

InWorld: From Imagination to Living Worlds with Agents and Fast Video Generation

Murea Labs

Abstract

Interactive world models aim to simulate environments that evolve under user actions while retaining the consequences of past interactions. Extending such worlds to sight and sound requires persistent entities, changing scene states, and local audiovisual dynamics to remain coherent across turns. This paper presents InWorld, an interactive audiovisual world model that couples explicit world-state management with neural audiovisual generation. Its world agent, Murea-Mind, maintains characters, objects, locations, and narrative progress, interprets user actions in context, and compiles planned events into generation conditions. Observations of generated frames inform subsequent planning and selected state updates, closing the interaction loop. Its audiovisual generator, Murea, realizes these events through Omni-Reference Generation, using visual references to anchor entity appearance and structured descriptions to specify actions and dialogue. Streaming generation uses modality-specific histories, with longer audio context for unfinished utterances. Degraded-history training prepares the model for imperfect rollout context, while sequential MeanFlow and distribution matching distillation enable few-step synthesis. Branching histories support alternative continuations, and co-op interaction brings multiple participants' actions into the same world. We evaluate the generation components in complementary configurations. Murea-Pro achieves an Overall score of 81.13 on WBench Full-289, the highest among the compared systems. Murea-Fast achieves a Total score of 64.98 on OpenS2V-Eval and an aggregate score of 69.9 on OS2V-Pro, an extended evaluation of reference fidelity, attribute binding, rendering quality, and audio. Murea-Pro and Murea-Fast achieve state-of-the-art performance on their respective benchmarks, and both models surpass most evaluated commercial systems in their respective evaluations.

Project page: <https://murea.ai>



Murea Labs



Figure 1. Conceptual overview of InWorld. User inputs are translated into actions that extend an ongoing story across different worlds (top), while co-op mode combines multiple players' actions within a shared world (bottom).

1 Introduction

Interactive world models offer a way to experience environments whose evolution can be shaped through actions. A user can explore a place, interact with its inhabitants, change the surroundings, and encounter the consequences in later observations. This requires continuity at the level of the world: characters remain identifiable, acquired objects affect subsequent events, and established places can be revisited. An audiovisual world further expresses these changes through motion, dialogue, and environmental sound. If a character opens a door while speaking, the resulting movement, sound, and ongoing utterance should belong to the same event and provide context for what happens next.

Recent world models advance this goal through complementary approaches. Persistent world-state registers and external spatial memory carry information across generated observations [29, 47]. Agent-guided interaction and language-based action interfaces extend the ways users can influence a world [9, 53]. Joint audiovisual models provide a foundation for synchronized sensory generation [11, 40], and Starchild-1 brings audiovisual rollout and online orchestration into an interactive world-model setting [38]. These developments make the connection between persistent state, user intervention, and evolving audiovisual observations central to world-model design.

Maintaining this connection involves several distinct demands. An entity’s appearance should persist even as its location, possessions, and role in the story change. Recent observations contain the motion and acoustic context needed to continue an event, but a finite window may omit an earlier interaction that matters now. Sound adds its own temporal dependencies: a visually smooth transition can still interrupt a sentence or repeat words. Moreover, the state change requested by a user and planned by an agent may only be partially realized in the generated scene. A world model must therefore connect remembered state and proposed actions with observations of what its neural generator actually depicts.

This paper presents InWorld, an interactive audiovisual world model built around an explicit state–action–observation loop. InWorld separates persistent world management from local audiovisual synthesis while coupling them through a shared generation interface. Its world agent, Murea-Mind, maintains the persistent state of the environment, while Murea renders the audiovisual consequences of the agent’s decisions.

At a high level, InWorld operates through two coupled loops. In the outer loop, Murea-Mind maintains registered characters, objects, and locations, together with their dynamic attributes, narrative objectives, connected places, and branching segment history. Given the current world state, a user instruction, and the latest visual observation, the agent produces a directive describing the next event or sequence of events. The directive specifies the participating entities, their actions, dialogue, proposed state changes, and the way the new segment connects to its parent. A deterministic compiler then resolves every entity to its registered visual or voice reference, expands the directive into the structured generation format, and selects either latent continuation or a new shot.

In the inner loop, Murea uses the compiled conditions to jointly synthesize the next video–audio segment. Visual references anchor the appearance of persistent entities, optional voice references provide speaker characteristics, and latent history supplies local temporal context for motion, layout, and ongoing speech. After rendering the segment, its frames become the next observation. An observation model compares the visible outcome with the planned event and writes selected facts back into the world state. This closes the loop between what the agent intended to happen and what the generator actually depicted. Persistent symbolic state therefore provides long-range memory, while audiovisual latents provide short-range temporal context.

Persistent entities connect the world representation to its sensory realization. Related work studies entity-level reference memory and cross-shot audiovisual continuity [12, 19]. InWorld uses registered visual references to anchor recurring entities, while dynamic attributes describe their current condition. Murea-Mind’s deterministic compiler binds each entity to the reference slots of the current request and expresses events using the structured caption convention employed during training. Murea’s *Omni-Reference Generation* (Section 3.1) independently encodes reference images as clean conditioning tokens, distinguishes them with source identifiers, and jointly synthesizes target video and audio. The generation interface also provides an optional audio-reference path. Continuation supplies parent latents for local temporal context, while a cut retrieves relevant references and starts a new shot.

Our few-step distillation recipe (Section 3.2) distills the bidirectional audiovisual generator into a few-step sampler: teacher-supervised MeanFlow learns finite-interval transport, and Distribution Matching Distillation (DMD) then refines the resulting sample distribution [10, 45, 46]. Efficient inference combines optimized attention, multi-GPU parallelism, caching, and spatial super-resolution to reduce rendering cost (Section 4).

The streaming path (Section 3.3) extends audiovisual generation over successive intervals with bounded modality-specific history. Video uses a short preceding context, while audio retains a longer window to preserve unfinished utterances and prosodic continuity. Degraded-history training prepares the model for the imperfect context encountered during rollout, following the long-horizon training direction of Helios and Alaya-EVOKE [47, 50]. The streaming model uses an analogous MeanFlow-to-DMD distillation procedure with degraded history included in the conditioning context.

InWorld also supports branching exploration and shared participation. Its segment tree lets users return to an earlier decision, restore branch-specific dynamic state and quest progress, and explore another continuation using the registered entities. In co-op mode, participants’ actions are bound to their characters and compiled into a single shared shot. Keyframe observations record the visible outcome for subsequent rounds, keeping event forecasts distinct from confirmed history. These mechanisms extend the same interaction loop to alternative world trajectories and coordinated actions by multiple participants.

Our contributions are:

- **InWorld**: an interactive audiovisual world-model system that connects persistent world state, language-based actions, neural generation, and visual feedback, while supporting branching and co-op interaction.
- **Murea-Mind**: a persistent world agent with explicit asset, dynamic-state, narrative, place-graph, and segment-tree representations, together with deterministic planning, condition compilation, observation-based updates, and branch-aware co-op control.
- **Murea-Fast/Pro**: a reference-conditioned audiovisual generator with Omni-Reference Generation, a MeanFlow-to-DMD recipe for few-step bidirectional generation, efficient inference, and a separate bounded-history streaming path with degraded-history training and analogous distillation. Murea-Pro is the flagship model for multi-turn interaction, while Murea-Fast is a lightweight, fast model built on LTX-2.5.

We evaluate the generation components in complementary configurations. Murea-Pro obtains an Overall score of 81.13 on WBench Full-289 [48], covering 289 cases and 1,058 interaction turns, the highest among the compared models. Murea-Fast, built on LTX-2.5 [22], obtains a Total score of 64.98 on OpenS2V-Eval [49] and an aggregate score of 69.9 on OS2V-Pro, an extended evaluation of the same cases covering reference fidelity, attribute binding, rendering quality, and audio. These experiments characterize multi-turn interaction and reference-conditioned synthesis, respectively; the complete agent-level state–action–observation loop is specified by the InWorld system design.

2 World Agent

World Agent guides audiovisual generation through an open-ended series of user interactions. To generate each new segment, the agent must account for context that spans multiple turns: which characters exist, where they are, how the previous segment ended, and which parts of a longer request remain unfinished. The agent maintains these facts in a persistent world state, uses them to plan and condition each new segment, and records the generated result for subsequent turns.

This design is based on two characteristics of the setting. The generator does not retain memory between generation calls, so the agent provides text, references, and latent context to maintain continuity across turns. There is also no simulator behind the world: a request describes an intended change, and whether the change happened can only be seen in the rendered frames. Related systems resolve these problems differently. WorldWeaver stores shared world information inside the generator as registers that are updated after each generated chunk [29], while tool-constrained authoring agents build scenes through operations checked against a simulator state [5]. World Agent keeps an explicit, inspectable state outside the generator and uses natural language as its control interface, which has been argued to be expressive enough to control several entities at once [53].

Based on these considerations, our design follows four principles. Registered assets use reference images and voices to maintain consistent identities across segments. A language model plans events, and a deterministic compiler translates those plans into generator conditions. A segment tree stores alternative branches and supports latent continuation, undo, and regeneration. Finally, observations of each segment’s final frame feed visible outcomes back into the world state.



Figure 2. Overview of World Agent. (a) The persistent world state $\mathcal{W}$: the asset registry $\mathcal{E}$ with reference images and voices, the dynamic state $\mathcal{S}$, the narrative context $\mathcal{N}$ with the current objective, the place graph $\mathcal{G}$, and the segment tree $\mathcal{T}$. (b) One turn. The planner π reads a snapshot of the state and the previous final frame and returns, in a single planner call, the directive D_t and the six-section prompt; the compiler $\mathcal{C}$ resolves the event sequence, motion plan, and state updates, binds every name to a registered asset, and locks the prompt; the video backend renders the segment by continuation or cut; the result is stored as a node with its directive, prompt, reference image order, frames, state before and after, and continuation link. (c) During playback, the observation model ρ compares the final frame with the planned event and writes back what was observed, the next-turn suggestions, the objective, and which assets appeared, were picked up, or were dropped.

Let $\mathcal{W}_{t-1}$ be the world state before turn t , u_t the instruction for turn t , and X_{t-1} the previous segment with final frame F_{t-1} . One turn is

$$\begin{aligned} D_t &= \pi(u_t, \Sigma(\mathcal{W}_{t-1}), F_{t-1}), & c_t &= \mathcal{C}(D_t, \mathcal{W}_{t-1}), \\ X_t &\sim p_\theta(X | c_t, \tilde{Z}_t), & \mathcal{W}_t &= \mathcal{U}(\mathcal{W}_{t-1}, D_t, X_t), \end{aligned} \quad (1)$$

where Σ takes a snapshot of the state, the planner π produces a directive D_t , the compiler $\mathcal{C}$ maps it to the generator condition c_t of Equation (12), $\tilde{Z}_t$ is optional latent context from the parent segment, and $\mathcal{U}$ commits the result. Figure 2 illustrates the loop. Sections 2.1–2.4 describe the state and these operators for a single player, and Section 2.5 extends them to a shared world in which several players act in the same shot.

2.1 World State

As shown in Figure 2(a), the world state collects all information that must persist across turns:

$$\mathcal{W}_t = (\mathcal{E}_t, \mathcal{S}_t, \mathcal{N}_t, \mathcal{G}_t, \mathcal{T}_t). \quad (2)$$

The five components respectively store asset identity and appearance, evolving attributes, narrative context, spatial layout, and the branching history of generated segments.

The asset registry $\mathcal{E}_t$ contains all registered characters, objects, and locations. Each asset $e = (k_e, I_e, A_e, d_e)$ has a kind k_e , a reference image I_e , an optional voice sample A_e , and a locked description d_e that is defined once and reused verbatim in later prompts. A shared visual style is applied to every generated reference image and prompt. The registry provides appearance consistency across revisits: when the story returns to a known location, the agent retrieves the same reference image instead of asking a model to describe the location again. Before their first appearance, new characters and locations are registered and assigned generated reference images. The dynamic state $\mathcal{S}_t$ stores evolving attributes. For each character, these include location, outfit, visible state, mood, and goals; for each location, they include time of day, weather, lighting, and props.

The narrative context $\mathcal{N}_t$ records the main storyline, its phase and milestones, side quests, the current objective, and turn history. Each history entry stores the instruction used for the turn, a short narrative response, and a one-line event summary. Generation prompts are excluded so that the history remains focused on story progression and does not copy earlier rendering details into later planning. The place graph $\mathcal{G}_t$ records connections between locations, allowing the planner to route movement along known paths. The segment tree $\mathcal{T}_t$ stores each generated segment as a node with its parent, directive, prompt, reference image order, output frames, continuation link to the parent latent context, and snapshots of dynamic state and quest progress before and after generation. The head pointer h_t identifies the current endpoint of the story.

2.2 Turn Planning

The planner π is a multimodal language model. Its input is the snapshot $\Sigma(\mathcal{W}_{t-1})$, the current instruction, a duration budget $\bar{d}$ for one segment, and the final frame F_{t-1} . The symbolic and visual contexts complement each other. References and locked descriptions define which entities exist and their canonical appearance, while the final frame records how the previous segment actually ended, including layout and poses that the symbolic state does not capture.

The directive D_t specifies the location ℓ_t , the participating assets, the event to depict, any dialogue and its speakers, how the segment connects to its parent, proposed state changes $\Delta\mathcal{S}_t$, and next-turn suggestions. An instruction such as “go to the market and talk to the guard” contains more events than a single segment can depict. The planner therefore decomposes the instruction into an ordered list of events $(\beta_1, \dots, \beta_K)$ and groups them into consecutive segments $B_1, \dots, B_J$, with each segment satisfying the duration budget,

$$\sum_{\beta \in B_j} \hat{d}(\beta) \leq \bar{d}, \quad j = 1, \dots, J, \quad (3)$$

where $\hat{d}(\beta)$ is the estimated duration of an event. Only B_1 is rendered in the current turn. The remaining events are rewritten as a complete instruction and automatically provided as input to the next turn. A multi-step user request can therefore span several turns, each producing one segment, while the user can interrupt at any time. In the example, the current segment shows the walk to the market, and the following segment contains the conversation with the guard.

When the planned event involves a physical interaction, the directive also includes a motion plan. It specifies the actors and target, their initial relation, the path, the contact or relation change, the end state, and the framing needed to show that change. The motion plan is checked for internal consistency before compilation, including whether the interaction type is consistent with its phase and resulting relation, and whether the framing can show the relation. Most violations are repaired locally by deterministic rules. If an inconsistency remains, the system repairs only the motion plan; errors in event decomposition trigger replanning.

Planning precedes rendering, so its latency contributes directly to user wait time. The planner can return the directive and the six-section generation prompt in a single planner call, placing the fields required for rendering first. Presentation fields such as next-turn suggestions are completed afterwards, while the generator inputs are being prepared. When suggestions are displayed, the planner prepares a directive for each suggestion in the background. Selecting one then reuses its prepared directive and avoids an additional planning call.

