Waypoint-1.5: A Real-Time Video World Model for Consumer Hardware

Rajit Rajpal* **Shahbuland Matiana*** **Liew Wei Pyn*** **Anmol Agarwal***
Ryan Craig Andrew Lapp Mithun Hunsur Sami BuGhanem Scottie Fox Aaron Sanders
Carson Poole Irene Park David Rossi Spencer Frazier Louis Castricato
Overworld
*Equal contribution
Correspondence: shahbuland@over.world

HuggingFace Project Page GitHub Demo

Abstract

We present **Waypoint-1.5**, a real-time diffusion world model for interactive video generation on consumer-grade hardware. Unlike general video diffusion models, interactive world models (iWMs) must respond to dense user controls under strict latency and throughput constraints. Waypoint-1.5 is pre-trained on 100,000 hours of diverse, control-aligned video game data across hundreds of games, and generates playable video conditioned on full keyboard and mouse input. The model includes two resolution variants that run across a wide spectrum of consumer hardware. To characterize this unique setting, we distinguish rendered FPS, latent FPS, and control rate. We describe the data pipeline, architecture, training methodology, and runtime system behind Waypoint-1.5. We evaluate interactivity through latency and throughput. We conclude with a discussion of the safety and ethical considerations unique to iWMs.

Figure 1: **Waypoint-1.5 teaser.** Two example autoregressive rollouts generated in real time on a single consumer GPU (top and bottom rows). The 1.28B parameter model generates high-resolution video conditioned on keyboard and mouse input.

Preprint.

1 Introduction

Recent advances in generative video models have shown that neural networks can synthesize coherent visual dynamics over space and time. Systems such as Sora ([14]), WAN ([23]), Hunyuan Video ([12]), LTX-Video ([9]), and similar models demonstrate impressive progress in visual quality, temporal consistency, and prompt adherence. However, these models are designed primarily for offline text-guided video generation. To this end, the natural evaluation axes are prompt adherence, visual fidelity, and temporal coherence. Common evaluation metrics are VideoBench ([15]) and FVD ([22]). Progress in this space demonstrates the strengths of video diffusion for high quality video synthesis, but it does not solve interaction.

There are several approaches toward interactive generation: asset generation (i.e. text-to-3D, gaussian splats), real-time text-to-video (i.e. LongLive ([24])), and action-conditioned world models. The final category consists of systems such as Genie ([3]) and Hunyuan-GameCraft ([12]). Generally, such systems require large models which can only run on large-scale/private inference infrastructure, rely on streamed inference for the end-user (cloud gaming), and are incapable of running at the framerates and latencies considered standard in video games.

Recent advances in generative video models have demonstrated that neural networks can learn rich, physically-grounded representations of visual environments. Models such as Sora ([5]), Genie ([3]), and subsequent work have shown that scaling diffusion transformers on video data yields systems capable of generating coherent visual scenes, and, in some cases, interactive worlds that respond to user input [26] [16] [12] [20] [21] [17] [1]. These results point toward a future in which learned world models serve as the foundation for interactive entertainment, simulation, and embodied AI.

However, a significant gap remains between research demonstrations and practical deployment. Existing world models typically require enterprise level inference infrastructure, generate at low frame rates, or produce short, non-interactive clips. For world models to become a practical tool in game development, creative applications, immersive entertainment, and real-time simulation, they must run on the hardware that end users actually own.

We present Waypoint-1.5, a video world model built from the ground up for real-time inference on consumer GPUs. Our inference architecture is designed around a single-stream causal Diffusion Transformer with frame-causal attention and grouped query attention, paired with a distilled autoencoder optimized for decode throughput—choices that directly target the latency and throughput requirements of real-time interactive generation on consumer hardware. Our training methodology is designed for efficiency and stability under limited compute: diffusion forcing pretraining enables autoregressive rollout, NorMuon optimization improves convergence for large-scale transformer training, and sequence packing maximizes GPU utilization. Together, these choices allow Waypoint-1.5 to excel and run in real time on a wide range of consumer GPUs.

