widehat{\mathbf{u}}_\tau^{(i)}$ be the commanded and recovered controls of agent i on $\mathcal{V}$, centered by subtracting the mean of the same source, agent, and dimension on the evaluator's training set. Robot dimensions are further divided by the corresponding training-set standard deviation, whereas game dimensions already lie in $[0,1]$ and are not rescaled. At frame τ, an unordered agent pair (i,i') is scored only if their commanded streams differ on at least one dimension in $\mathcal{V}$, since exchanging identical commands is uninformative. We compare the commanded assignment with the one in which only agents i and i' are exchanged,

$$E^{\mathrm{c}}_{\tau,ii'} = \sum_{k\in\{i,i'\}} \left\|\widehat{\mathbf{u}}_\tau^{(k)} - \mathbf{u}_\tau^{(k)}\right\|_2^2, \qquad E^{\mathrm{s}}_{\tau,ii'} = \left\|\widehat{\mathbf{u}}_\tau^{(i')} - \mathbf{u}_\tau^{(i)}\right\|_2^2 + \left\|\widehat{\mathbf{u}}_\tau^{(i)} - \mathbf{u}_\tau^{(i')}\right\|_2^2, \tag{18}$$

where the remaining agents contribute equally to both assignments and are omitted. The pair scores 1 if $E^{\mathrm{c}}_{\tau,ii'} < E^{\mathrm{s}}_{\tau,ii'}$, 0.5 on a tie, and 0 otherwise. It is therefore correct when the controls recovered from the pixels match the commands better in the commanded order than with the two agents exchanged, that is, when each of the two streams drives its intended agent rather than the other one. A frame is scored if it has at least one scored pair, and its score is the mean over these pairs; with $N_{\mathrm{ag}}{=}2$ there is a single pair and the test reduces to exchanging the two agents. The centering fixes the chance level at 0.5 for any N_{ag}: a predictor that ignores the video and outputs each agent's training mean is zero for every agent after centering, so $E^{\mathrm{c}}_{\tau,ii'} = E^{\mathrm{s}}_{\tau,ii'}$ for every pair. Frame scores are averaged within each clip, then over the clips of each source, and finally over sources without weighting, four for games and two for robots. Averaging within clips first prevents clips with many scored frames from dominating, and the source-level mean gives every source equal weight.

F Additional Results

F.1 Three-Stage Causal Transfer

Tab. 3 summarizes the generation settings of the three stages together with their control, attribution, and throughput results. Stage 2 runs slower than Stage 1 despite its causal attention: it repeats 50 denoising steps with classifier-free guidance (CFG) sequentially for each of the five four-latent chunks and refreshes the KV cache after every chunk, whereas the bidirectional teacher denoises all 21 latents jointly at each step. These many small, sequential forward passes use the GPU less efficiently than the teacher's large joint passes. Stage 3 reduces each chunk to four denoising steps without CFG and reaches 17.70 FPS, $10.7\times$ faster than Stage 2.

Table 3: **Three-stage causal transfer.** Generation settings and results of each stage; throughput is measured at 512×448 on a single NVIDIA A800, and Stage 3 samples without classifier-free guidance (CFG). Best results are bolded.

	Stage 1 **Foundational Ability Acquisition**	**Stage 2** **Teacher Forcing for AR Tuning**	**Stage 3** **DMD for Streaming Video**
Role	Full-context teacher	Causal initialization	Rollout distillation
Attention	Bidirectional	Blockwise causal	Blockwise causal
Training history	Full video	Clean chunks	Model rollouts
Streaming	×	✓	✓
Chunk size	Full sequence	4 latent frames	4 latent frames
Denoising steps	50	50	4
CFG scale	5.0	5.0	–
Game Act-F1 ↑	0.539	0.565	**0.594**
Game Attr-Acc ↑	0.844	**0.898**	**0.898**
Robot NRMSE ↓	**0.430**	0.473	0.558
Robot Attr-Acc ↑	0.969	**0.970**	0.942
FPS ↑	2.77	1.66	**17.70**

F.2 Detailed Ablations

Tab. 5 extends the main-text ablation along two axes: the construction of the schema-guided control tokens and the temporal alignment of App. B.2.