2.3 Conditioning and Continuity

The compiler $\mathcal{C}$ maps a directive to the generator condition $c_t = (y_t, \mathcal{R}_{v,t}, \mathcal{R}_{a,t})$, which contains prompt text, ordered visual references, and voice references. It aligns each entity name, reference image, and voice with the corresponding registered asset.

Let $e_{(1)}, \dots, e_{(n_t)}$ be the assets selected for the segment, ordered with explicitly selected characters first, then the remaining characters and focal objects, and the location last. The compiler builds

$$\mathcal{R}_{v,t} = (I_{e_{(1)}}, \dots, I_{e_{(n_t)}}), \quad \iota_t(e_{(i)}) = i, \quad \mathcal{R}_{a,t} = (A_e)_{e \in \text{spk}(Q_t)}, \quad (4)$$

where n_t is the number of selected reference images and satisfies $n_t \leq N_{\max}$, $\iota_t(e)$ is the reference-image index assigned to asset e in the current generation call, Q_t is the dialogue of the segment, and $\text{spk}(Q_t)$ is the set of characters resolved as its speakers. Reference-image indices are assigned separately for each generation call, while asset identities persist across calls. Subject definitions are written from the locked descriptions d_e using

the index assigned to each asset. Voice bindings follow speaker identity rather than reference-image order, so a location or object cannot be assigned a voice solely because of its position in the reference list.

Each backend receives prompts in the format used during training. For structured audiovisual prompts, this format consists of separate sections for subject definitions, a summary, reference-retention statements, the shot description, the soundscape, and music. Expanding a short instruction into a detailed description follows the prompt-rewriting approach of Movie Gen [35]. Every name in the expanded prompt is tied to a registered asset.

The compiler handles content that requires no model-based judgment, including subject definitions, reference-retention statements, speaker and voice bindings, visual style, and the soundscape of a location. A location’s soundscape is fixed on the first visit and reused afterwards.

Continuation uses the parent segment’s tail latents, $\tilde{Z}_t = Z_{h_{t-1}}$, to provide context for pose, motion, camera, and layout across the segment boundary. A cut sets $\tilde{Z}_t = \emptyset$ and uses the references without the parent’s latent context. The mode is selected according to spatial and temporal continuity:

$$\text{mode}_t = \begin{cases} \text{continue,} & \text{if } \ell_t = \ell_{t-1}, \\ \text{continue,} & \text{if a continuous transition reaches a location } \ell_t \in \text{adj}_G(\ell_{t-1}) \cup \mathcal{L}_t^{\text{new}}, \\ \text{cut,} & \text{otherwise,} \end{cases} \quad (5)$$

Here, adj_G denotes neighbouring locations in the place graph, and $\mathcal{L}_t^{\text{new}}$ contains locations introduced in the current turn. Continuation is used when the story remains in one space or moves continuously into the next; jumping to a distant known location begins a new shot from its reference image. When the backend supports both, references provide persistent identity, while latent context provides short-term temporal continuity. The directive also records each character’s carried items alongside the dynamic state, because locked descriptions do not reflect items acquired during the story.

2.4 Feedback and Branching

To guide the generation of visible changes, such as a new outfit or an item in hand, the agent incorporates the planner’s proposed state changes before rendering. During directive compilation, these updates are merged into the dynamic state through a projection that retains only updates to registered entities and allowed fields:

$$\mathcal{S}_t = \mathcal{S}_{t-1} \oplus \Pi_{\mathcal{F}}(\Delta\mathcal{S}_t), \quad (6)$$

where $\oplus$ merges updates into the dynamic state, and $\Pi_{\mathcal{F}}$ removes updates to unknown entities or fields not allowed by the schema $\mathcal{F}(k)$ for entity kind k . Visible attributes are included in the prompt, while goals, knowledge, and relations are used only in later planning.

Other updates depend on the rendered output. Once a segment is ready, the observation model ρ examines its final frame, the planned event in D_t , and the current characters and inventory while the segment plays. This model corresponds to the adjudicator in Figure 2(c). It revises next-turn suggestions and the objective, and updates the registry to reflect acquired or dropped items and newly introduced characters. The observation result is applied only if the current world version matches the version of the state used for observation. This prevents outdated results from updating the world after the user continues, undoes a turn, or edits the world:

$$\hat{o}_t = \rho(F_t, D_t, \mathcal{E}_t), \quad \mathcal{W} \leftarrow \begin{cases} \mathcal{U}_{\text{obs}}(\mathcal{W}, \hat{o}_t), & \text{ver}(\mathcal{W}) = \text{ver}(\mathcal{W}_t), \\ \mathcal{W}, & \text{otherwise.} \end{cases} \quad (7)$$

Observation runs asynchronously, so playback can begin without waiting for the observation result.

Each node in the segment tree stores its parent and state snapshots. Undoing a turn restores the dynamic state recorded before its segment was generated and moves the head to the parent node, $\mathcal{S} \leftarrow \mathcal{S}_h^-$ and $h \leftarrow \text{pa}(h)$, where $\mathcal{S}_h^-$ is the saved snapshot and $\text{pa}(h)$ is the parent node. The undone segment and its latents remain stored. After an undo, the next turn creates a sibling of the undone segment. Regenerating a turn creates a sibling of the segment being regenerated. These alternatives share a parent and coexist without overwriting one another. The asset registry and the place graph are retained across branches, preserving registered assets and established connections between locations.

2.5 Co-op Mode

In co-op mode, two or more players share one world and act in the same story. Each shared turn is called a round. Each player is associated with a character and its reference image. The host can add a shared environment and a non-player creature, and AI companions can fill available player slots. Each round produces a single joint segment in which all participants act together. The reference images for the players' characters, the creature, and the environment must fit within the limit of $N_{\max}$ images per generation call. Co-op reuses the world state, segment tree, continuation, and versioned commits described above. It differs in how joint actions are planned, how observed outcomes are recorded, and how waiting time is reduced.

Joint shot compilation. Single-player planning distributes the events in a user's request across segments. Co-op planning must additionally coordinate simultaneous actions from multiple players within a single continuous shot while keeping each action associated with its intended actor. Each submitted action u_t^i is converted into an action contract κ_t^i , a structured description of the actor, physical action, target, path, and requested end state. Once all players have submitted their actions, the agent fixes the inputs for the round as

$$\mathcal{W}_t^* = \left(\mathcal{W}_{t-1}, \{ (e_i, \kappa_t^i) \}_{i=1}^M, r_t \right), \quad (8)$$

where e_i is player i 's character, M is the number of players, and r_t is a revision counter that increments whenever the participating characters or a shared asset changes. Work prepared for an earlier revision or different actions is discarded.

A director model plans the opening and closing compositions, camera strategy, actor staging, temporal overlap, and sequence of visual emphasis. A writer model converts this plan into shot text $\tilde{y}_t$, using a placeholder $\langle\langle i \rangle\rangle$ for each action. The compiler then inserts the locked action contracts:

$$y_t = \tilde{y}_t \left[\langle\langle i \rangle\rangle \mapsto \langle \text{Subject } \iota_t(e_i) \rangle \kappa_t^i \right]_{i=1}^M, \quad (9)$$

Each placeholder must occur exactly once. The director and writer models determine how the actions are presented, while the compiler inserts the locked action contracts. Each contract appears exactly once in the compiled prompt and is bound to the corresponding actor. Segment duration is computed from $\mathcal{W}_t^*$ by a deterministic function that allocates more time for additional actors, more complex actions, dialogue, and a pending world event. All computations using the same round snapshot therefore use the same duration.

Forecasts and observed facts. Co-op distinguishes planned events from observed outcomes. A session begins with a short opening segment generated from the world setting and the selected references. If reference images for the shared environment or creature were not provided in advance, the agent extracts them from the opening segment's keyframes, provided that the corresponding environment or creature is visible. A creature introduced later is registered only after it appears on screen.

Each round also produces a forecast g_{t+1} for the next round, consisting of a world event, such as an environmental change or a creature action, and one suggested action per player. The forecast is computed while the current segment is being rendered, using confirmed history and the submitted actions without video frames as input:

$$g_{t+1} = \Gamma(\mathcal{N}_{t-1}^{\text{conf}}, \{u_t^i\}_{i=1}^M), \quad \mathcal{N}_t^{\text{conf}} = \mathcal{N}_{t-1}^{\text{conf}} \cup \{ \rho(\text{frames}(X_t), g_t) \}, \quad (10)$$

The forecast g_{t+1} is released only after the current segment has been committed. The observation model ρ reviews the keyframes of X_t and classifies the world event forecast for the current round as observed, partially observed, not observed, or unknown. It also records the visible states of the environment, the creature, and each character, together with the story phase. Requested actions are treated as observed outcomes only when supported by the generated frames. Only observations are added to the confirmed history $\mathcal{N}^{\text{conf}}$.

Reducing waiting time. A co-op round must wait for all players to submit their actions, and several model calls precede rendering. Action contracts are therefore prepared as soon as each player submits an action, while the others are still deciding. During playback of the current segment, the joint shot for the suggested actions is planned and rendered speculatively. Let $\hat{\mathcal{W}}_{t+1}^*$ denote the round snapshot used for speculative generation. The speculative segment $\hat{X}_{t+1}$ is reused only if the submitted round matches this snapshot according to $\text{sig}(\mathcal{W}_{t+1}^*) = \text{sig}(\hat{\mathcal{W}}_{t+1}^*)$. The signature includes the head node, shared assets, participating characters, and exact action text. Any mismatch cancels the speculative job. The speculative segment is added to the world history only after it is accepted.

Audio-visual Generation

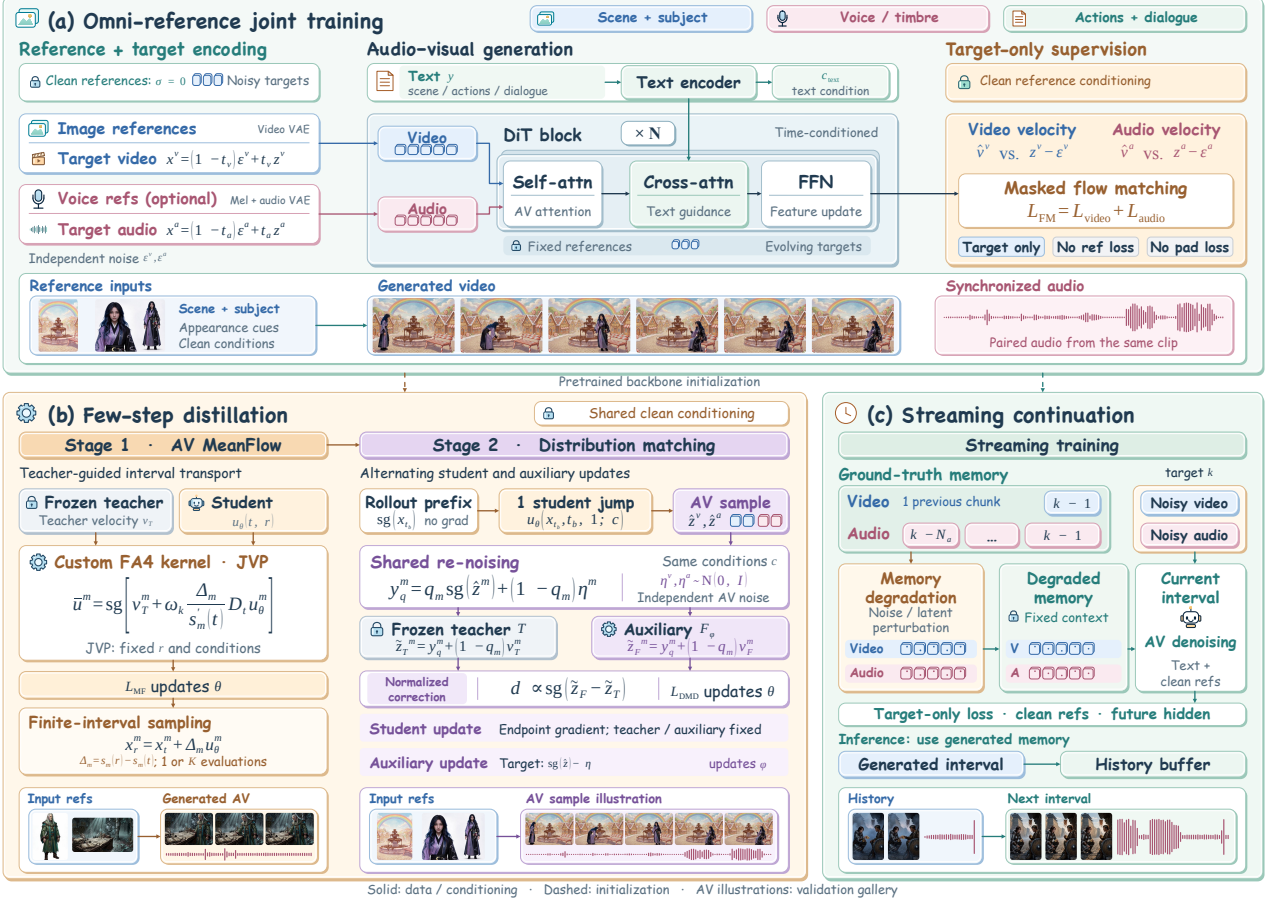


Figure 3. Overview of the audiovisual generation framework. (a) Omni-reference joint training encodes visual references and optional voice examples as clean conditions, injects text through cross-attention, and predicts video and audio velocities from noisy target latents. Flow-matching supervision applies only to valid target entries. (b) Few-step distillation first learns teacher-guided finite-interval transport with AV MeanFlow, using a custom FlashAttention-4 kernel for the Jacobian–vector product, and then applies distribution matching with alternating student and auxiliary updates. In Stage 2, sg denotes stop-gradient and ϕ denotes the auxiliary parameters. (c) Streaming continuation conditions the current interval on bounded video and audio histories. During training, modality-specific noise or latent perturbations degrade ground-truth history into fixed memory conditions for current-interval denoising. References stay clean, and only current targets receive supervision. At inference, generated intervals update the history buffer. Dashed arrows denote initialization from the pretrained backbone. Attention blocks are shown schematically; N denotes the number of repeated DiT blocks. Video frames and paired audio waveforms illustrate the workflows using validation samples; they do not constitute stage-specific evaluation results.

3 Audio-visual Generation

The audiovisual generator realizes the segment-level conditions assembled by World Agent (Section 2). Text specifies the scene, actions, and dialogue; visual references specify appearance; and optional audio references provide voice characteristics. Unlike the agent’s persistent symbolic state, the generator uses supplied audiovisual latents to continue previously generated content. Figure 3 summarizes the reference-conditioned backbone, sequential MeanFlow-to-DMD distillation, and streaming continuation. Section 3.1 defines the reference-conditioned bidirectional backbone. Section 3.2 describes two-stage distillation: MeanFlow learns finite-interval transport, and Distribution Matching Distillation (DMD) refines the generated distribution. Section 3.3 introduces bounded-history training for stream continuation and explains how distillation extends to this setting. These components address reference fidelity, sampling cost, and temporal continuity; their presentation order does not prescribe a single training sequence.