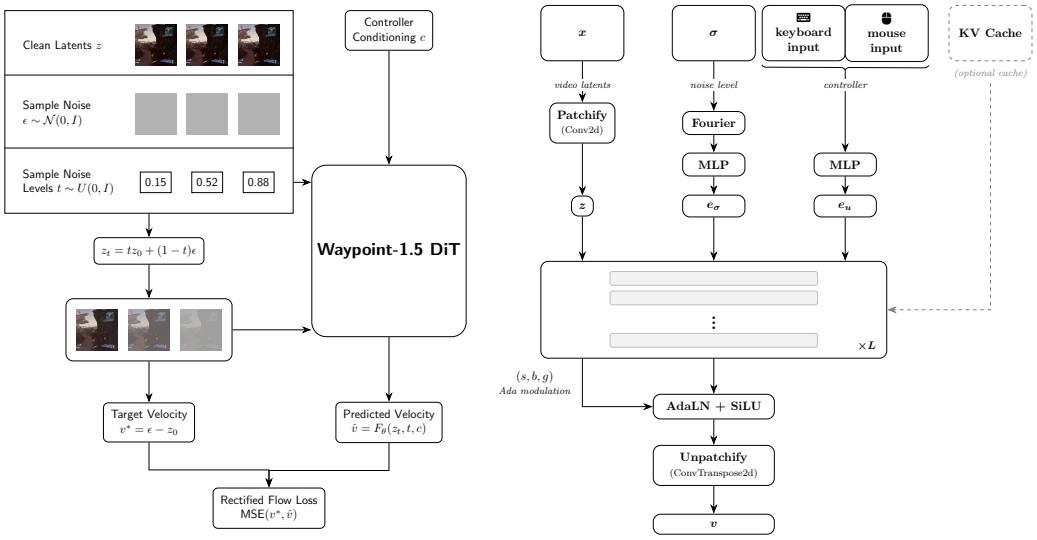
In this technical report, we describe the model architecture (Section 2), pretraining methodology and input conditioning (Section 3), post-training distillation (Section 4), inference system (Section 5), autoencoder design (Section 6), data pipeline (Section 7), evaluation (Section 8), and safety considerations (Section 9).

2 Architecture

Waypoint-1.5 is built on a **single-stream causal Diffusion Transformer (DiT)**, a 1.28 billion parameter model that processes all video latent frames in a clip as a unified sequence of tokens. Unlike dual-stream or cross-attention-based video DiTs that maintain separate pathways for conditioning and content, our architecture processes all tokens within a single transformer stream. The architecture is defined by three key design choices: frame-causal attention for autoregressive generation, grouped query attention with KV caching for inference efficiency, and weight-tied noise conditioning for parameter efficiency. The training pipeline and overall architecture are illustrated in Figure 2.

2.1 Spatial Tokenization

Each video frame in the TAEHV 1.5 latent space is spatially tokenized via a learned 2×2 convolutional projection, mapping the 32-channel latent to model dimension $d_{\text{model}} = 2048$. At 512×1024 pixels, this yields $16 \times 32 = 512$ tokens per frame; at 256×512 pixels, it yields $8 \times 16 = 128$ tokens per



(a) **Training Pipeline.** Per-frame independent noise levels under Diffusion Forcing training, with controller telemetry synchronized to each latent.

(b) **Waypoint-1.5 DiT.** Single-stream causal DiT with tied AdaLN noise conditioning and per-block MLPFusion controller injection.

Figure 2: **Diffusion Transformer Model.** (a) Training pipeline with per-frame Diffusion Forcing and synchronized controller inputs. (b) Single-stream causal DiT architecture with frame-causal attention.

frame. Frames are arranged to align with the block boundaries of the sparse attention kernel, ensuring spatially adjacent patches map to contiguous attention blocks for maximum kernel efficiency.

2.2 Frame-Causal Attention with KV Caching

Attention follows a **frame-causal** pattern implemented via FlexAttention with custom block masks (Figure 3). Within each frame, tokens attend bidirectionally to all other tokens in the same frame. Across frames, each frame attends only to tokens from past frames—never future frames—enabling autoregressive generation without architectural changes between training and inference.

To balance dense local context with long-range temporal coherence, we use a two-level attention hierarchy:

- **Local window**: each frame densely attends to the preceding 16 frames.
- **Global dilated window**: every 4th transformer block also attends to a set of evenly-spaced frames within a 128-frame context window, using a pinned dilation factor of 8. This sparse global attention efficiently covers long-range context without quadratic memory growth.

Static Rolling KV Cache. At inference time, a static ring-buffer KV cache stores the key-value representations of past frames. Crucially, only *clean* (fully denoised) frames are written to the cache; noisy intermediate frames during multi-step denoising are not cached. This ensures the autoregressive context always consists of high-quality reconstructed frames, improving long-horizon rollout stability.