Table 4: **Design of the Schema-Guided Action Adaptation variants.** Each variant changes only how the control tokens of agent i are constructed; the superscript (i) is omitted. All variants keep Agent-Aware Spatial Modulation, per-agent cross-attention, the one-latent action window, and the time and agent embeddings of Eq. 5. u_j is the projected schema embedding of Eq. 3, $(\gamma_{\tau,j}, \beta_{\tau,j}) = \mathrm{MLP}_{\mathrm{mod}}(a_{\tau,j})$ as in Eq. 4, e_j^{ID} is a learned per-dimension embedding, and π is a fixed permutation of the control dimensions.

Variant	Control identity	Tokens per frame	Value fusion	Training	Control token
Full LiveAct	schema u_j	C	affine modulation	trained	$h_{\tau,j} = u_j \odot (1+\gamma_{\tau,j}) + \beta_{\tau,j}$
w/o SGAA	–	1	flat MLP	trained	$h_\tau = \mathrm{MLP}_{\mathrm{flat}}(A_\tau)$
w/o value modulation	schema u_j	C	addition	trained	$h_{\tau,j} = u_j + \mathrm{MLP}_{\mathrm{val}}(a_{\tau,j})$
w/o schema semantics	learned ID e_j^{ID}	C	affine modulation	trained	$h_{\tau,j} = e_j^{\mathrm{ID}} \odot (1+\gamma_{\tau,j}) + \beta_{\tau,j}$
shuffled schema	schema $u_{\pi(j)}$	C	affine modulation	inference only	$h_{\tau,j} = u_{\pi(j)} \odot (1+\gamma_{\tau,j}) + \beta_{\tau,j}$

Schema-Guided Action Adaptation. SGAA combines three factors: a schema description that identifies each control, a separate token for each control dimension, and affine modulation of that token by the control value. Tab. 4 summarizes the variants that separate these factors. Every variant changes only how the control tokens are constructed and keeps Agent-Aware Spatial Modulation, per-agent cross-attention, the one-latent action window, and the time and agent embeddings of Eq. 5.

The main-text w/o SGAA variant replaces the module with a conventional flat action encoder that maps the full action vector of each frame to a single token. It removes all three factors at once and reduces the action memory of each latent from rC to r tokens. The remaining variants keep the per-dimension tokens and remove one factor at a time. The w/o value modulation variant keeps the schema token but adds a value embedding to it instead of modulating it. The w/o schema semantics variant keeps the modulation but replaces the schema embedding with a learned per-dimension embedding, so that each dimension remains distinguishable but carries no language description; comparing it with the full model tests whether the schema description contributes beyond identifying the dimension.

Replacing modulation with addition alone collapses control to the level of removing SGAA in both domains (game Act-F1 0.209 vs. 0.225, robot NRMSE 1.071 vs. 1.120), and attribution is no better (game Attr-Acc 0.513 vs. 0.603, robot Attr-Acc 0.480 vs. 0.484), with robot Attr-Acc near the chance level of 0.5. Schema descriptions and per-dimension tokens are therefore insufficient without modulation. This is consistent with the design of Sec. 3.3: addition mixes the value embedding into the schema token, whereas modulation keeps the schema token as the carrier of the control identity and lets the value only rescale and shift it.

Replacing the schema embedding with learned per-dimension IDs isolates the contribution of the language description. On games, this variant falls to the level of removing SGAA (Act-F1 0.230 vs. 0.225, Attr-Acc 0.588 vs. 0.603), even though it keeps the per-dimension tokens and value modulation of the full model. The four game sources assign different controls to the same index, so an index-level ID cannot distinguish these controls, whereas the schema description can. On robots, learned IDs retain part of the gain over removing SGAA (NRMSE 0.799 vs. 1.120, Attr-Acc 0.778 vs. 0.484) but remain clearly below the full model (0.558 and 0.942). The schema description therefore contributes beyond identifying each dimension, most strongly on the games, where the same index carries different meanings across sources.

We additionally test whether the trained full model actually reads the schema descriptions. Without retraining, we pair each control dimension with the description of another dimension at inference, keeping the action values, per-dimension tokens, and value modulation unchanged. Control collapses to the level of the reduced variants: game Act-F1 drops to 0.207, and Attr-Acc falls to 0.504 on games and 0.496 on robots, both near chance. The full model therefore relies on the schema embedding to identify each control, and the per-dimension tokens and value modulation cannot produce the commanded behavior once the descriptions are mismatched.