3.1 Omni-Reference Generation

We build a joint video–audio generator conditioned on text, reference images, and optional voice examples [35, 39]. Text specifies scenes, actions, and dialogue; images provide appearance cues; voice examples provide acoustic

evidence for timbre conditioning. References enter as clean latent tokens with modality-specific source IDs and controlled attention connections. We distinguish image-reference SFT, which uses target audio without voice references, from its audio-reference extension below.

From Noise to Video and Audio. Flow Matching learns a time-dependent velocity field that transports a simple noise distribution to the data distribution [23]. We apply it in the latent space of synchronized video and audio. Let $z = (z^v, z^a)$ denote the clean target latent sequences, with modality index $m \in \{v, a\}$. Under the noise-to-data convention used in this report, we interpolate between Gaussian noise ϵ^m and the clean latents using a shared time variable $\tau \in [0, 1]$:

$$x_\tau^m = (1 - t_m)\epsilon^m + t_m z^m, \quad \epsilon^m \sim \mathcal{N}(0, I), \quad t_m = s_m(\tau). \quad (11)$$

Each s_m increases from zero to one, so the sample-wise velocity with respect to t_m is $z^m - \epsilon^m$. The implementation expresses the same path using a noise level $\sigma_m = 1 - t_m$: $x^m = (1 - \sigma_m)z^m + \sigma_m\epsilon^m$, with regression target $\epsilon^m - z^m$. Accordingly, the velocity $\hat{v}^m$ below denotes the negative of that implementation output. This change of convention preserves the path, loss, and sampling trajectory.

The image-reference SFT configuration uses a shared noise level $\sigma_v = \sigma_a$, sampled from a shifted logit-normal distribution with a uniform-mixture component; hence $s_v = s_a$ in this setting. We retain the modality-indexed notation for the subsequent generation formulation. At inference, integration proceeds from noise to data while references stay clean. If integration is parameterized by τ , the velocity for modality m is multiplied by $s'_m(\tau)$.

Multimodal Reference Conditioning. Each image I_i is independently encoded by the video VAE into one clean latent frame and patchified into r_i^v , preserving its spatial grid. For voice conditioning, each reference waveform A_j is converted to a mel spectrogram, encoded by the matching audio VAE, and patchified into r_j^a . Audio references may have different durations and retain their temporal structure; they supply acoustic latent sequences rather than a global speaker embedding. Text y and the reference sequences form the following conditioning inputs:

$$\begin{aligned} c &= (c_{\text{text}}, \mathcal{R}_v, \mathcal{R}_a), & c_{\text{text}} &= E_{\text{text}}(y), \\ \mathcal{R}_v &= ((r_i^v, i))_{i=1}^{N_v}, & r_i^v &= P_v(E_v(I_i)), \\ \mathcal{R}_a &= ((r_j^a, j))_{j=1}^{N_a^{\text{ref}}}, & r_j^a &= P_a(E_a(\text{Mel}(A_j))). \end{aligned} \quad (12)$$

Here E_m and P_m are the VAE encoder and patchifier for modality m . N_a^{ref} counts supplied voice examples, with $\mathcal{R}_a = ()$ when none are supplied. References are concatenated along the *token dimension*, leaving latent channel widths unchanged. At inference, we concatenate target and reference tokens:

$$\begin{aligned} \tilde{x}_\tau^v &= [x_\tau^v; r_1^v; \dots; r_{N_v}^v], \\ \tilde{x}_\tau^a &= [x_\tau^a; r_1^a; \dots; r_{N_a^{\text{ref}}}^a]. \end{aligned} \quad (13)$$

The audio suffix is omitted when absent. The image-reference trainer stores visual groups as $[r_1^v; \dots; r_{N_v}^v; x_\tau^v]$; coordinates, IDs, and attention masks follow the groups, so source identity is independent of storage order relative to the target.

Only target latents are corrupted by Gaussian noise. Both reference modalities use zero per-token noise levels ($\sigma = 0$), are held fixed by the denoising mask, and are excluded from the training loss. Reference tokens are removed before decoding. Their clean latent values are fixed; their hidden states still evolve through permitted network operations.

Modality-Specific Reference IDs. Within each sample, images and audio references are numbered independently in list order: $1, \dots, N_v$ and $1, \dots, N_a^{\text{ref}}$. All tokens from one reference share its ID; target and padding tokens use zero. Each modality supports up to 20 references. Let q_i be the one-hot vector for a reference slot and $q_0 = 0$. For $m \in \{v, a\}$, we add an ID embedding to each projected token:

$$h_{i,k}^m = W_m r_{i,k}^m + b_m + U_m q_i = [W_m \ U_m] \begin{bmatrix} r_{i,k}^m \\ q_i \end{bmatrix} + b_m, \quad (14)$$

where k indexes tokens and U_v, U_a are separate learned ID projections. An embedding lookup adds $U_m q_i$ after latent projection, equivalently a linear projection of the latent token and its one-hot slot marker. The audio-ID table is independent of the existing visual-ID table and is initialized to zero when extending a checkpoint; index

zero stays zero. IDs are preserved through packing, continuation-pair expansion, distributed partitioning, and validation. Multiple audio references require ordered IDs; numbering restarts for each sample and modality.

These IDs identify reference slots, not persistent speaker identities. In particular, visual ID 1 and audio ID 1 do not automatically denote the same person. A character–voice assignment must be specified by paired metadata and the structured caption convention (Section 5.2), which associates a character with its `Audio_j` reference and dialogue. Slot IDs make the supplied examples distinguishable; learning the association requires corresponding supervision. ID differences alone also impose no attention isolation.

Reference Positions. RoPE [37] is computed from explicit token coordinates. For reference images, spatial coordinates retain each image’s own grid, while the temporal interval of every token in image i is shifted to share the following center:

$$\alpha_i = o - (N_v - i)\Delta, \quad i = 1, \dots, N_v. \quad (15)$$

The current setting uses $o = -1$ s and $\Delta = 1$ s, placing the ordered references at $(-N_v, \dots, -1)$ seconds before the target timeline. For example, three reference images occupy temporal centers -3 , -2 , and -1 seconds, respectively. These are conditioning coordinates; they do not indicate that the images were observed at those physical times. Both endpoints of each token’s temporal interval receive the same shift, preserving its extent and the image’s spatial structure.

Each audio reference is positioned independently. Let $p_{j,k}^a$ be the temporal interval of token k in reference j , and D_j its final right endpoint. We translate both interval endpoints by the same amount:

$$\tilde{p}_{j,k}^a = p_{j,k}^a - D_j - g, \quad g = 0.04 \text{ s}. \quad (16)$$

Every reference therefore ends at -0.04 seconds while preserving its internal timing; the target timeline is unchanged. These examples are not laid out as consecutive past utterances: their negative coordinates may overlap, while ordered audio IDs distinguish their sources. Voice references supply acoustic examples independently of the chronological audio history used for streaming continuation.

Attention mechanism. The generator maintains separate video and audio streams, each with self-attention and cross-attention to its text condition. Target video tokens attend to the full target clip and all visual references. Within video self-attention, each visual reference attends only to its own tokens. Audio tokens use full self-attention within the same sample, including any appended audio-reference tokens. Packed training examples remain isolated from one another by their attention masks.

Audio–video cross-attention excludes *both* visual and audio reference tokens from queries, keys, and values in both directions. Each modality’s reference positions are removed before the cross-attention projections. Target audio and video remain connected; continuation additionally retains history connections permitted by the temporal layout. Reference information first enters its own modality through self-attention and can then reach the other modality through target tokens. Thus the design separates direct reference access while retaining audiovisual coupling. Reference hidden states still receive permitted self-attention, text attention, and feed-forward updates. With the noisy targets $\mathcal{X}_\tau = (x_\tau^v, x_\tau^a)$ and time pair $\mathbf{t} = (t_v, t_a)$, the model predicts both target velocities:

$$(\hat{v}^v, \hat{v}^a) = F_\theta(\mathcal{X}_\tau, \mathbf{t}; c, \mathcal{G}), \quad (17)$$

where $\mathcal{G}$ specifies the attention connections and F_θ uses the velocity convention defined above. Target attention is bidirectional within the clip; Section 3.3 describes streaming history visibility.

Joint Audiovisual Training Objective. Image-reference SFT uses a target video, its available synchronized audio, text, and clean image references, with $\mathcal{R}_a = ()$. The audio-reference extension additionally supplies the ordered clean voice latents and their IDs, retaining the same joint flow-matching objective. We sample noise, construct the noisy target sequences using Equation (11), and predict both velocities in one joint forward pass. For each modality, let $M_{b,m}$ be a binary mask for example b , broadcast over channels, that selects valid target entries and excludes references, padding, and other conditioning entries. For a packed batch $\mathcal{B}$, let $\mathcal{B}_m$ contain the examples with target modality m . We minimize this loss:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{\mathcal{B}, \epsilon, \tau} \left[\sum_{m \in \{v, a\}} \frac{\lambda_m}{\max(|\mathcal{B}_m|, 1)} \sum_{b \in \mathcal{B}_m} \frac{\|M_{b,m} \odot (\hat{v}_b^m - (z_b^m - \epsilon_b^m))\|_2^2}{\max(\|M_{b,m}\|_1, 1)} \right], \quad (18)$$

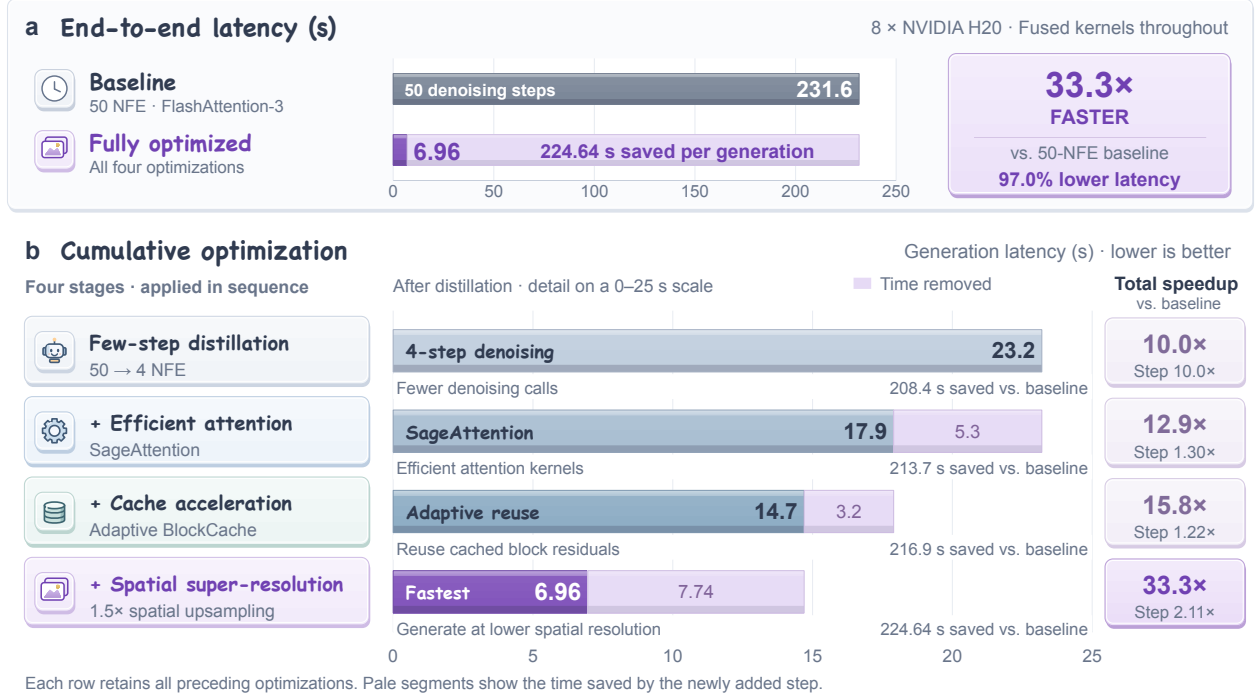


Figure 4. Cumulative inference speedup for the Murea-Pro deployment on eight NVIDIA H20 GPUs, with fused kernels enabled throughout. (a) End-to-end latency decreases from 231.6 s for the 50-NFE baseline to 6.96 s for the fully optimized system, yielding a 33.3× speedup. (b) Few-step distillation (50 to 4 NFE), SageAttention, Adaptive BlockCache, and lower-resolution generation with 1.5× spatial super-resolution are applied cumulatively, with each row retaining all preceding optimizations. Pale segments in (b) indicate the time saved by each newly added step; the right column reports total speedup relative to the baseline and the incremental gain from each step.

where an empty $\mathcal{B}_m$ contributes zero. The mask norm counts valid scalar entries after channel broadcasting. Each example’s squared error is averaged over its valid target tokens and channels; packed examples are then averaged per modality, and the video and audio losses are added with $\lambda_v = \lambda_a = 1$. This prevents a longer sequence from dominating solely through its token count. Samples without target audio contribute only the video term. Cross-modal attention couples the two predictions even though the loss is a sum of modality-specific errors. Reference images and any voice examples influence these predictions through attention without acquiring denoising targets of their own. This bidirectional model provides the starting point for the few-step and streaming extensions.

Voice-conditioned supervision requires speaker-matched references that do not overlap the supervised target interval, with source and pairing provenance retained. This separates the desired voice cue from direct access to target content. The audio-reference extension has been functionally validated; these checks and the image-reference evaluations do not establish voice preservation or correct multi-speaker binding, which require dedicated assessment.

3.2 Few-step Distillation

We distill the bidirectional audiovisual generator in two stages. First, audiovisual (AV) MeanFlow learns finite-interval transport from high-quality audiovisual data under a frozen teacher, allowing the sampling step count to vary. Second, Distribution Matching Distillation (DMD) refines the generated distribution to improve visual detail and motion. Both stages jointly model video and audio ($m \in \{v, a\}$) under the text and reference conditions c of Section 3.1. Conditioning latents are held fixed during target denoising and excluded from target losses. We retain the noise-to-data convention throughout.