2.3 Grouped Query Attention

We use Grouped Query Attention (GQA) [2] with 32 query heads and 16 key-value heads (2:1 ratio). This halves the KV memory footprint relative to standard multi-head attention, reducing both the KV cache size and memory bandwidth requirements during long-horizon autoregressive generation. Each head has dimension 64.

2.4 WAN-Style AdaLN Noise Conditioning

We perform noise conditioning following Wan [23]: the scalar noise level σ is encoded through a Fourier feature embedding (512-dimensional) passed through a small MLP to produce a conditioning vector. This vector is consumed by a `CondHead` module generating three modulation parameters per sub-layer—scale s, bias b, and gate g—used in adaptive layer normalization:

$$\mathrm{AdaLN}(x;\, s, b, g) = g \odot \mathrm{LN}(x) \cdot (1 + s) + b, \tag{1}$$

where LN denotes layer normalization. Critically, the `CondHead` weights are shared across all 24 transformer layers. This weight tying substantially reduces parameter count while maintaining effective noise conditioning throughout the network depth. The `CondHead` output projection is zero-initialized (AdaLN-Zero), ensuring identity-like behavior at the start of training. In addition, following WAN [23], a learnable per-layer bias is added to each conditioning layer's output, providing layer-specific offsets alongside the shared head.

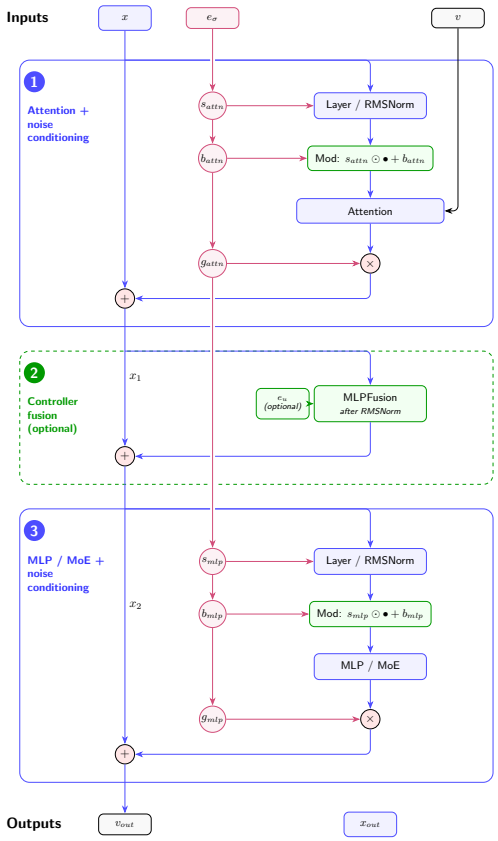


Figure 3: **Diffusion Transformer Block.** Each block applies frame-causal FlexAttention with a static rolling KV cache, WAN-style AdaLN noise conditioning, and MLPFusion controller input injection.

2.5 Positional Encoding: OrthoRoPE

Position is encoded via **OrthoRoPE**, an orthogonal variant of Rotary Position Embeddings [19] that applies independent rotations for spatial (x, y) and temporal (t) dimensions within a single per-head embedding. With head dimension 64, we allocate 8 dimensions per spatial axis and 16 dimensions to the temporal axis, providing distinct Nyquist-tuned frequency bands for spatial proximity and temporal ordering. Spatial frequencies are scaled to the video's spatial Nyquist frequency; temporal frequencies use a standard base of 10,000.

2.6 Controller Conditioning

At each transformer block, the model receives controller inputs: discretized button states (256-bucket encoding), continuous mouse displacement $(\Delta x, \Delta y)$, and scroll sign. These are concatenated and projected to d_{model} via a learned MLP, then fused into the block's intermediate representation through `MLPFusion`—a lightweight residual module:

$$x \leftarrow x + \mathrm{SiLU}(\mathrm{W}_1 x + \mathrm{W}_2 c) \cdot \mathrm{W}_3, \tag{2}$$

where c is the projected controller embedding. During training, controller conditioning is randomly dropped with probability p_{cfg} to support classifier-free guidance (CFG) at inference time.

2.7 Model Configuration

A summary of the model configuration is provided in Table 1.

3 Training

3.1 Dataset Details

Waypoint-1.5 is trained on **Owl-Control**, an internal dataset of over 100,000 hours of video game gameplay paired with synchronized controller telemetry. The dataset spans hundreds of distinct

Table 1: Waypoint-1.5 model configuration.