Temporal alignment. The action window length L sets how many latent frames of controls each future latent attends to: the main model uses $L{=}1$, so every latent reads exactly the $r{=}4$ raw-time steps compressed into it, while $L{=}2$ and $L{=}4$ extend the window over the following latents. Longer windows dilute the per-latent control signal and degrade all four metrics monotonically, on games

(Act-F1 $0.594 \rightarrow 0.449 \rightarrow 0.325$, Attr-Acc $0.898 \rightarrow 0.846 \rightarrow 0.721$) as well as on robots (NRMSE $0.558 \rightarrow 0.591 \rightarrow 0.700$, Attr-Acc $0.942 \rightarrow 0.937 \rightarrow 0.926$).

Table 5: **Detailed ablations of Schema-Guided Action Adaptation and temporal alignment.** The SGAA variants are defined in Tab. 4. All trained variants use the same data and optimization budget as the full model. L is the action window length in latent frames (App. B.2). Best results are bolded.

Variant	Game Act-F1 ↑	Game Attr-Acc ↑	Robot NRMSE ↓	Robot Attr-Acc ↑
Full LiveAct (L=1)	**0.594**	**0.898**	**0.558**	**0.942**
Schema-Guided Action Adaptation				
w/o SGAA (flat MLP)	0.225	0.603	1.120	0.484
w/o value modulation (additive value)	0.209	0.513	1.071	0.480
w/o schema semantics (ID tokens)	0.230	0.588	0.799	0.778
shuffled schema (inference only)	0.207	0.504	1.063	0.496
Temporal alignment				
action window L=2	0.449	0.846	0.591	0.937
action window L=4	0.325	0.721	0.700	0.926

F.3 Spatial Localization of Agent-Specific Residuals

Fig. 2 provides a complementary spatial diagnostic of Agent-Aware Spatial Modulation: each agent-specific residual-magnitude heatmap concentrates around the corresponding visual agent, consistent with the intended stream-to-agent binding.

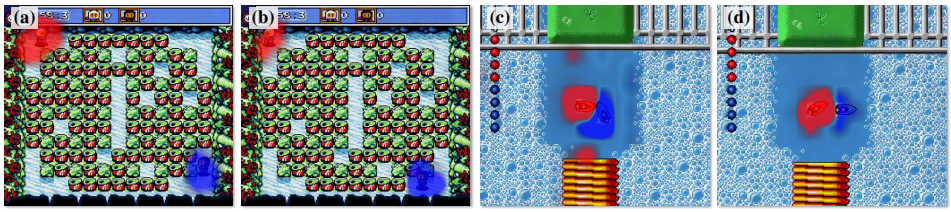

Figure 2: **Spatial localization of agent-specific control residuals.** Panels (a)–(d) show representative spatial residual-magnitude heatmaps from the agent-specific branches. Red and blue denote agents 1 and 2, respectively, and higher intensity indicates a larger response. Across the examples, the high-response regions concentrate around the corresponding visual agents.

F.4 Additional Qualitative Comparisons

Fig. 3 and Fig. 4 add two pairs of cases with the same methods and sampling protocol as Fig. 5: a Mega Bomberman round and a Street Fighter II round, each paired with a RoboTwin scene in which one arm places an object into a basket. They confirm that the failure modes of the main paper are not specific to the two scenes shown there. First, the scene drift of LingBot-World 2.0 can be abrupt: In Fig. 3, the generated frames gradually turn black and then transitions to an unrelated scene. However, in both robot cases, it started to break down from the first sampling frame. Second, the game-track failures of Incantation grow with time, from a darkened board to black frames, while MultiWorld and ActionParty keep the board and the table but smear the players and arms. Although WanToFight maintains the scene and completes the action, there is a mismatch between the arm and the action. LiveAct matches the target arm pose more closely at each sampled frame.