AV MeanFlow with Variable-Step Sampling. Let $0 \leq t < r \leq 1$ be shared current and destination times. Using the modality schedules from Section 3.1, define $t_m = s_m(t)$, $r_m = s_m(r)$, and $\Delta_m = r_m - t_m$. We assume smooth schedules with $s_m(0) = 0$, $s_m(1) = 1$, and $s'_m > 0$. The frozen teacher v_T^m predicts velocity with respect to modality time. Following MeanFlow [10], we define its interval-average velocity through the joint teacher

trajectory X_s :

$$\begin{aligned} \frac{dX_s^m}{ds} &= s'_m(s)v_T^m(X_s, s; c), & X_t &= x_t, \\ u_T^m(x_t, t, r; c) &= \frac{X_r^m - x_t^m}{\Delta_m}. \end{aligned} \quad (19)$$

Here $x_t = (x_t^v, x_t^a)$, and the shared time argument determines both modality times through s_v and s_a . Training samples $x_t^m = t_m z^m + (1 - t_m)\epsilon^m$ with independent modality noise. These interpolated states initialize teacher trajectories; X_r need not lie on the data-noise line. Teacher endpoints define u_T but are not integrated during training.

We initialize the student from the pretrained audiovisual backbone and add an r -conditioning branch that embeds Δ_m alongside the current time. Its output projection is initialized to zero, preserving the initial backbone prediction. The student advances each modality as follows:

$$x_r^m = x_t^m + \Delta_m u_\theta^m(x_t, t, r; c). \quad (20)$$

The update approximates teacher transport. Taking $(t, r) = (0, 1)$ gives one-step generation; a partition $0 = t_0 < \dots < t_K = 1$ gives K model evaluations. Thus one parameterization supports different step counts, although approximation error and sample quality depend on the schedule. Training draws ordered time pairs, optionally mixed with diagonal pairs $r = t$, and shares each pair across modalities.

With r and c fixed, we differentiate along the joint teacher flow:

$$D_t u_\theta^m = \partial_t u_\theta^m + \sum_{n \in \{v, a\}} s'_n(t) (\partial_{x^n} u_\theta^m) v_T^n. \quad (21)$$

Differentiating Equation (19) along this trajectory gives $u_T^m = v_T^m + (\Delta_m / s'_m(t)) D_t u_\theta^m$. We use the corresponding bootstrapped student target:

$$\bar{u}^m = \text{sg} \left[v_T^m + \omega_k \frac{\Delta_m}{s'_m(t)} D_t u_\theta^m \right], \quad \omega_k = \min(1, k/N_w). \quad (22)$$

Here sg is stop-gradient, k counts optimizer updates, and N_w is the warmup length; without warmup, $\omega_k = 1$. We compute the directional derivative as a Jacobian-vector product (JVP) using our custom FlashAttention-4 (FA4) kernel, holding r and all conditioning fixed. Fixed conditioning entries have zero state direction. For $s_m(t) = t$, the correction reduces to $(r - t) D_t u_\theta^m$; the schedule factors above are needed when video and audio use different clocks. On the diagonal, we set the target directly to v_T^m , using the limit $u_T^m(x_t, t, t; c) = v_T^m(x_t, t; c)$ without evaluating a zero denominator.

For each example, let M_m mask valid target scalar entries and $N_m = \|M_m\|_1 > 0$. We compute the residual and normalize the regression direction:

$$e^m = M_m \odot (u_\theta^m - \bar{u}^m), \quad h^m = \text{sg} \left[\frac{e^m}{\sqrt{N_m} (\|e^m\|_2 / \sqrt{N_m} + \kappa)^p} \right]. \quad (23)$$

We then train the student with the resulting MeanFlow loss:

$$\mathcal{L}_{\text{MF}}(\theta) = \sum_m w_m \mathbb{E}_m \left[\|M_m \odot (u_\theta^m - \text{sg}(u_\theta^m - h^m))\|_2^2 \right]. \quad (24)$$

Here $\mathbb{E}_m$ averages over examples with active targets in modality m , noise, and intervals. The norm sums valid entries, with scaling supplied by h^m ; defaults are $p = 1$, $\kappa = 10$, and $w_v = w_a = 1$.

Distribution Matching Refinement. Starting from the AV MeanFlow model, DMD [45, 46] refines the generated distribution to improve visual quality and motion under few-step sampling. A frozen teacher (T) estimates the score of the noised target distribution, while a trainable auxiliary denoiser (F) tracks the noised student distribution. Their score difference guides the student without paired teacher-generated targets. Only the student inherits the MeanFlow weights and endpoint branch. Both denoisers start from the pretrained instantaneous-velocity model and receive no destination time r .

We run a prefix of the student’s sampling schedule without gradients. At a selected nonterminal boundary t_b , a single differentiable forward maps the detached state directly to the clean endpoint:

$$\hat{z}^m = x_{t_b}^m + (1 - s_m(t_b)) u_\theta^m(x_{t_b}, t_b, 1; c), \quad x_{t_b} = \text{sg}(x_{t_b}). \quad (25)$$

Selecting the final boundary gives the full scheduled rollout; selecting an earlier boundary gives a shorter prefix followed by an endpoint jump. It does not backpropagate through the remaining schedule edges. At a fresh shared $q \in (0, 1)$, let $q_m = s_m(q)$ and form $y_q^m = q_m \text{sg}(\hat{z}^m) + (1 - q_m)\eta^m$ with independent Gaussian noise. Both denoisers see the same noisy state and conditions. We compute their clean estimates and the normalized correction:

$$\begin{aligned}\hat{z}_Q^m &= y_q^m + (1 - q_m)v_Q^m(y_q, q; c), & Q \in \{T, F\}, \\ d^m &= \text{sg}\left[\frac{\hat{z}_F^m - \hat{z}_T^m}{\max(\text{mean}_{M_m}|\hat{z}^m - \hat{z}_T^m|, 10^{-8})}\right].\end{aligned}\quad (26)$$

For this Gaussian perturbation, the denoiser-implied score in modality m is $(q_m \hat{z}_Q^m - y_q^m)/(1 - q_m)^2$. Since $0 < q_m < 1$, $\hat{z}_F^m - \hat{z}_T^m$ has the direction of the auxiliary-minus-teacher score difference. The normalization rescales this direction by a positive factor. Here masks include the minibatch dimension, and masked means pool valid scalar entries within each modality. With $\|a\|_{M_m}^2 = \|M_m \odot a\|_2^2 / \max(\|M_m\|_1, 1)$, we train the student with the following loss:

$$\mathcal{L}_{\text{DMD}}(\theta) = \frac{1}{2} \sum_m w_m \mathbb{E} \left[\|\hat{z}^m - \text{sg}(\hat{z}^m - d^m)\|_{M_m}^2 \right]. \quad (27)$$

On valid entries, the gradient with respect to $\hat{z}^m$ is proportional to d^m ; the parameter gradient additionally passes through the student Jacobian. Gradient descent therefore uses the negative auxiliary-minus-teacher correction. The expectation covers minibatches, student noise, boundary selection, and score time and noise. Both denoisers are fixed during this update. DMD retains the interval-conditioned student, but quality at each inference step count must still be evaluated; endpoint refinement does not guarantee equal quality across all schedules.

Training Schedule. The student objective uses phase-dependent weights:

$$\mathcal{L}_{\text{student}}(\theta; k) = \lambda_{\text{MF}}(k) \mathcal{L}_{\text{MF}}(\theta) + \lambda_{\text{DMD}}(k) \mathcal{L}_{\text{DMD}}(\theta). \quad (28)$$

We first train AV MeanFlow with $(\lambda_{\text{MF}}, \lambda_{\text{DMD}}) = (1, 0)$, then use its weights to initialize DMD refinement with $(0, 1)$. The stages are sequential; the MeanFlow loss is not evaluated during DMD. The auxiliary denoiser has separate parameters ϕ and learns from detached student outputs:

$$\mathcal{L}_{\text{aux}}(\phi) = \sum_m w_m \mathbb{E} \left[\|v_F^m(y_q, q; c; \phi) - (\text{sg}(\hat{z}^m) - \eta^m)\|_{M_m}^2 \right]. \quad (29)$$

After optional auxiliary warmup, student and auxiliary updates alternate. The auxiliary loss updates only ϕ ; the teacher remains frozen.

3.3 Streaming Audio-Visual Generation

Long-horizon streaming training. We initialize streaming training from the bidirectional audiovisual backbone and learn to extend a stream with bounded context. Let $z_k = (z_k^v, z_k^a)$ be the target interval at position k , and let $H_k = (H_k^v, H_k^a)$ contain its retained history. For a fixed condition c , we define a finite-memory autoregressive model:

$$p_\theta(z_{1:K} | c) = \prod_{k=1}^K p_\theta(z_k | c, H_k), \quad (30)$$

where c contains text and reference conditions. This factorization is the model’s finite-memory assumption, not a claim that the data depend only on the retained past. If the agent changes the prompt or references, the corresponding factor instead uses the current condition c_k . Following the long-horizon training direction explored by Helios and Alaya-EVOKE [47, 50], each example supervises a complete spatiotemporal interval through one joint denoising objective, rather than decomposing supervision into frame or token predictions. The model therefore learns stream extension under the same bounded-context regime used at inference time.

Our bounded-memory model retains one preceding video chunk and up to N_a audio chunks. The first interval has $H_1^v = H_1^a = ()$. For $k > 1$, we define the histories as follows:

$$\begin{aligned}H_k^v &= [z_{k-1}^v], \\ H_k^a &= [z_{\max(1, k-N_a)}^a, \dots, z_{k-1}^a], & N_a > 1.\end{aligned}\quad (31)$$

Here N_a counts retained audio chunks; their combined duration depends on the chunk layout. The compact audio representation permits a longer window, intended to retain linguistic and prosodic cues across boundaries and

reduce sentence restarts or repetition. Voice references provide a persistent timbre cue, whereas this chronological memory supplies context from the ongoing utterance.

Each training video can yield multiple (H_k, z_k) pairs. We apply a history operator $\mathcal{A}$ before conditioning:

$$\tilde{H}_k = \mathcal{A}(H_k), \quad \mathcal{A} \sim p(\mathcal{A}). \quad (32)$$

During training, H_k comes from preceding ground-truth intervals. The operator $\mathcal{A} = (\mathcal{A}_v, \mathcal{A}_a)$ applies modality-specific corruption, such as noise or low-cost latent perturbations, to history only. The resulting history is held fixed while denoising the current target; it need not be clean. Reference latents remain clean, and the target retains its original clean supervision. This teacher-forced training exposes the model to imperfect context, but does not reproduce the full distribution of its own rollout errors.

At inference, previously generated intervals fill the history slots. The training augmentation does not imply that generated history must be degraded again at inference. Sequence packing combines variable-length histories and targets for efficient training while preserving sample boundaries in the attention mask and the per-example loss normalization.

For modality $m \in \{v, a\}$, we perturb the current target interval along the path in Equation (11):

$$x_{k,\tau}^m = (1 - t_m)\epsilon_k^m + t_m z_k^m, \quad \epsilon_k^m \sim \mathcal{N}(0, I). \quad (33)$$

Let M_k^m select only the target tokens in the current interval. We train the streaming model with the following flow-matching objective:

$$\mathcal{L}_{\text{stream}} = \mathbb{E} \left[\sum_{m \in \{v, a\}} \lambda_m \frac{\|M_k^m \odot (v_{\theta, \text{st}}^m(x_{k,\tau}, \mathbf{t}; c, \tilde{H}_k) - (z_k^m - \epsilon_k^m))\|_2^2}{\max(\|M_k^m\|_1, 1)} \right], \quad (34)$$

where the expectation samples sequences, interval boundaries, time, noise, and history degradations. Packed batches use the per-modality example averaging of Equation (18). Reference and history tokens are context only and are excluded from the loss. All video and audio tokens in the target interval may interact bidirectionally during denoising, while future intervals remain unavailable through H_k . Video and audio histories can have different lengths, but their positions are anchored to a shared temporal origin to preserve audiovisual alignment.

Extending distillation to streaming. The objectives in Section 3.2 also admit history conditioning. This extension requires a teacher trained for stream continuation and a student with the same history layout; a bidirectional teacher without that conditioning is not interchangeable. Replacing c with $(c, \tilde{H}_k)$ gives the corresponding MeanFlow and DMD objectives. Within each training example, the teacher, student, and auxiliary denoiser must receive identical fixed history and references. Derivatives and denoising losses act only on the current interval. During inference, each generated interval updates the history before the next interval is sampled. This describes how the distillation formulation extends to streaming; it does not by itself establish long-horizon quality, which requires evaluation on generated-history rollouts.

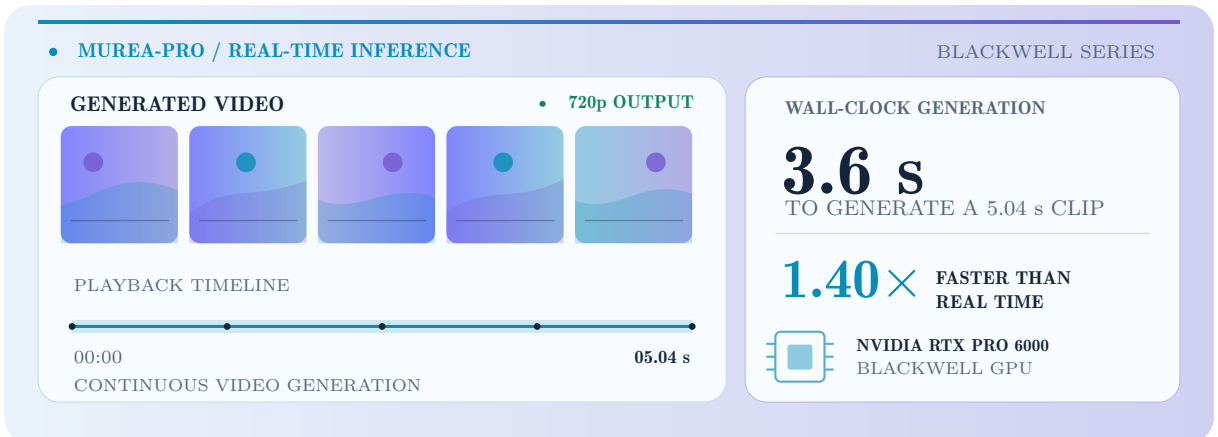


Figure 5. Murea-Pro generates 5.04s of 720p video in 3.6s on NVIDIA RTX PRO 6000 Blackwell (1.40× real time; frames are schematic).

4 Efficient Inference

Long audiovisual token sequences make each denoising evaluation expensive. Our inference implementation reduces the number of function evaluations (NFE) through few-step distillation (Section 3.2). Adaptive block caching, fused kernels, multi-GPU parallelism, and optimized attention backends lower the cost of each evaluation. Native reference caching avoids repeated encoding, while spatial super-resolution allows the base video generator to operate at a lower resolution.

Multi-GPU Parallelism. Ulysses [13] partitions the audiovisual token sequence across GPUs. All-to-all exchanges before and after attention convert between sequence and head partitions, allowing each device to attend to the full sequence for a subset of heads. This distributes activations and computation across devices while preserving the full attention context. The same parallel layout supports either attention backend described below.