Parameter	Value
Total Parameters	1,281,959,992
Transformer Layers	24
Query Heads / KV Heads	32 / 16
Model Dimension (d_{model})	2048
MLP Expansion Ratio	$4\times$
Head Dimension	64
Input Resolution	512×1024 / 256×512 pixels
Latents per Clip	192
Tokens per Frame	512 / 128
Positional Encoding	OrthoRoPE
Noise Conditioning	WAN-style tied weights
Value Residual	Yes
Autoencoder	TAEHV 1.5

games across a variety of genres and visual styles. Raw video is captured at 720P resolution and 60 FPS and compressed offline using TAEHV 1.5 with a temporal compression factor of 4, yielding 15 latent frames per second. Controller inputs—keyboard button states, mouse displacement, and scroll events—are recorded at up to 10,000 Hz and synchronized to the video timeline. During preprocessing, controller inputs are aligned and downsampled to 15 buckets per second of latent video, matching the latent frame rate. Details of the preprocessing pipeline are described in Section 7.

3.2 Diffusion Forcing

We pretrain using **Diffusion Forcing** [6][18], a training objective that enables causal autoregressive rollout while preserving the expressiveness of a full diffusion model. Unlike standard video diffusion training, which adds a uniform noise level across all frames in a clip, Diffusion Forcing assigns an *independent* noise level σ_i to each frame i, sampled from a per-frame distribution over $[0, 1]$:

$$\sigma_i = \text{sigmoid}(\epsilon_i), \quad \epsilon_i \sim \mathcal{N}(0, 1), \quad i = 1, \ldots, N. \tag{3}$$

Given clean video latents $x \in \mathbb{R}^{B\times N\times C\times H\times W}$, the rectified-flow target velocity, noised input, and training objective are:

$$v^* = \varepsilon - x, \quad \varepsilon \sim \mathcal{N}(0, I), \tag{4}$$

$$x_t = x + v^* \cdot \sigma_i, \tag{5}$$

$$\hat{v} = F_\theta(x_t,\ \sigma,\ \mathbf{c}), \tag{6}$$

$$\mathcal{L} = \mathbb{E}\left[\|v^* - \hat{v}\|^2\right], \tag{7}$$

where $\mathbf{c}$ denotes all conditioning inputs (previous frames, controller). Training on frames with *heterogeneous, independently sampled* noise levels teaches the model to denoise each frame conditioned on all preceding frames at their current noise levels—directly mirroring the autoregressive inference regime, in which past frames are fixed at a low noise level ($\sigma_{\text{ctx}} \approx 0.15$) as clean context while the current frame is denoised from $\sigma = 1$ to $\sigma = 0$. It mitigates autoregressive drift by enforcing some corruption of previous latents during pretraining.

3.3 NorMuon

The 2D weight matrices (query, key, value, and output projections, and MLP weight matrices) are optimized with **NorMuon** [11][13]. We use a learning rate of 1×10^{-3} for NorMuon-updated parameters. All remaining parameters (embeddings, biases, normalization scalars) are updated with AdamW at a learning rate of 1×10^{-4}.

3.4 Input Conditioning

Waypoint-1.5 implements a learned transition function $p(z_t \mid z_{<t}, c_{<t})$, conditioning on previous TAEHV 1.5 latent frames and per-timestep controller inputs (button states, mouse displacement, scroll)

rather than free-form text. Controller inputs are fused into each transformer block via MLPFusion as described in Section 2; CFG dropout during training enables unconditional rollouts at inference. At deployment, user inputs are sanitized and mapped to the expected controller format before reaching the model, serving as an additional safety layer as described in Section 9.

4 Post-Training

Direct inference with the pretrained Waypoint-1.5 model faces two key limitations. First, it is computationally expensive: each frame requires solving a multi-step ODE (typically 60 solver steps), and classifier-free guidance doubles the effective step count by requiring independent forward passes for conditioned and unconditioned predictions, yielding an effective inference cost of $\sim$120 forward passes per frame of generated video. Second, the model exhibits *exposure bias*: it is trained on ground-truth context frames corrupted with varying noise, but at inference it must condition on its own imperfect predictions from prior steps—a training-inference mismatch that causes visual drift and incoherence over long rollouts.

We address both limitations with a custom **Self-Forcing DMD** post-training procedure, drawing on Distribution Matching Distillation (DMD) [25], Self-Forcing [10], and Self-Forcing++ [7]. In our procedure, a student model F_θ, a frozen teacher copy F_ϕ, and a trainable critic F_ψ are jointly optimized. Self-forcing trains the student to generate video autoregressively, conditioning exclusively on its own previously generated clean frames:

$$p_\theta(z_{1:T}) = \prod_{t=1}^{T} p_\theta(z_t \mid \hat{z}_{<t},\, c_{<t})\,, \tag{8}$$

where $\hat{z}_{<t}$ are clean self-generated latent frames and $c_{<t}$ are the corresponding controller inputs. This directly eliminates the training-inference context mismatch that causes rollout drift.