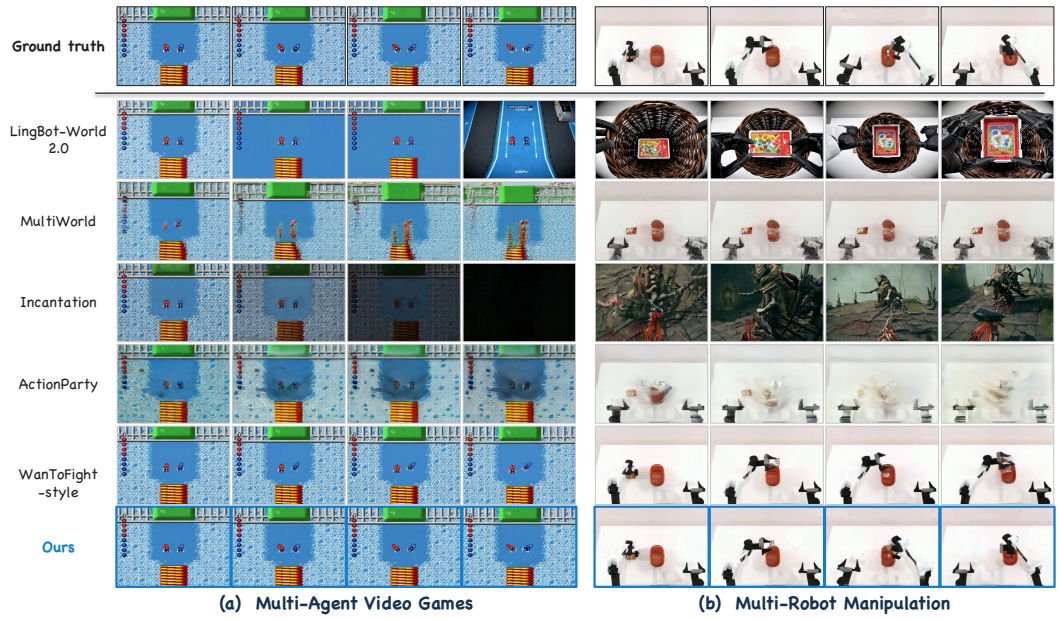

Figure 3: **Additional qualitative comparison (set 1).** Same layout and methods as Fig. 5.

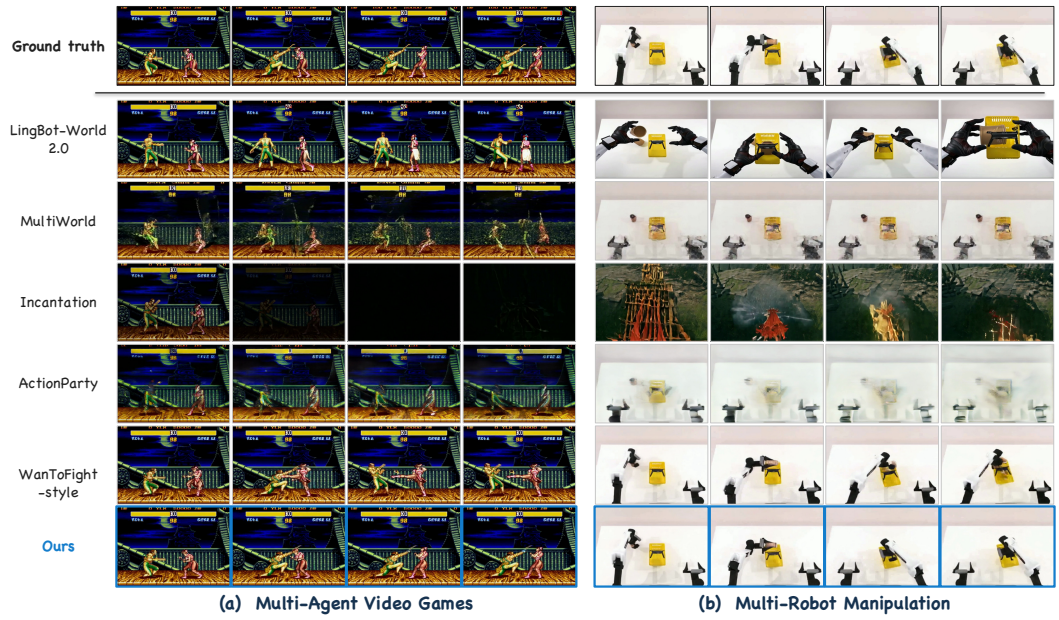

Figure 4: **Additional qualitative comparison (set 2).** Same layout and methods as Fig. 5.

F.5 Additional LiveAct Showcases

Fig. 5 presents additional generations from LiveAct across the game and robot domains. Each row shows one evaluation condition, and columns show the generated rollout at successive times.