Efficient Attention. Attention backends are selected according to the target GPU architecture. The configuration evaluated in Figure 4 uses SageAttention [52], which accelerates attention matrix products through low-precision quantization and specialized GPU kernels. FlashAttention-4 (FA4) [51] provides an alternative on supported hardware. Its tiled computation reuses intermediate values in on-chip storage, reducing memory traffic without materializing the full attention matrix in device memory. Each deployment uses one of these backends for the attention operation.

Cache Acceleration. Cache acceleration reuses computation across denoising steps and repeated requests. Adaptive BlockCache approximates a contiguous range of intermediate Transformer blocks by reusing their residual across denoising steps. Cumulative hidden-state drift and a limit on consecutive reuse determine when the cached residual can replace block execution. Blocks before and after the cached range continue to run at every step; a full evaluation refreshes the residual when the reuse criteria are no longer met. Native reference caching additionally reuses VAE encodings of unchanged reference images, avoiding repeated encoding for recurring assets.

Spatial Super-Resolution. VideoSharpenSR applies $1.5\times$ spatial upscaling to the decoded video, allowing the base generator to operate at a lower spatial resolution while retaining the delivered output dimensions. Reducing the number of spatial tokens lowers the denoising workload, while the lower-resolution latents also reduce decoding cost.

Acceleration Evaluation. Figure 4 reports cumulative inference speedup for Murea-Pro, measured on eight NVIDIA H20 GPUs, with fused kernels enabled throughout. The baseline uses 50 NFE with FlashAttention-3 and both reference caching and Adaptive BlockCache disabled. Subsequent configurations introduce few-step distillation to reduce NFE to 4, SageAttention, cache acceleration, and lower-resolution generation with spatial super-resolution, in that order. On NVIDIA RTX PRO 6000 Blackwell GPUs, the same pipeline generates a 5.04 s, 720p clip in 3.6 s, faster than its playback duration (Figure 5).

5 Experiments

5.1 Data Preparation

We construct a dataset for multi-reference audiovisual generation by combining data created for this work with a curated subset of open-source dataset. Automated filtering screens samples for resolution, duration, audio availability, visual quality, and reference–target identity consistency. Human annotators then evaluate reference consistency, audio consistency, and video quality to determine the final selection.

Data Acquisition. We construct a dataset for multi-reference audiovisual generation by combining data created for this work with a curated subset of HuMoSet [4]. Each sample consists of a target video, its accompanying audio, and an ordered set of reference images depicting characters, scenes, or objects. For HuMoSet, we filter videos by resolution, duration, and audio availability, followed by visual-quality assessment using DOVER [44] and LAION-Aesthetics [20]. We then use an ArcFace-based face-recognition pipeline [6] to evaluate reference–target identity consistency across sampled video frames, excluding mismatched pairs and references without valid faces, while flagging ambiguous cases for human review.

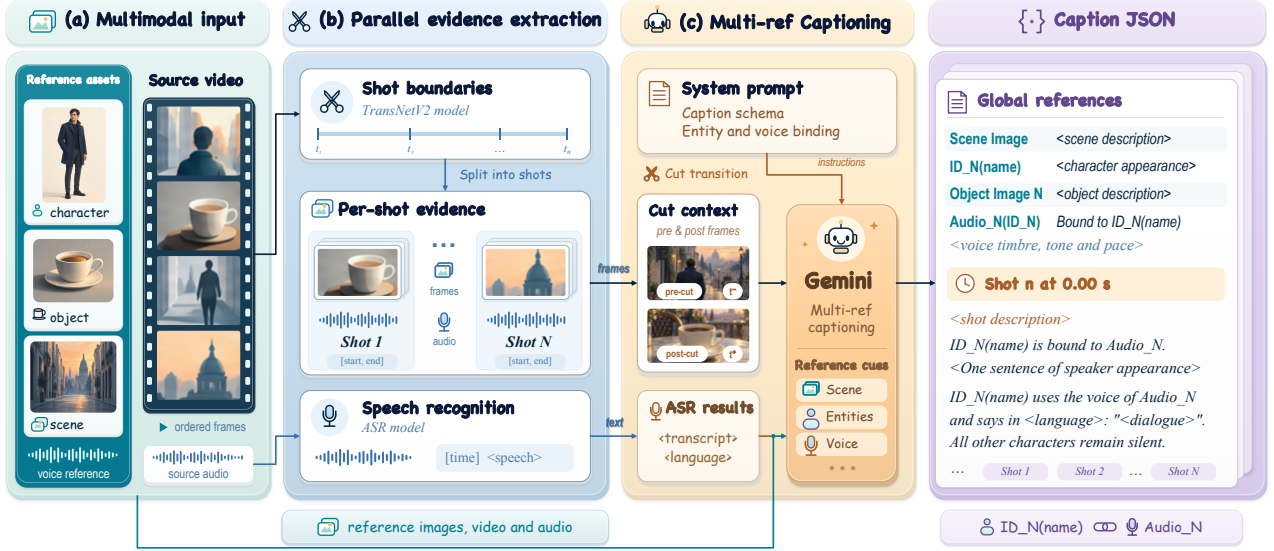


Figure 6. Multi-reference audio-visual captioning pipeline. TransNet detects shot boundaries, and ASR model extracts transcripts and speech timing. Guided by a structured system prompt, Gemini integrates pre- and post-cut frames, ASR results, and visual and voice references to generate caption JSON containing reference descriptions and timestamped shot narratives, with explicit character–voice bindings and dialogue attribution.

Data Annotation. We use human annotation to select high-quality samples according to three criteria: reference consistency, audio consistency, and video quality. Reference consistency evaluates whether the visual references match the corresponding characters, scenes, or objects in the target video. Audio consistency assesses agreement between reference voices and target speakers, where applicable, and alignment between the soundtrack and visual content. Video quality is assessed in terms of visual clarity, temporal coherence, shot-boundary accuracy, and the absence of artifacts such as object interpenetration, geometric distortions, and flickering.

5.2 Captioning

High-quality captions are essential for precise instruction following and video generation quality [35, 40]. For multi-reference audiovisual generation, captioning must specify not only what happens, but also which visual reference defines each entity, which voice reference belongs to each speaker, and when actions and dialogue occur. Ambiguous reference identities, incomplete temporal descriptions, and incorrect speaker attribution can otherwise produce conflicting supervision. We address these annotation challenges with a three-step pipeline: shot-level evidence extraction, multimodal reference grounding, and structured audiovisual caption generation, as illustrated in Figure 6.

Parallel evidence extraction. We apply TransNet [36] to identify shot boundaries and obtain the number of shots and their transition timestamps. Single-shot automatic speech recognition (ASR) extracts dialogue transcripts and speech timing from the corresponding audio segments. Shot annotations, camera metadata, and ASR results are supplied alongside the target video and its soundtrack, providing complementary evidence for visual events, camera behavior, dialogue, and background sounds. This temporal structure anchors the narration to individual shots and supports alignment between actions and speech.

Structured caption generation. We use Gemini 3.1 Pro [41] to integrate the target audio-visual content, extracted evidence, and supplied references into structured caption JSON. The system prompt specifies the reference labels, character–voice bindings, temporal format, and dialogue-attribution rules. Each caption contains two complementary parts:

- **Multi-Reference Description.** Establishes the entities and reference assignments used throughout the caption. It describes the visual attributes of each referenced subject or scene and the vocal characteristics of each voice reference, using the identifiers defined above. Explicit character–voice associations link speakers to their corresponding `Audio_N` references.

Table 1. Quantitative results on OpenS2V-Eval Open-Domain. All scores are recomputed with a common evaluation pipeline. N denotes the number of available clips out of 180 cases, rather than the valid sample count for every metric. Bold and underline indicate the highest and second-highest scores among the reference-to-video systems. Motion reports motion magnitude.

Model	N	Total	Aes.	Smooth.	Motion	FaceSim	Gme	Nexus	Natural
Murea-Fast (8 steps)	180	<u>64.98</u>	52.20	97.91	19.78	58.21	63.69	51.66	86.44
<i>Commercial systems</i>									
Seedance 2.0	148	67.13	49.90	98.25	22.24	60.85	68.80	<u>51.42</u>	<u>92.06</u>
HappyHorse 1.1	177	63.04	52.55	97.84	23.27	56.12	68.97	43.25	83.95
Vidu Q3 Mix	174	61.99	55.84	98.37	24.05	39.62	68.76	47.95	87.12
Kling V3 Omni	167	61.33	48.24	92.78	21.03	52.88	67.24	47.06	81.54
HappyHorse 1.0	163	61.16	52.71	97.66	27.40	36.75	68.07	46.64	89.47
Kling O3	168	60.70	48.88	94.08	20.41	53.00	67.39	45.45	79.37
Wan 2.7 R2V	159	60.25	47.02	95.84	20.53	53.80	63.75	44.63	80.14
Vidu Q3 Turbo	170	58.94	52.38	98.19	<u>34.67</u>	58.09	69.07	41.22	65.93
Kling 1.6	166	58.54	44.49	86.88	40.75	41.37	66.42	44.72	84.19
Grok Video	180	53.42	<u>54.58</u>	99.46	21.80	4.66	72.79	34.30	90.65
Pika 2.1	176	53.33	47.00	87.18	26.60	28.46	69.59	45.15	70.74
Vidu 2.0	180	52.95	41.40	90.22	13.68	36.68	68.49	43.08	68.61
<i>Open-source models</i>									
Bernini	180	63.21	46.89	90.11	25.45	<u>58.93</u>	66.91	48.96	84.12
MiniMax-H3	180	61.01	48.13	<u>99.03</u>	29.63	40.44	69.19	41.05	92.41
VACE-14B	180	59.28	47.21	94.92	15.02	55.09	67.28	44.08	74.26
Phantom-14B	180	58.19	45.96	96.17	33.08	53.59	70.66	37.44	73.84
LTX-2.5 MSR LoRA (Ref2V, 30 steps)	180	57.71	49.26	97.67	24.32	36.32	68.10	46.36	78.24
RefAlign-14B	180	56.60	45.57	97.52	15.33	58.14	59.66	42.13	66.39
Phantom-1.3B	180	55.72	46.27	93.29	14.01	50.02	69.65	41.46	65.79
MAGREF-480P	180	55.04	45.12	92.75	21.92	35.55	70.55	43.48	73.10
SkyReels-A2-P14B	180	52.91	37.92	86.63	26.48	46.31	65.17	43.76	63.66
VACE-1.3B	180	51.05	48.23	97.19	18.83	20.58	71.26	37.93	70.28
VACE-P1.3B	180	50.79	47.28	96.87	12.01	17.89	<u>71.41</u>	41.06	70.05

- **Detailed Audiovisual Description.** Describes the scene chronologically, covering character behavior, spatial relations, camera motion, and ambient sound. Each shot forms a separate paragraph, headed by its index and supplied start time to two decimal places. Each single-speaker turn identifies the character, visual appearance, and associated `Audio_N`, followed by the spoken language and verbatim dialogue; other visible characters are marked as silent. When no voice reference is supplied, the caption states this and describes the voice in the target audio.

This format connects reference attributes with shot-level events and speaker-specific dialogue, providing explicit supervision for reference correspondence, temporal structure, and audiovisual attribution.

5.3 Reference-Conditioned Generation

Benchmark. We evaluate reference-conditioned generation on OpenS2V-Eval and OS2V-Pro. OpenS2V-Eval [49] assesses subject preservation, text alignment, and visual quality. We use its Open-Domain split, which contains 180 cases with real or synthetic references across seven configurations: single-face, single-body, single-entity, multi-face, multi-body, multi-entity, and human-entity. We introduce OS2V-Pro as an extended evaluation on the same benchmark cases, covering reference fidelity, attribute binding, rendering quality, and audio. Both evaluations use the same generated clips.

OpenS2V-Eval reports facial identity similarity (FaceSim), subject similarity within detected regions (NexusScore), text relevance (GmeScore), subject naturalness (NaturalScore), visual aesthetics (AestheticScore), temporal smoothness (MotionSmoothness), and motion magnitude (MotionAmplitude). We summarize these measurements

Table 2. Quantitative results on OS2V-Pro using the available OpenS2V-Eval Open-Domain clips. Pro denotes the OS2V-Pro aggregate, and OpenS2V reproduces the Total score from Table 1. All quality metrics use a 0–100 scale except Decay, Limb, Sharp, and Flick, which retain their native scales. Bold and underline indicate the best and second-best distinct values among the reference-to-video systems listed here; ties at the displayed precision share the same rank. Latency is not ranked. Wall reports seconds per clip under each system’s deployment settings. An em dash denotes unavailable or inapplicable measurements.