The teacher is evaluated under multi-condition CFG:

$$\hat{v}_{\text{teacher}} = v_\varnothing + \omega_{\text{ctrl}}(v_{\text{ctrl}} - v_\varnothing) + \omega_{\text{ctx}}(v_{\text{ctx}} - v_\varnothing)\,, \tag{9}$$

with $\omega_{\text{ctrl}} = 15.0$, $\omega_{\text{ctx}} = 2.0$, and $v_\varnothing$ the unconditioned prediction. The student learns to match this guided distribution directly, eliminating the need to compute $v_\varnothing$ at inference. The DMD loss [25] minimizes an approximate KL divergence between the student's denoising distribution and the teacher's guided distribution, using a trainable critic to provide the score gradient signal. Training stability is further supported by two regularisation terms from SenseFlow [8]: **IDA** (Implicit Distribution Alignment) keeps the critic parameters close to the student via an EMA update after each student step, preventing critic divergence; and **ISG** (Intra-Segment Guidance) enforces consistency between adjacent noise levels by matching student and teacher predictions at intermediate points along the denoising trajectory.

Sequence-Packed Teacher Forcing. To compute gradients over autoregressive sequences without costly per-frame backward passes, we use **sequence-packed teacher forcing**: context and current frames are concatenated into a single forward pass, with a binary mask (`curr_frame_mask`) distinguishing context-only tokens (`cid` $= 0$, no gradient) from gradient-carrying tokens (`cid` $= 1$). A custom attention rule prevents leakage between the two copies of the same frame: context tokens attend only to context; current-frame tokens attend only to their own same-frame tokens; and current-frame tokens are explicitly blocked from attending to the context copy of the same frame. Concurrently and independently, [17] developed the same sequence-packing strategy.

The distillation procedure achieves an approximately 30$\times$ total inference speedup relative to the pretrained model. Baking in CFG eliminates the 2$\times$ per-step overhead, reducing the effective forward pass count from 120 to 60 (a $\sim$2$\times$ improvement). Reducing the denoising step count from 60 to 4 provides a further $\sim$15$\times$ improvement. The resulting 4-step model generates high-quality frames conditioned on a rolling window of clean self-generated context, enabling stable long-horizon autoregressive rollout.

5 Inference

5.1 Autoregressive Generation and KV Caching

Waypoint-1.5 generates video autoregressively, producing one latent frame at a time. At each step, the model receives a single noisy frame and all controller inputs for that frame, and denoises it conditioned on the full history of past clean frames. Rather than re-encoding the entire history at every step, we use a static rolling KV cache that stores key-value representations of past frames and is updated incrementally.

The cache is structured as a ring buffer per transformer layer. Each layer maintains a buffer of capacity $(L + t_{\text{pf}})$ token slots, where L is the layer's context window in tokens and t_{pf} is the number of tokens per frame. The final t_{pf} slots form a dedicated tail that always holds the current frame under active denoising. During multi-step denoising of a frame, the cache is set to frozen: the tail is written on every denoising step (allowing the model to attend to the current noisy frame), but the ring buffer is not updated. Once the frame is fully denoised to $\sigma = 0$, the cache is unfrozen and the clean frame is committed to the ring. This ensures the persistent context always consists exclusively of clean, fully denoised frames—avoiding error accumulation from noisy context.

For layers with global dilated attention (every 4th layer), the ring buffer uses a pinned dilation factor of 8: new frames evict slots at spacing 8 within the 128-frame window, retaining an evenly-spaced temporal sample of the long-range history. Local attention layers maintain a dense 16-frame ring.

5.2 Seed Frames and Context Priming

Generation begins with a set of seed frames—clean ground-truth latents provided as initial context. By default, a single seed image is used as initial context. The first generated latent is conditioned on this seed frame; subsequent latents are generated in groups of 4. The seed frame is not re-denoised; instead, a zero-noise forward pass ($\sigma = 0$) is run to populate the KV cache without modifying the latent. This primes the cache with the seed before the first generated frame is produced.