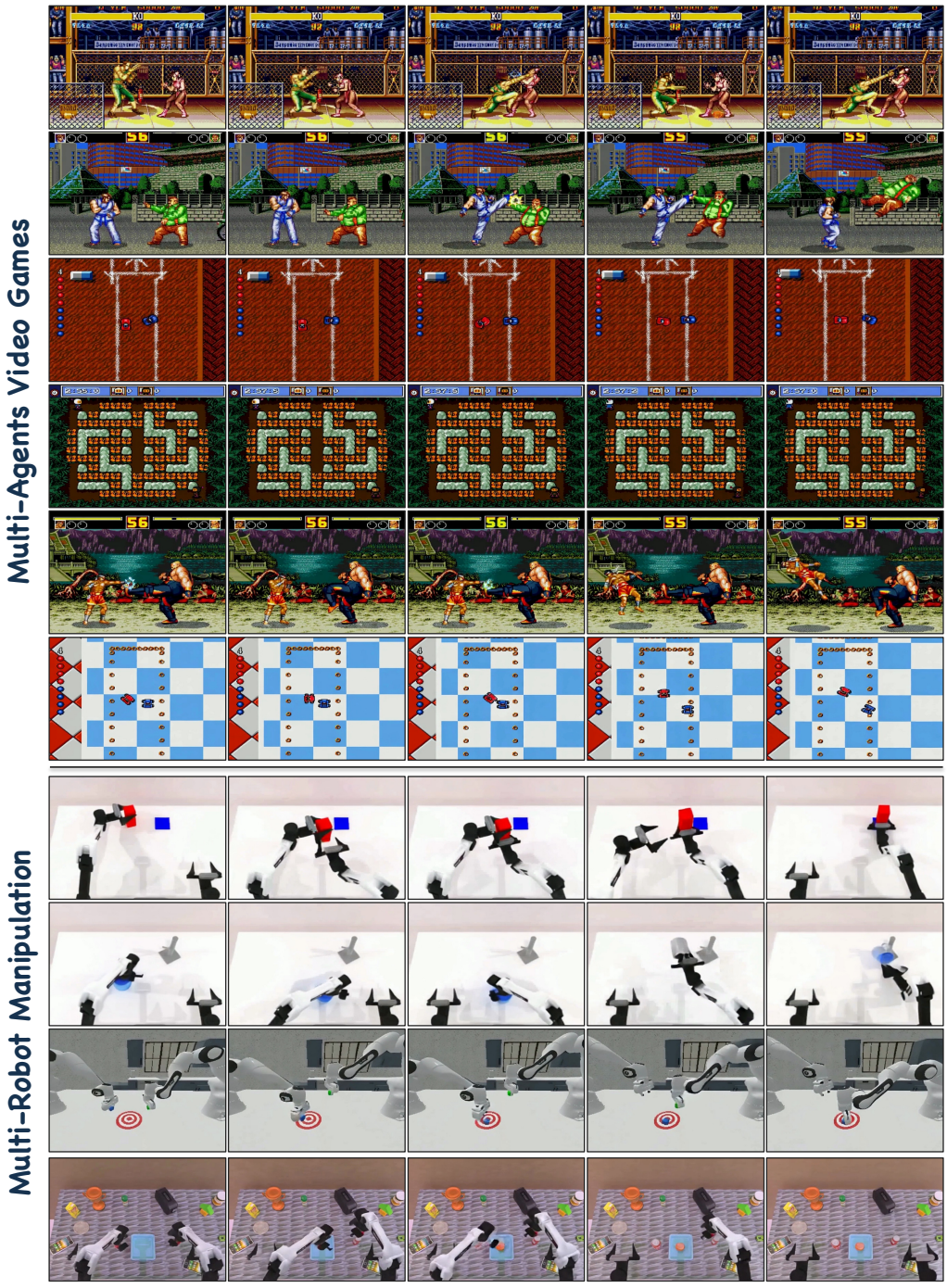

Figure 5: **Additional LiveAct showcase.**

The interface is not tied to a fixed number of agents: the dataset contains scenes with two to four concurrently controlled agents, and each additional agent only adds one agent description, one action stream, and one per-agent cross-attention residual, without any architectural change. Fig. 6 shows scenes with more than two agents: a Micro Machines race and a Mega Bomberman round with binary controls, and a RoboFactory cube-stacking episode with continuous controls.