Model	Overall		Identity			Instruction		Rendering							Audio		Efficiency
	Pro $\uparrow$	OpenS2V $\uparrow$	Fid $\uparrow$	IntID $\uparrow$	Decay $\uparrow$	VTF $\uparrow$	Bind $\uparrow$	Anat $\uparrow$	Limb $\downarrow$	VQ $\uparrow$	IQA $\uparrow$	Sharp $\uparrow$	Flick $\downarrow$	Usable $\uparrow$	ATQ $\uparrow$	VESA $\uparrow$	Wall (s) $\downarrow$
Murea-Fast (8 steps)	<u>69.9</u>	64.98	46.0	89.0	0.58	95.5	19.6	76.2	0.34	21.6	39.9	−0.28	0.06	91.1	59.7	75.0	43.7
<i>Commercial systems</i>																	
Kling V3 Omni	70.1	61.33	41.9	84.7	0.56	77.1	22.6	82.1	<u>0.31</u>	19.1	41.9	−0.04	0.06	88.0	66.3	88.7	131.8
Vidu Q3 Turbo	69.7	58.94	39.8	89.0	0.62	74.6	<u>21.8</u>	77.6	0.30	20.0	48.7	−0.11	0.07	93.5	59.2	83.3	107
Kling O3	69.4	60.70	41.1	83.8	0.56	77.3	20.9	77.0	0.32	19.0	42.4	−0.05	0.06	88.7	65.7	81.1	—
HappyHorse 1.1	67.7	63.04	41.2	84.8	0.62	74.7	18.7	73.9	0.33	21.5	46.4	−0.38	0.07	94.3	62.6	87.2	126.8
Vidu Q3 Mix	67.4	61.99	41.8	81.7	0.64	76.8	15.2	68.4	0.36	23.2	46.6	0.04	0.08	<u>96.0</u>	66.0	76.7	135
Grok Video	67.0	53.42	30.1	82.4	<u>0.75</u>	70.0	20.2	<u>79.2</u>	0.30	<u>23.0</u>	51.4	−0.08	<u>0.05</u>	91.7	61.0	90.5	64.3
Wan 2.7 R2V	65.6	60.25	39.9	78.5	0.77	72.2	17.8	74.6	0.38	21.3	<u>51.3</u>	−0.12	0.10	96.2	63.6	88.7	106.6
HappyHorse 1.0	64.6	61.16	39.3	78.4	0.68	78.8	16.0	67.7	0.33	21.1	47.8	−0.51	0.08	94.5	76.6	<u>89.2</u>	106.4
Vidu 2.0	44.1	52.95	36.0	79.2	0.63	63.0	20.9	61.4	0.30	15.8	38.1	−0.04	0.04	88.3	—	—	55
Pika 2.1	41.6	53.33	37.6	74.6	0.73	68.6	19.5	49.0	0.33	15.8	33.1	−0.14	0.07	90.9	—	—	—
Kling 1.6	38.7	58.54	38.8	73.7	0.61	74.8	18.9	51.4	0.40	16.9	33.0	−0.11	0.13	93.4	—	—	—
<i>Open-source models</i>																	
MiniMax-H3	64.9	61.01	32.1	83.5	0.72	<u>81.0</u>	16.0	77.0	0.37	20.0	37.0	−0.29	0.08	86.1	64.7	86.1	494.2
LTX-2.5 MSR LoRA (Ref2V)	62.2	57.71	37.9	84.0	0.56	71.5	17.4	58.0	0.34	20.7	38.3	−0.46	0.09	95.0	<u>66.8</u>	<u>74.2</u>	197.4
VACE-14B	45.6	59.28	42.1	85.8	0.57	65.5	19.4	64.1	0.33	19.7	46.5	−0.04	<u>0.05</u>	91.7	—	—	357.4
RefAlign-14B	45.5	56.60	39.2	<u>87.7</u>	0.67	62.7	19.6	54.7	0.33	19.6	45.7	−0.03	<u>0.05</u>	90.6	—	—	473.1
Phantom-1.3B	44.3	55.72	37.3	81.8	0.77	62.8	19.7	59.4	0.37	19.0	50.7	<u>0.09</u>	<u>0.05</u>	83.9	—	—	102.9
Bernini	43.8	<u>63.21</u>	<u>44.8</u>	81.7	0.58	71.8	17.2	62.6	0.35	19.1	44.7	−0.19	0.07	92.2	—	—	527.5
MAGREF-480P	43.6	55.04	35.1	76.6	0.70	69.1	19.2	49.4	0.32	17.6	44.6	0.04	0.07	89.4	—	—	206.7
Phantom-14B	43.5	58.19	34.6	84.4	0.69	67.9	18.3	51.8	0.34	17.7	49.1	−0.14	0.08	89.4	—	—	421.3
VACE-P1.3B	42.9	50.79	33.1	80.1	0.63	61.7	18.9	60.7	0.30	18.8	48.4	0.03	0.04	81.7	—	—	112.0
VACE-1.3B	42.5	51.05	33.4	80.4	0.61	62.4	17.2	58.6	<u>0.31</u>	18.9	50.1	0.00	0.06	88.3	—	—	112.8
SkyReels-A2-P14B	41.9	52.91	40.6	77.6	0.55	63.8	18.4	48.2	0.37	18.3	42.6	0.13	0.06	88.3	—	—	265.4

Fid per-reference fidelity; IntID intra-clip identity consistency; VTF task following; Bind detail binding; Anat anatomical consistency; Limb forearm-length variation; VQ technical quality; IQA single-frame image quality; Sharp signed relative sharpness change (negative = loss of sharpness); Flick temporal flicker; ATQ audio production quality; VESA event–sound alignment; Decay three-reference to one-reference fidelity ratio (1.0 = equal group means); Usable proportion of clips passing the duplicate-frame, freezing, black/flat-frame, and subject-detection checks.

using the fixed aggregate

$$\begin{aligned} \text{Total} = & 0.16 \text{ Aes} + 0.06 \text{ Smooth} + 0.02 \text{ Motion} + 0.20 \text{ FaceSim} \\ & + 0.12 \text{ Gme} + 0.20 \text{ Nexus} + 0.24 \text{ Natural}, \end{aligned} \quad (35)$$

with component scores normalized to [0, 100]. FaceSim is evaluated on the 90 single-person cases, and NexusScore on cases with valid subject detections. MotionAmplitude measures the magnitude of movement without assessing its agreement with the requested action. We recompute NaturalScore for all systems in a single evaluation batch using gpt-4o-2024-11-20 [33] and the original evaluation prompt, averaging three judgments per clip. All NaturalScore comparisons use these recomputed values.

OS2V-Pro measures per-reference fidelity (Fid), intra-clip identity consistency (IntID), checklist-based task following (VTF), and attribute binding (Bind), alongside the rendering and audio metrics defined in Table 2. Decay compares the mean fidelity on three-reference prompts with that on one-reference prompts; it is a ratio across different prompt groups. Gemini 3.1 Pro [41] evaluates VTF, Bind, anatomical plausibility (Anat), and event–sound alignment (VESA).

The OS2V-Pro aggregate assigns equal weight to five groups: identity (FaceSim, Fid, IntID), instruction following (VTF, Bind), rendering (Anat, Limb, VQ, IQA, Sharp, Flick), audio-track completeness, and set-level measures (Decay, Usable, speech-following score). Each group contributes 20%, with equal weights for its constituent metrics. Bounded metrics are mapped to [0, 100] using their defined ranges; Limb, Sharp, and Flick use min–max normalization over the fixed evaluation pool, including configurations not displayed in the table. Audio production quality (ATQ) and VESA are reported separately and are excluded from the aggregate. Audio-track completeness and the speech-following score contribute to the aggregate but are not displayed in the table. Completeness requires an audio track whose duration differs from the video by less than 10%. The speech-following score measures the speech-detection hit rate on prompts requesting speech minus the false-positive rate on other prompts, mapped from [−1, 1] to [0, 100]. Thus, the aggregate rewards audio availability; its audio group does not measure audio quality.

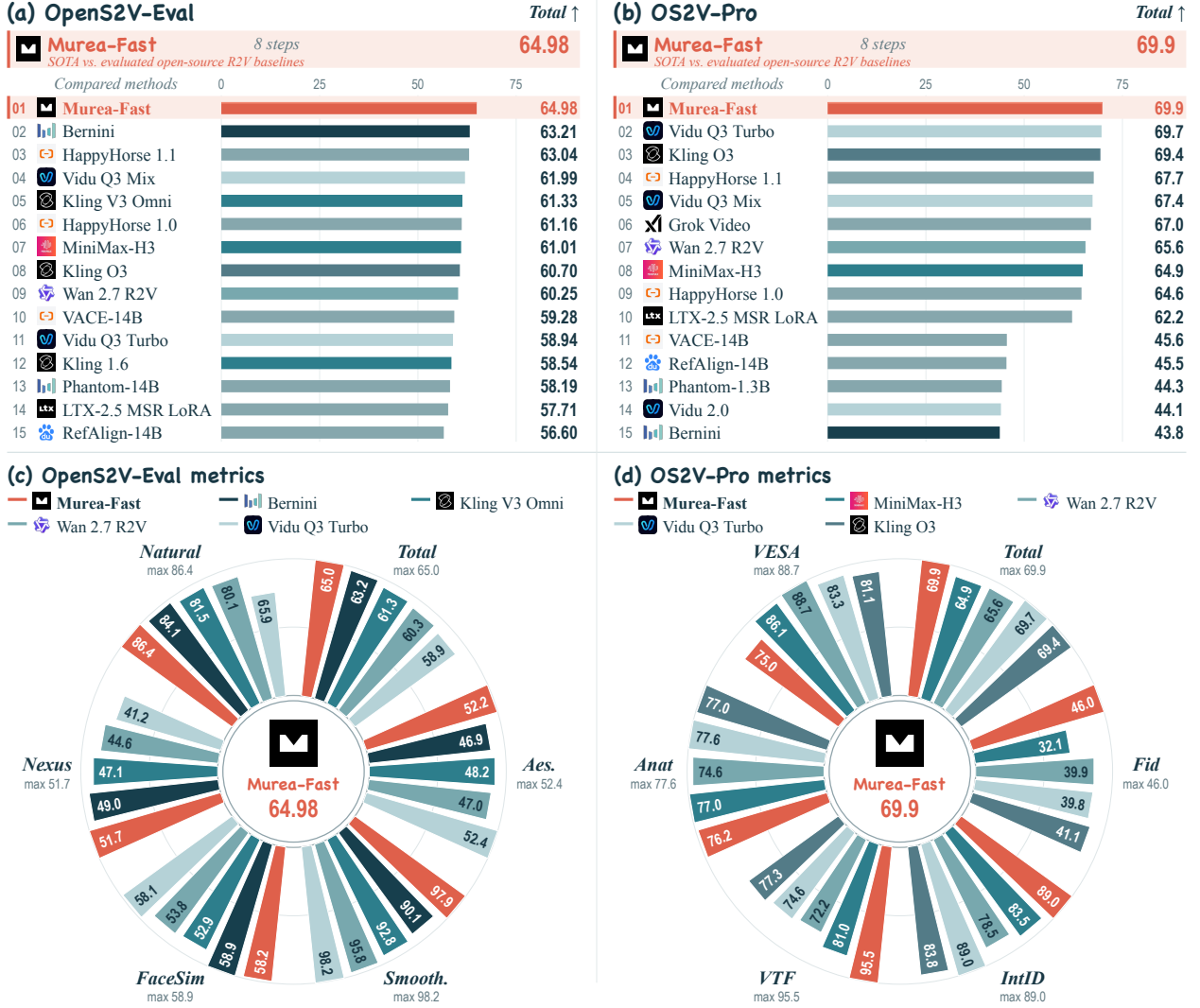


Figure 7. Comparison of Murea-Fast with representative open-source and commercial systems on OpenS2V-Eval (left) and OS2V-Pro (right). Scores correspond to Tables 1 and 2; aggregate scores retain the reported precision, while values in the circular plots are rounded to one decimal place. The right panel shows the OS2V-Pro aggregate as Total, corresponding to the Pro column in Table 2. Using only eight sampling steps, Murea-Fast achieves state-of-the-art aggregate performance among the evaluated open-source reference-to-video systems on both benchmarks and outperforms most evaluated commercial systems, including HappyHorse 1.1 and Kling O3, under the reported evaluation settings. The expanded-prompt protocol used for Murea-Fast is described in the text and affects the VTF/Bind comparison.

Comparison methods. We compare Murea-Fast with 12 commercial and 11 open-source configurations. The commercial baselines include Seedance 2.0 [39], HappyHorse [1], Vidu [42], Kling [16–18], Wan 2.7 R2V [2], Grok Video, and Pika 2.1 [34]. The open-source baselines include Bernini [3], MiniMax-H3 [28], VACE [14], Phantom [24], MAGREF [7], SkyReels-A2 [8], RefAlign, and LTX-2.5 [22] with the LiconStudio multi-subject-reference LoRA. The tables specify the evaluated variants of each method.

Experimental Settings. Murea-Fast uses EMA weights and eight distilled sampling steps to generate 121 frames at 24 fps, with a resolution of either 1280×704 or 1248×736 depending on the sample. Baselines retain their available inference configurations. The LTX-2.5 reference-to-video baseline combines the Dev checkpoint with the LiconStudio MSR LoRA and uses 30 sampling steps with CFG 3.0.

For generation, Murea-Fast uses the ground-truth-caption prompt variant (`gt_caption`), with expanded descriptions of reference appearance, shot-level actions, camera motion, and sound. We query Kling V3 Omni, HappyHorse 1.1, Grok Video, Vidu Q3 Turbo, and Wan 2.7 R2V through public APIs using the official prompts and also include released baseline clips. For evaluation, OpenS2V-Eval scores every system against the official prompts through a common pipeline. The comparisons therefore reflect both model and prompt-construction differences. OS2V-Pro derives its VTF and Bind checklists from the expanded prompts provided to Murea-Fast. These checklists contain details absent from the official prompts, giving Murea-Fast an input advantage on the

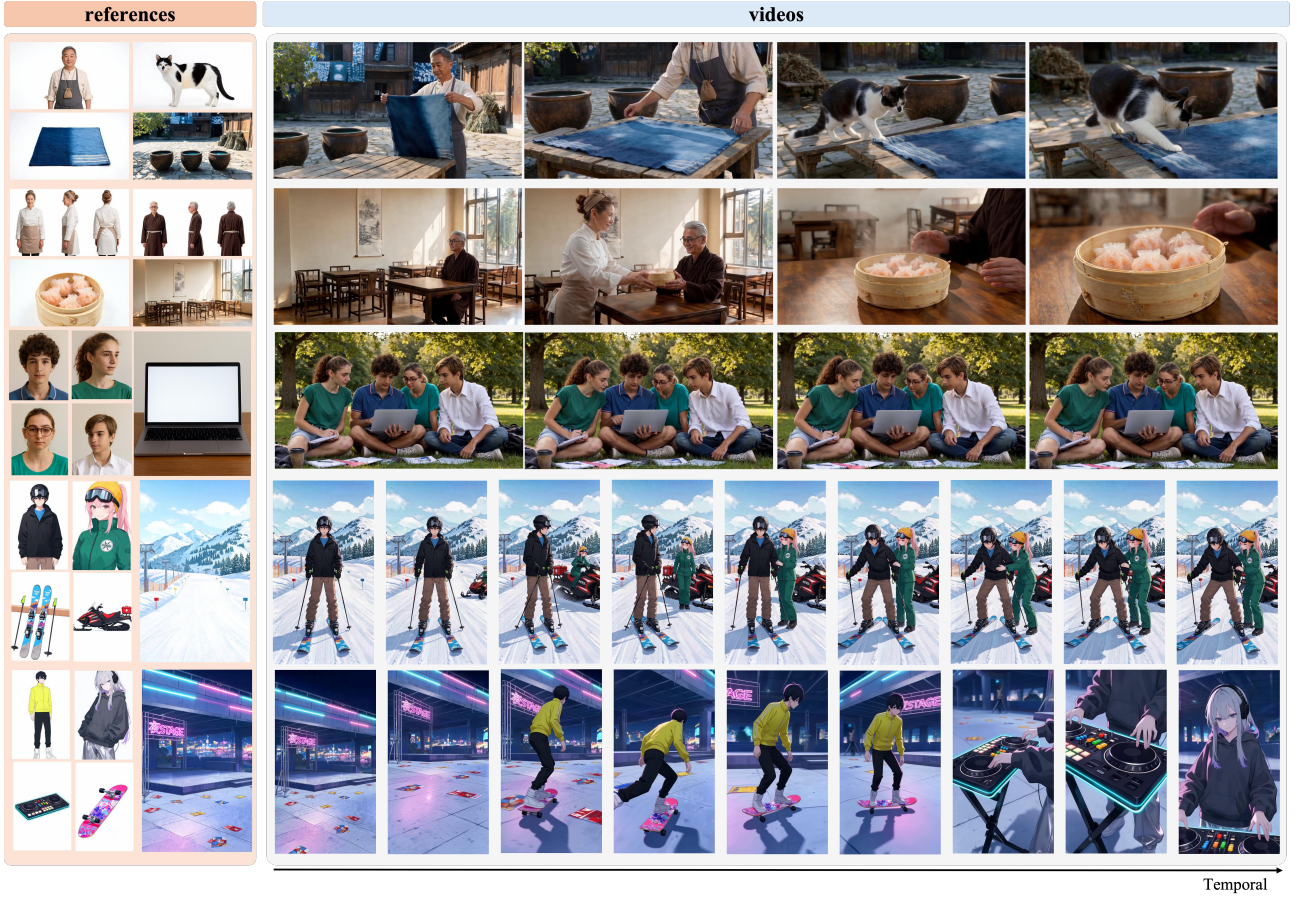


Figure 8. Qualitative results of reference-conditioned generation. Each row pairs input visual references on the left with sampled frames from the corresponding generated video on the right, ordered from left to right. The examples cover photorealistic and stylized settings with multiple subjects, objects, and scene references, illustrating reference preservation and compositional generation across time.

two metrics and affecting the OS2V-Pro aggregate. For RefAlign-14B, our evaluation yields a GmeScore of 59.66, compared with the previously reported 68.32; we use the recomputed value throughout.