During the pretrained (pre-distillation) inference regime, previously generated context frames are held at a small fixed noise level $\sigma_{\text{ctx}} = 0.15$ when used as conditioning. This `noise_prev` parameter provides a slight corruption that helps bridge the gap between the clean seed frames and the partially noisy frames the model encountered during Diffusion Forcing training. After Self-Forcing distillation, the model is trained to condition on fully clean self-generated history, so `noise_prev` is set to 0.0.

5.3 Denoising Schedules

Pre-distillation. The pretrained model uses 60 denoising steps per frame with a `FlowMatchEulerDiscreteScheduler`. A `sigma_shift` of 3.0 compresses the schedule toward lower noise levels, concentrating sampling budget in the refinement regime. Classifier-free guidance (CFG) is applied at each step by running two forward passes—one conditioned, one unconditioned—and combining them with a guidance weight. This doubles the effective per-frame cost, yielding $60 \times 2 = 120$ model evaluations per frame.

Post-distillation. After Self-Forcing distillation, the denoising schedule collapses to 4 steps per frame with a fixed trajectory: $\sigma \in \{1.0, 0.9, 0.75, 0.3, 0.0\}$. CFG is no longer applied separately—it is baked into the student during distillation (teacher CFG scales: $\omega_{\text{ctrl}} = 15.0$, $\omega_{\text{ctx}} = 2.0$)—so each frame requires only 4 single forward passes. The distilled model conditions on zero-noise clean self-generated context, eliminating the `noise_prev` corruption.

5.4 Runtime Optimizations

Two critical code paths—the frozen-cache denoising step and the unfrozen cache-writing step—are compiled ahead of time with `torch.compile(mode="max-autotune", fullgraph=True, dynamic=False)`, enabling kernel fusion, memory layout optimization, and CUDA graph capture. With `dynamic=False`, the compiler assumes fixed input shapes and generates fully static execution graphs, eliminating Python dispatch overhead on each forward pass. The full diffusion model is deployed in INT8 quantized precision for throughput-critical inference, as reflected in the benchmark results of Section 8. BF16 inference is also supported where VRAM permits.

6 Autoencoders & Tokenization

6.1 Latent Autoencoder

Waypoint-1.5 encodes video into a compact latent representation using **TAEHV 1.5** [4], a tiny autoencoder designed to approximate the latent space of the Hunyuan VAE 1.5. TAEHV 1.5 applies spatial downsampling alongside a temporal compression factor of 4, mapping 60 raw FPS to 15 latent frames per second. The encoder produces 32-channel latent frames, significantly more compact than the original VAE latent. For a 512×1024 input clip, the encoded latent has spatial dimensions 32×64; for a 256×512 input clip, the latent spatial dimensions are 16×32. This compression reduces a 13-second video clip (768 raw frames) to 192 latent frames for the diffusion model, making transformer-scale training over long sequences tractable.

6.2 Distilled Autoencoder Decode

Standard high-fidelity autoencoders are designed for quality rather than throughput, and naive inference can make the decode step a significant bottleneck in a real-time pipeline. TAEHV 1.5 sidesteps this problem by design: as a tiny distilled approximation of the full Hunyuan VAE 1.5, it trades a small amount of reconstruction fidelity for a dramatically reduced parameter count and arithmetic cost. Inference runs in FP16 under `torch.inference_mode()`, achieving approximately **800 FPS** decode throughput on an NVIDIA RTX 5090—fast enough that the decoder contributes negligible overhead relative to the diffusion model generation step.

6.3 Patchification

After encoding, each latent frame is further tokenized by a learned 2×2 patch embedding (Conv2d with kernel size and stride $[2, 2]$). At 512×1024, this produces $16 \times 32 = 512$ tokens per latent frame; at 256×512, it produces $8 \times 16 = 128$ tokens per latent frame. The spatial layout of patches is arranged to maximize alignment with the block boundaries used by FlexAttention's sparse kernel, ensuring block-level parallelism maps to spatially contiguous regions.

7 Data

7.1 Owl-Control Dataset

Waypoint-1.5 is trained on **Owl-Control**, a proprietary dataset of controller-synchronized video game footage comprising over 100,000 hours of gameplay across hundreds of distinct games. The dataset is designed to pair high-fidelity visual observations with the full controller input stream that produced them, enabling the model to learn a grounded mapping from game state to player action.

Raw video is captured at 720P resolution and 60 FPS. Controller inputs—including all keyboard binary states, raw mouse displacement, and scroll events—are recorded at up to 10,000 Hz per channel via a high-fidelity hardware logging layer. This high-frequency controller stream is precisely synchronized to the video timeline, enabling accurate alignment between visual frames and the controller actions that produced them.