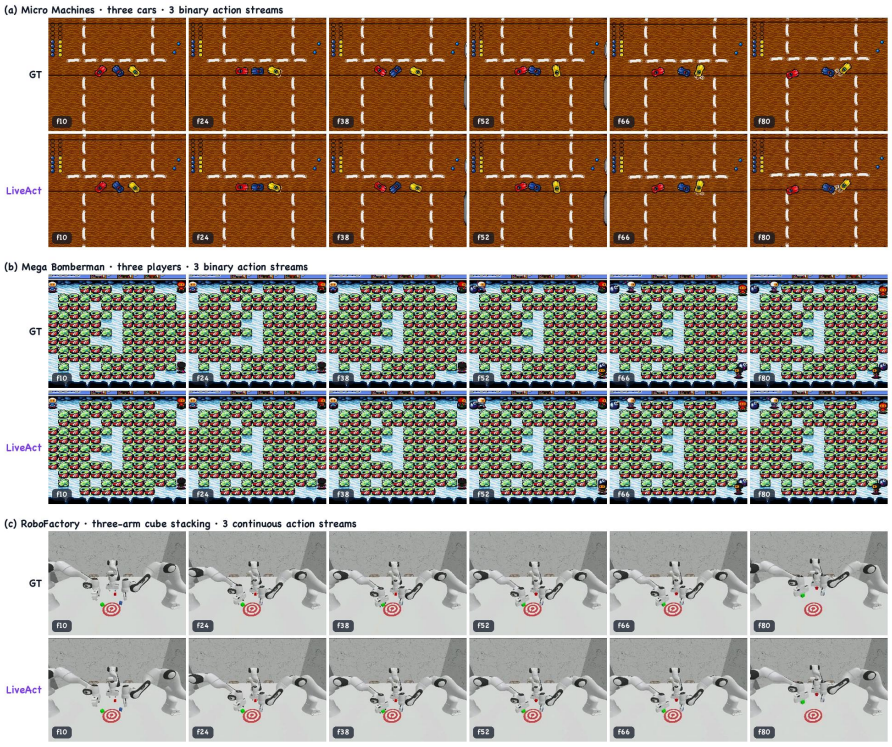

Figure 6: **Showcase beyond two agents.** Scenes with more than two concurrently controlled agents. For each scene, the top row shows the ground truth and the bottom row the LiveAct generation from the same initial frame and per-agent action streams.

F.6 Long-Horizon Rollout Showcase

Fig. 7 shows an additional dynamic case that exceeds the 81-frame training and evaluation window used for quantitative assessment.

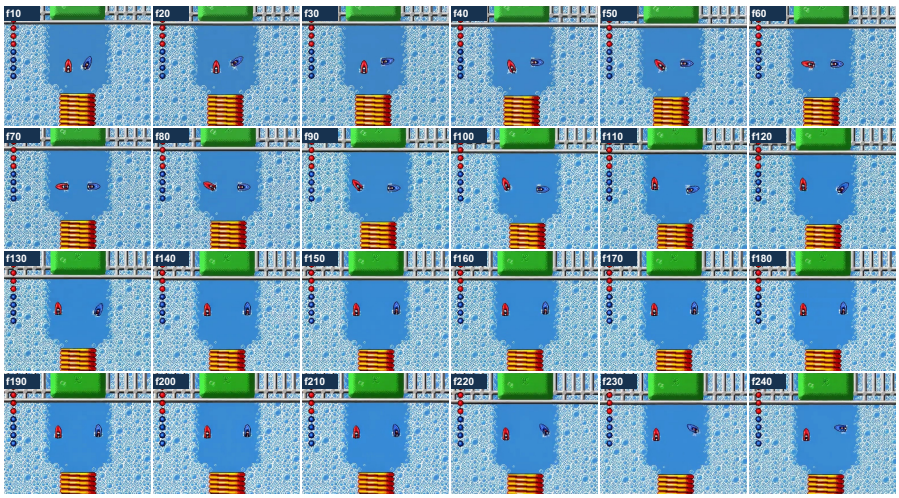

Figure 7: **Single-case long-horizon rollout as a dense frame matrix.** We show 24 generated frames from the same 241-frame rollout, sampled every ten frames and ordered from left to right and top to bottom.