We evaluate all available clips without post-hoc prompt or seed selection. Table 1 reports the clip count N for each system; individual metrics use their applicable subsets of successfully scored clips. Missing outputs are not assigned zero scores, so comparisons need to be read alongside coverage. Seedance 2.0 is included in OpenS2V-Eval, with its coverage discussed alongside the results below. The OS2V-Pro comparison comprises the systems in Table 2 and does not include Seedance 2.0.

We measure local inference latency on eight H20 GPUs after model loading. LTX pipelines with components loaded on demand are warmed up before timing, and Murea-Fast reuses cached text and reference encodings. API latency includes queueing and request completion. These measurements characterize each deployment and are excluded from both aggregate scores.

Quantitative Results on OpenS2V-Eval. Murea-Fast achieves a higher Total score than all evaluated open-source baselines and several commercial systems (Table 1). Its score of 64.98 improves on Bernini by 1.77 points and MiniMax-H3 by 3.97 points, and exceeds the reported scores of HappyHorse 1.1 (63.04), Vidu Q3 Mix (61.99), and Kling V3 Omni (61.33). Seedance 2.0 obtains the highest reported Total of 67.13, but returns 148 outputs with 32 refusals in face- and human-related categories. On these 148 shared cases, Murea-Fast scores 66.11 versus 67.13 for Seedance 2.0, a gap of 1.02 points under the same aggregation rule. This comparison accounts for clip coverage, while each component metric still uses its applicable subset.

Reference preservation is the clearest advantage across the component scores. Murea-Fast attains the highest NexusScore (51.66) among the ranked reference-to-video systems, exceeding Bernini (48.96). Against MiniMax-H3, it improves NexusScore by 10.61 points and FaceSim by 17.77 points. MiniMax-H3 scores higher on smoothness and naturalness, illustrating that these visual-quality measures capture different properties from fidelity to the supplied references. The two identity measures also favor Murea-Fast over Kling V3 Omni and

Table 3. Quantitative comparison on WBench Full-289 (289 cases and 1,058 interaction turns). All scores are on a 0–100 scale; higher is better. Overall is the unweighted mean of the five dimension scores. The first row reports our model configuration. Public baseline scores are taken from a fixed WBench leaderboard snapshot [27]. Bold and underline denote the highest and second-highest scores in each column, respectively.

Model	Overall	Quality	Setting	Interaction	Consistency	Physical
Murea-Pro	81.13	79.97	86.48	82.60	82.16	74.42
Kling 3.0	<u>79.42</u>	80.02	<u>90.96</u>	<u>73.08</u>	83.86	69.19
Wan 2.7	78.37	<u>80.99</u>	91.45	72.09	75.77	<u>71.57</u>
H3 FL2VA	77.24	79.15	86.22	71.05	83.09	66.68
Seedance 1.5	76.27	81.94	82.94	68.34	79.91	68.24
LTX 2.5	74.80	79.11	87.91	56.93	80.81	69.26
HY-Video 1.5	74.30	76.61	85.56	54.70	87.48	67.15
Cosmos3-Super	72.44	74.68	89.40	44.90	84.30	68.90
LTX 2.3	70.96	77.05	85.23	49.36	78.05	65.13
Cosmos 2.5	70.35	71.67	83.34	43.47	86.28	67.01
LongCat-Video	69.91	77.17	72.32	45.05	<u>86.62</u>	68.42
Cosmos3-Nano	69.28	74.98	86.60	36.50	84.45	63.85
YUME 1.5	68.93	77.60	72.38	48.38	80.85	65.42
Kairos 3.0	65.73	73.09	70.33	41.64	83.16	60.46

HappyHorse 1.1, with NexusScore gains of 4.60 and 8.41 points, respectively. Its FaceSim of 58.21 exceeds every evaluated configuration from HappyHorse, Kling, and Vidu.

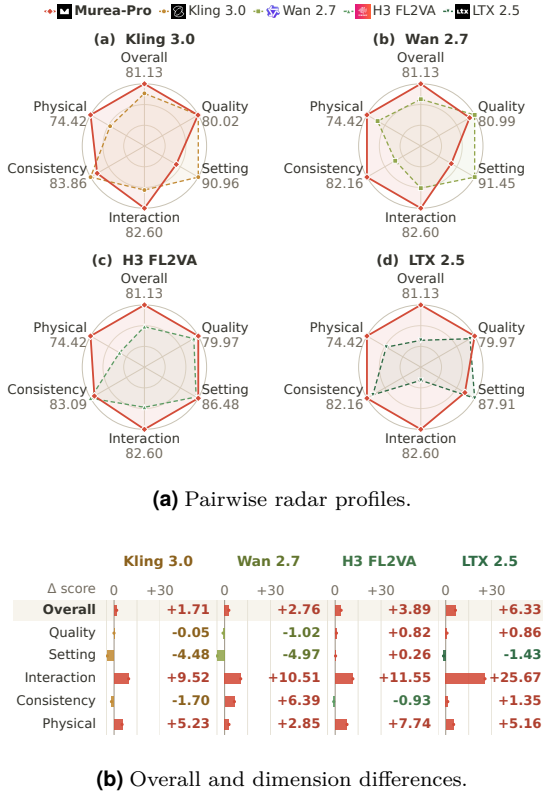
The higher detected-subject similarity relative to Bernini is accompanied by improvements in video quality, although Bernini retains a slightly higher FaceSim (58.93 versus 58.21). Murea-Fast obtains higher aesthetics (52.20 versus 46.89), motion smoothness (97.91 versus 90.11), and naturalness (86.44 versus 84.12). It also improves all three metrics over Kling V3 Omni, by 3.96, 5.13, and 4.90 points, respectively. These comparisons associate the higher subject-similarity scores with gains in both the appearance and temporal quality of the generated clips. Against HappyHorse 1.1, motion smoothness is nearly unchanged (97.91 versus 97.84), while naturalness increases from 83.95 to 86.44. Together with the higher FaceSim and NexusScore, this comparison locates the advantage in subject preservation and naturalness while maintaining comparable motion smoothness.

Quantitative Results on OS2V-Pro. Murea-Fast achieves an OS2V-Pro aggregate of 69.9, close to Kling V3 Omni (70.1) and above Vidu Q3 Turbo (69.7) and HappyHorse 1.1 (67.7), as shown in Table 2. It also exceeds MiniMax-H3 (64.9), the highest-scoring open-source reference-to-video baseline. Its strongest component results concern reference fidelity and identity consistency. Murea-Fast achieves the highest per-reference fidelity (46.0) among the ranked reference-to-video systems and shares the highest intra-clip identity consistency (89.0) with Vidu Q3 Turbo. At this matched level of intra-clip consistency, its fidelity exceeds Vidu Q3 Turbo’s 39.8. Both identity measures improve over Bernini, which scores 44.8 and 81.7, and Kling V3 Omni, which scores 41.9 and 84.7, respectively. These comparisons distinguish resemblance to the supplied references from identity stability within a generated sequence and show advantages on both measurements.

The reference-preservation results are accompanied by improvements in several rendering measures. Murea-Fast scores 21.6 on technical video quality, compared with 19.1 for Kling V3 Omni and 20.0 for Vidu Q3 Turbo. Its usable-clip rate also exceeds Kling V3 Omni’s (91.1 versus 88.0). Against Bernini, technical quality improves from 19.1 to 21.6 and anatomical consistency from 62.6 to 76.2. The comparison with Bernini thus combines higher fidelity and identity consistency with improvements in technical quality and anatomical plausibility.

The instruction metrics separate task following from attribute binding. Murea-Fast obtains the highest VTF score of 95.5 under the expanded-prompt protocol, whereas its Bind score is 19.6, below Kling V3 Omni’s 22.6. Thus, the VTF ranking does not extend to attribute binding. The checklists use details supplied in Murea-Fast’s expanded generation prompts, so the VTF comparison reflects the input advantage described above as well as the generated outputs.

Audio remains a weaker aspect of this configuration: its ATQ of 59.7 and VESA of 75.0 are below Kling V3 Omni’s 66.3 and 88.7, respectively. These differences are not reflected in the audio group of the aggregate. Its three-reference to one-reference fidelity ratio is 0.58; because the two groups contain different cases, this is a descriptive comparison rather than a controlled ablation of reference count.



Metric	Murea-Pro	Kling 3.0	Wan 2.7	H3 FL2VA	LTX 2.5
Quality					
Aesthetic quality	60.83	61.32	59.58	61.32	62.09
Imaging quality	67.67	67.66	68.06	67.77	61.20
Temporal flickering	93.77	94.50	93.04	94.52	96.41
Dynamic degree	89.62	89.93	99.31	82.35	88.24
Motion smoothness	97.98	97.90	96.55	98.06	98.95
HPSv3 quality	69.96	68.81	69.43	70.85	67.79
Setting					
Scene adherence	83.86	89.04	88.32	82.53	82.66
Subject adherence	89.10	92.88	94.58	89.91	93.15
Interaction					
Navigation trajectory	71.79	70.32	65.97	69.56	62.31
Event edit adherence	90.92	81.40	84.00	85.52	65.02
Subject action adherence	87.04	85.60	83.40	79.11	68.14
Perspective switch adherence	80.65	55.00	55.00	50.00	32.26
Consistency					
Background consistency	90.71	92.71	89.48	90.51	89.04
Segment continuity	96.54	92.71	65.62	97.23	86.85
Perspective consistency	68.46	76.75	62.18	72.58	66.21
Subject consistency	88.96	88.45	88.73	89.98	86.98
Geometric consistency	87.06	89.44	82.63	88.11	90.62
Photometric consistency	77.02	80.40	75.49	77.94	84.38
Spatial consistency	74.25	75.25	71.00	74.33	71.40
Gated spatial consistency	74.25	75.14	71.00	74.04	71.01
Physical					
Visual plausibility	60.19	60.37	59.84	60.01	59.31
Causal fidelity	88.65	78.00	83.30	73.34	79.20

Figure 9. Comparison on WBench Full-289. (a) Radar minima are the five-model minima rounded down to multiples of 10; axis labels show pairwise maxima. (b) Murea-Pro minus each baseline in score points, using a shared scale. (c) Scores on 22 metrics; bold, underlined values indicate the best score. Higher is better throughout.

Murea-Fast attains these identity and quality scores with the eight-step EMA checkpoint used for the latency evaluation. It generates a 5.04-second clip in 43.7 seconds on eight H20 GPUs, compared with 494.2 seconds for MiniMax-H3 and 527.5 seconds for Bernini, also using eight H20 GPUs. This combines a higher OpenS2V-Eval aggregate than both baselines with lower measured generation latency. The timings characterize the tested deployments, including their respective output settings and implementations. Figure 7 summarizes the aggregate scores and selected metric-level results across the two benchmarks.

Qualitative results. Figure 8 presents qualitative examples of multi-reference-conditioned video generation across photorealistic and stylized scenes. The examples cover diverse reference categories, including humans, objects, animals, and scenes, illustrating the model’s ability to combine heterogeneous visual inputs within a single video. The generated content preserves recognizable characteristics of the referenced elements, with their correspondence to the inputs visible across successive frames. The examples also span different reference formats, including single-view images, three-view reference sheets, and other layouts, demonstrating flexibility in using visual information presented in different forms.

Beyond this diversity in reference categories and formats, the figure includes examples conditioned on four or five reference images. These cases illustrate the model’s ability to incorporate several references simultaneously, bringing the corresponding subjects, objects, and environments together in the generated scenes. Their visual correspondence to the supplied references remains evident across the displayed frames, extending the qualitative demonstration to settings with more reference inputs. Together, these results highlight the versatility of our model across reference categories, formats, and counts, supporting a broad range of multi-reference video generation scenarios.

5.4 WBench

Benchmark. WBench [48] evaluates interactive video generation under sequential instructions, assessing both the execution of requested changes and the preservation of visual consistency across turns. We use the Full-289 split, which comprises 289 test cases and 1,058 interaction turns covering four task categories: navigation, subject action, event editing, and perspective switching.

In our evaluation, each case is initialized with the benchmark-provided image. For each subsequent turn, the model receives the current instruction and the final generated frame from the preceding segment. The generated segments are concatenated and evaluated as a complete interaction sequence. This protocol assesses successive instructions under the visual context produced by the model itself.

We report the five WBench dimensions: video quality (Quality), adherence to scene and subject specifications (Setting), instruction execution (Interaction), visual consistency (Consistency), and physical plausibility and causal fidelity (Physical). Each metric is averaged over the applicable cases, and each dimension is computed as the unweighted mean of its constituent metrics. Overall is the unweighted mean of the five dimension scores. All scores are reported on a 0–100 scale, with higher values indicating better performance.

Comparison methods. We compare against Kling 3.0 [18], Wan 2.7 [2], MiniMax-H3 with FL2VA conditioning (H3 FL2VA) [28], Seedance 1.5 [40], HunyuanVideo 1.5 [43], LongCat-Video [26], Yume 1.5 [25], and Kairos 3.0 [15]. We also include the LTX family (LTX-2.3 and LTX-2.5) [11, 21, 22], and the Cosmos family (Cosmos 2.5, Cosmos3-Super, and Cosmos3-Nano) [30–32]. Public baseline scores are taken from a fixed WBench leaderboard snapshot [27]. Each baseline retains its reported inference configuration; the results therefore compare complete systems under their respective settings. We report Murea-Pro with 50 sampling steps. This configuration uses I2VA for first-frame conditioning and constructs the inference-time prompt from the current image and a pre-existing structured prompt. It generates video at 1344×768 and 24 fps, with a target duration of five seconds per turn and a fixed video-generation seed of 42.