7.2 Preprocessing Pipeline

The preprocessing pipeline is illustrated in Figure 4. Raw 60 FPS RGB video is processed offline through the TAEHV 1.5 encoder with a temporal compression factor of 4, producing latent sequences at 15 latent frames per second. Concurrently, the high-frequency controller input stream is aligned to the latent frame grid and bucketed into 15 controller value buckets per second of latent video. Each snapshot encodes the full button state vector (256-bucket discretization), the net mouse displacement $(\Delta x, \Delta y)$ accumulated since the previous snapshot, and the scroll direction. The resulting paired (latent frames, controller sequence) clips are stored in a preprocessed format for efficient training data loading.

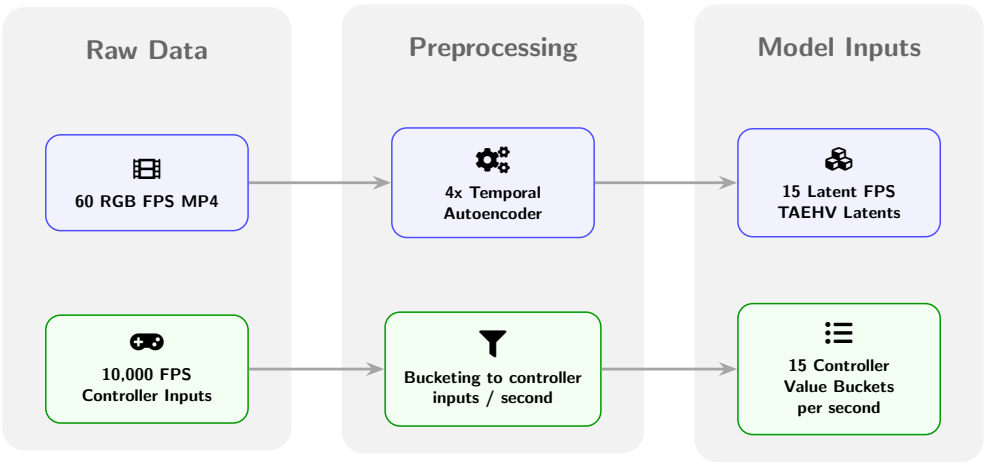


Figure 4: **Data Loading Pipeline Workflow.** Raw 60 FPS gameplay video and 10,000 Hz controller telemetry are encoded offline into TAEHV 1.5 latents at 15 latent FPS and bucketed controller snapshots, then packed into training sequences with cross-document attention masking.

7.3 Sequence Packing

To maximize GPU utilization and minimize padding waste during training, we apply cross-document sequence packing: latent sequences from multiple game clips are concatenated into a single training sequence with document IDs (`doc_id`) tracked per token. Attention masking prevents information from crossing clip boundaries, preserving the causal structure across packed sequences while allowing the model to train on long effective context windows without padding overhead.

8 Evaluation

The central design objective of Waypoint-1.5 is real-time interactive generation on consumer hardware. We therefore evaluate the system primarily on throughput—measured in latent frames per second (latent FPS)—across a range of consumer NVIDIA GPUs. Latent FPS measures the rate at which the diffusion model generates latent frames; pixel-space video is subsequently decoded by the TAEHV 1.5 autoencoder at approximately 800 FPS, adding negligible overhead. Latent FPS therefore governs the interactive responsiveness of the system.

Table 2: Latent frames per second (latent FPS) throughput of Waypoint-1.5 across consumer GPUs. OOM indicates the BF16 model footprint exceeds available VRAM at 720P; INT8 quantization resolves this constraint.

Hardware	720P INT8	720P BF16	360P INT8	360P BF16
RTX PRO 6000 Blackwell	132.40	87.76	371.08	303.72
RTX 5090	111.40	66.28	346.64	277.52
RTX 4090	72.88	66.60	275.84	207.52
RTX 5070 Ti	59.68	39.96	270.48	178.56
RTX 3090 Ti	43.64	29.00	198.16	104.36
RTX 3090	37.36	21.60	188.44	116.12
RTX 5060 Ti	31.92	17.72	162.08	85.68
RTX 4060 Ti	28.32	16.40	126.84	78.68
RTX 3070	23.24	OOM	120.08	OOM
RTX 3060	15.76	8.80	81.88	36.76

We benchmark the 720P and 360P variants under two precision regimes: INT8 quantization (enabled via custom kernels and `torch.compile`) and standard BF16. Results are reported in Table 2. We consider ≥ 30 latent FPS a practical threshold for smooth real-time interactive generation.