Quantitative results. Table 3 summarizes the Full-289 results, while Figure 10 shows the corresponding official leaderboard snapshot. Murea-Pro achieves the highest Overall score among the compared systems (81.13), exceeding the highest-scoring baseline, Kling 3.0 (79.42), by 1.71 points. This advantage is concentrated in Interaction (82.60 versus 73.08; +9.52 points) and Physical (74.42 versus 69.19; +5.23 points). Its Quality score is similar to that of Kling 3.0 (79.97 versus 80.02), while its Setting and Consistency scores are lower by 4.48 and 1.70 points, respectively.

Figure 9 compares Murea-Pro with four baselines across five evaluation dimensions and 22 metrics. The pairwise radar plots in Figure 9(a) show the relative strengths and weaknesses across the five dimensions, with the largest relative gains appearing on Interaction and Physical. The dimension-difference plots in Figure 9(b) make this pattern explicit: Murea-Pro is ahead on Overall for every comparison model, with gains of 1.71, 2.76, 3.89, and 6.33 points over Kling 3.0, Wan 2.7, H3 FL2VA, and LTX 2.5, respectively. Interaction is the largest positive difference in all four comparisons, whereas Setting is the main dimension in which Murea-Pro trails the two strongest setting-oriented baselines, Kling 3.0 and Wan 2.7.

The metric-level heatmap in Figure 9(c) resolves these dimension averages into the 22 constituent metrics. Murea-Pro leads all four comparison models on navigation trajectory, event-edit adherence, subject-action adherence, perspective-switch adherence, and causal fidelity. Across the 22 metrics, it scores higher than Wan 2.7 on 18 metrics, LTX 2.5 on 16, and Kling 3.0 and H3 FL2VA on 10 each. The remaining gaps are concentrated in image and motion quality, scene and subject adherence, and several consistency measures, which explains why the strong Interaction and Physical scores translate into the best Overall score despite smaller deficits in Quality, Setting, and Consistency.

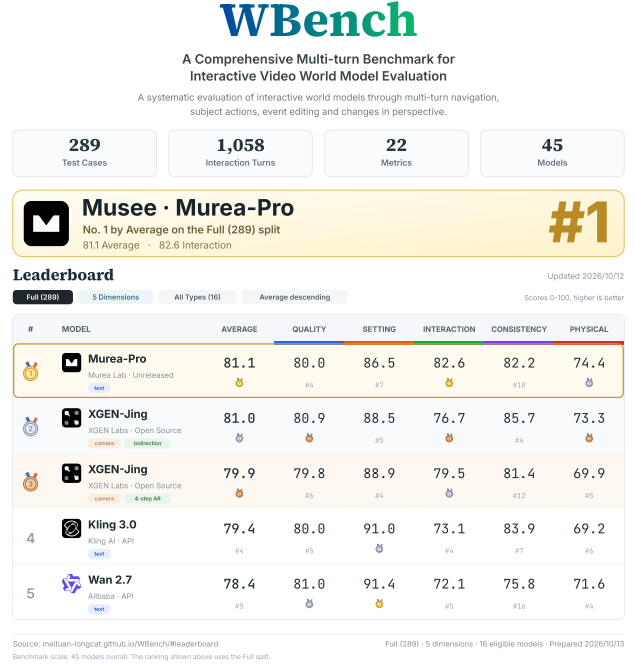


Figure 10. Official WBench Full-289 leaderboard, updated October 12, 2026. Scores are rounded to one decimal; Table 3 gives the full-precision values. Source: WBench leaderboard [27].

At the task level, Murea-Pro scores 90.92 on event editing, 87.04 on subject action, 80.65 on perspective switching, and 71.79 on navigation. It therefore performs better on event editing and subject action than on perspective switching and navigation. Navigation yields its lowest task-level score, identifying a remaining limitation in multi-turn instruction following.

6 Discussion

InWorld couples persistent world state with audiovisual generation through a state–action–observation loop. Murea-Mind tracks registered assets, dynamic changes, and alternative histories, compiles plans into Murea conditions, and uses generated observations to ground later turns and shared or branching interaction. Murea jointly generates video and audio from text and clean visual or voice references; target-only flow matching, sequential MeanFlow-to-DMD distillation, custom FlashAttention-4 JVP computation, and degraded-history streaming training provide a practical path from faithful generation to few-step, long-horizon rollout. Murea-Pro reaches an Overall score of 81.13 on WBench Full-289 and surpasses most evaluated commercial systems, while Murea-Fast reaches a Total score of 64.98 on OpenS2V-Eval and an aggregate score of 69.9 on OS2V-Pro achieves aggregate open-source SOTA on both reference-to-video benchmarks, and also surpasses most evaluated commercial systems. Navigation, long-range planning, and setting consistency remain the main areas for improvement.

7 Authors

Core Contributors

Linux^{4,5}, Alex^{3,4}, Yi Fu^{1,3,4}, North Ge⁵, Jun Huang^{1,2}, Yuankang Shi^{1,2}, Lingfei Li^{1,2}, Ryan⁶, Captain^{6,✉}

Listed by task: ¹ Agent, ² Bench, ³ Data, ⁴ Training, ⁵ Infra, ⁶ Project Lead, [✉] Corresponding Author

Contributors

Bin Chen, Chenyang Yin, Haodong Yuan, Junyi Mao, Lu Fan, Rui Lan, Shaobin Gao, Weijie Xue, Xiwei Zheng, Zifeng Xie

Listed in alphabetical order.

Project Sponsors

Jianqi Zhang, Xu Chen

References

- [1] Alibaba Cloud. HappyHorse: Reference-to-video API reference, 2026.
- [2] Alibaba Cloud. Video generation: Wan model documentation, 2026.
- [3] Bernini Team, C. Liu, J. Chen, L. Li, L. Chi, M. Sun, Z. Li, Y. Fu, R. Guo, Y. Wu, G. Bai, and Z. Yuan. Bernini: Latent semantic planning for video diffusion. *arXiv preprint arXiv:2605.22344*, 2026.
- [4] L. Chen, T. Ma, J. Liu, B. Li, Z. Chen, L. Liu, X. He, G. Li, Q. He, and Z. Wu. Humo: Human-centric video generation via collaborative multi-modal conditioning, 2025.
- [5] N. Cudlenco, M. Masala, and M. Leordeanu. Authoring for living worlds: Tool-constrained LLM agents for executable multi-actor scenarios. *arXiv preprint arXiv:2604.10383*, 2026.
- [6] J. Deng, J. Guo, N. Xue, and S. Zafeiriou. ArcFace: Additive angular margin loss for deep face recognition. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [7] Y. Deng, X. Guo, Y. Yin, J. Z. Fang, Y. Yang, Y. Wang, S. Yuan, A. Wang, B. Liu, H. Huang, and C. Ma. MAGREF: Masked guidance for any-reference video generation. *arXiv preprint arXiv:2505.23742*, 2025.
- [8] Z. Fei, D. Li, D. Qiu, J. Wang, Y. Dou, R. Wang, J. Xu, M. Fan, G. Chen, Y. Li, and Y. Zhou. SkyReels-A2: Compose anything in video diffusion transformers. *arXiv preprint arXiv:2504.02436*, 2025.
- [9] Z. Gao, Q. Wang, J. Zhu, J. Chen, Z. Liu, et al. Infinite Worlds with Versatile Interactions. *arXiv preprint arXiv:2607.07534*, 2026.
- [10] Z. Geng, M. Deng, X. Bai, J. Z. Kolter, and K. He. Mean flows for one-step generative modeling. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*, 2025.
- [11] Y. HaCohen, B. Brazowski, N. Chiprut, Y. Bitterman, A. Kvochko, et al. LTX-2: Efficient joint audio-visual foundation model. *arXiv preprint arXiv:2601.03233*, 2026.
- [12] J. Huang, Y. Zhang, B. Xia, J. Wang, X. He, Z. Tang, M. Chu, X. Tao, P. Wan, and J. Jia. UnityShots: Memory-Driven Multi-Shot Audio-Video Generation with Boundary-Aware Gating. *arXiv preprint arXiv:2606.21661*, 2026.
- [13] S. A. Jacobs, M. Tanaka, C. Zhang, M. Zhang, S. L. Song, S. Rajbhandari, and Y. He. DeepSpeed Ulysses: System optimizations for enabling training of extreme long sequence transformer models. *arXiv preprint arXiv:2309.14509*, 2023.
- [14] Z. Jiang, Z. Han, C. Mao, J. Zhang, Y. Pan, and Y. Liu. VACE: All-in-one video creation and editing. *arXiv preprint arXiv:2503.07598*, 2025.
- [15] Kairos Contributors. Kairos: Official world model repository, 2025.
- [16] Kling AI. Kling VIDEO 3.0 Omni model user guide, 2026.
- [17] Kuaishou Technology. Kuaishou Kling AI unveils multi-image reference feature to further tackle AI video consistency challenges, 2025.
- [18] Kuaishou Technology. Kling AI launches 3.0 model, ushering in an era where everyone can be a director, 2026.
- [19] Y. Lai, T. Shao, W. Dou, S. Zhu, and J. Wang. GroundShot: Visually Consistent Multi-Shot Long Video Generation via Entity-Grounded Shot Scheduling. *arXiv preprint arXiv:2606.20799*, 2026.
- [20] LAION. Laion-aesthetics predictor. Official model repository, 2022.
- [21] Lightricks. LTX-2.3: Official model card, 2026.
- [22] Lightricks. LTX-2.5: Official model card, 2026.
- [23] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

- [24] L. Liu, T. Ma, B. Li, Z. Chen, J. Liu, G. Li, S. Zhou, Q. He, and X. Wu. Phantom: Subject-consistent video generation via cross-modal alignment. *arXiv preprint arXiv:2502.11079*, 2025.
- [25] X. Mao, Z. Li, C. Li, X. Xu, K. Ying, T. He, J. Pang, Y. Qiao, and K. Zhang. Yume-1.5: A text-controlled interactive world generation model. *arXiv preprint arXiv:2512.22096*, 2025.
- [26] Meituan LongCat Team, X. Cai, Q. Huang, Z. Kang, H. Li, S. Liang, L. Ma, S. Ren, X. Wei, R. Xie, and T. Zhang. LongCat-Video technical report. *arXiv preprint arXiv:2510.22200*, 2025.
- [27] Meituan LongCat Team and Fudan University. WBench: Official project page and leaderboard, 2026.
- [28] MiniMax. MiniMax H3: An open model breaking the boundaries between tasks and modalities, 2026.
- [29] S. Mo, Y. Li, Z. Leng, K. K. Singh, and B. Zhou. Streaming multi-agent autoregressive diffusion model with world state registers. *arXiv preprint arXiv:2607.21594*, 2026.
- [30] NVIDIA. Cosmos-Predict2.5: Official model repository, 2025.
- [31] NVIDIA. Cosmos3-Nano: Official model card, 2026.
- [32] NVIDIA. Cosmos3-Super: Official model card, 2026.
- [33] OpenAI. GPT-4o: Model documentation, n.d. Model snapshot used in the evaluation: gpt-4o-2024-11-20.
- [34] Pika. Pika: Frequently asked questions, n.d.
- [35] A. Polyak, A. Zohar, A. Brown, A. Tjandra, A. Sinha, et al. Movie Gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.
- [36] T. Souček, J. Moravec, and J. Lokoč. TransNet: A deep network for fast detection of common shot transitions. *arXiv preprint arXiv:1906.03363*, 2019.
- [37] J. Su, Y. Lu, S. Pan, A. Murtadha, B. Wen, and Y. Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.
- [38] Team Odyssey. Starchild-1: A real-time multimodal world model. Technical report, Odyssey, 2026.
- [39] Team Seedance, D. Chen, L. Chen, X. Chen, et al. Seedance 2.0: Advancing video generation for world complexity. *arXiv preprint arXiv:2604.14148*, 2026.
- [40] Team Seedance, H. Chen, S. Chen, X. Chen, et al. Seedance 1.5 pro: A native audio-visual joint generation foundation model. *arXiv preprint arXiv:2512.13507*, 2025.
- [41] The Gemini Team. Gemini 3.1 Pro: A smarter model for your most complex tasks, 2026.
- [42] Vidu. Vidu: Reference to video API documentation, n.d.
- [43] B. Wu, C. Zou, C. Li, D. Huang, F. Yang, et al. HunyuanVideo 1.5 technical report. *arXiv preprint arXiv:2511.18870*, 2025.
- [44] H. Wu, E. Zhang, L. Liao, C. Chen, J. Hou, A. Wang, W. Sun, Q. Yan, and W. Lin. Exploring video quality assessment on user generated contents from aesthetic and technical perspectives. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.
- [45] T. Yin, M. Gharbi, T. Park, R. Zhang, E. Shechtman, F. Durand, and W. T. Freeman. Improved distribution matching distillation for fast image synthesis. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, 2024.
- [46] T. Yin, M. Gharbi, R. Zhang, E. Shechtman, F. Durand, W. T. Freeman, and T. Park. One-step diffusion with distribution matching distillation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.
- [47] Y. Yin, G. Wang, Y. Zhan, C. Li, K. Zhang, and F. Zhao. Alaya-EVOKE: From linear-scaling supervision to endless world. *arXiv preprint arXiv:2608.13546*, 2026.

- [48] K. Ying, H. Hu, S. Ren, J. Li, F. Chen, Z. Wang, X. Cao, X. Cai, and H. Ding. WBench: A comprehensive multi-turn benchmark for interactive video world model evaluation. *arXiv preprint arXiv:2605.25874*, 2026.
- [49] S. Yuan, X. He, Y. Deng, Y. Ye, J. Huang, B. Lin, C. Ma, J. Luo, and L. Yuan. OpenS2V-Nexus: A detailed benchmark and million-scale dataset for subject-to-video generation. *arXiv preprint arXiv:2505.20292*, 2025.
- [50] S. Yuan, Y. Yin, Z. Li, X. Huang, X. Yang, and L. Yuan. Helios: Real real-time long video generation model. *arXiv preprint arXiv:2603.04379*, 2026.
- [51] T. Zadouri, M. Hoehnerbach, J. Shah, T. Liu, V. Thakkar, and T. Dao. FlashAttention-4: Algorithm and kernel pipelining co-design for asymmetric hardware scaling. *arXiv preprint arXiv:2603.05451*, 2026.
- [52] J. Zhang, J. Wei, P. Zhang, J. Zhu, and J. Chen. SageAttention: Accurate 8-bit attention for plug-and-play inference acceleration. In *International Conference on Learning Representations (ICLR)*, 2025.
- [53] S. Zhu, Q. Peng, Z. Pu, Z. Shu, X. Ke, Z. Xing, Z. Tong, Z. Wang, X. Cui, H. Wang, J. Zhao, Y. Jin, F. Cheng, and R. Feng. Incantation: Natural language as the action interface for multi-entity video world models. *arXiv preprint arXiv:2605.18601*, 2026.