Figure 5: **Waypoint-1.5 autoregressive rollouts.** Five example action-conditioned rollouts of generated in real time on consumer hardware (5 frames per row, left to right). Each row is an independent autoregressive sequence conditioned on keyboard and mouse input. The model maintains visual coherence and scene consistency over long horizons across diverse environments.

At 720P, INT8-quantized Waypoint-1.5 meets the 30 latent FPS threshold on all GPUs from the RTX 5060 Ti upward. The 360P variant reaches this threshold on all tested hardware, including the RTX 3060. INT8 quantization provides a consistent 1.5–2$\times$ throughput improvement over BF16 across all GPUs, with the additional benefit of resolving out-of-memory (OOM) conditions for the 720P BF16 model on the RTX 3070, whose 8 GB VRAM is insufficient to hold the full BF16 KV cache and model weights simultaneously. The RTX PRO 6000 Blackwell achieves the highest throughput at 132 latent FPS (720P INT8), more than $4\times$ the real-time threshold.

9 Safety & Ethics

Safety in interactive world models presents challenges distinct from those in text, image, or even video generation systems. Unlike models that produce a single static output, world models generate persistent environments at high frame rates that respond continuously to user input. A single prompt can yield minutes of interactive content spanning diverse visual states, making post-hoc output review insufficient on its own. Safety must therefore be addressed across the full pipeline: in the training data, at prompt ingestion, during generation, at output delivery, and through ongoing monitoring.

At Overworld, safety has been part of the development process since the company's founding. Rather than treating it as a post-deployment addition, we have built layered safeguards into each stage of

the Waypoint pipeline. This section describes those layers, the tradeoffs involved, and our plans for continued improvement.

We acknowledge that safety in generative systems involves inherent tensions between creative freedom and content restriction, between open research and controlled deployment, and between moving quickly as a startup and building robust safeguards. We do not claim to have resolved these tensions. Instead, we describe the systems we have built so far and the framework under which we continue to iterate.

9.1 Safety Procedures Per Major Model Release

Each major release of the Waypoint model family follows a structured safety review process. Before deployment, we conduct internal testing to identify known failure modes, calibrate filtering thresholds, and evaluate the model's behavior under adversarial prompting. This process is informed by prior releases and evolves with each iteration. A soft, controlled release is always conducted via our self-hosted overworld.stream service to evaluate how users are attempting to circumvent safeguards or telegraph intended use.

Safety review is not a single gate but a recurring cycle. Waypoint 1.0 established the initial pipeline; Waypoint-1.5 incorporated lessons from deployment of the hosted Overworld.stream service, including observed user behavior patterns, attempted jailbreaks, and content distribution analysis from real-world usage.

9.2 Data Source Curation

Training data and the decisions surrounding the data included in training is the first step towards ensuring safety. We apply source-level filtering before any data enter the training pipeline. This includes excluding sources known to contain disproportionate amounts of personally identifiable information (PII), adult content, or content that violates terms of service.

Domain selection is guided by the model's intended use case. We are primarily concerned with creating an interactive entertainment experience and enabling creative use. We prioritize sources whose content distribution aligns with that goal.

9.3 Dataset Analysis & Filtering

Beyond source-level curation, we perform clip-level and frame-level analysis of the pretraining corpus. We built a video scanning pipeline that processes the full dataset using a multi-stage classification approach. The pipeline runs two classifiers sequentially: a ResNet-50-based explicit content detector and a CLIP-based classifier (ViT-B/32) queried with prompts targeting potential depictions of minors across photorealistic, cartoon, and stylized formats.

Each processed video produces structured metadata recording flagged frames by category and a determination of whether the clip should be excluded from training. This frame-level approach avoids the blunt removal of entire clips when only isolated frames are problematic, preserving useful training signal while removing unsafe content.

Flagged content above confidence thresholds is downloaded and manually reviewed before final calibration, resulting in a dataset corpora with metadata which can be audited. This metadata also enables retrospective analysis: if filtering criteria change in future releases, previously processed data can be re-evaluated without full reprocessing.

9.4 Prompt Filtering & Sanitation

User prompts pass through an intermediate sanitation layer before reaching the world model. This layer serves a dual purpose: 1) safety enforcement and 2) structural alignment with the model's training distribution.

On the safety side, the sanitizer detects and transforms prompts containing references to explicit sexual content, depictions of minors, celebrity likeness, recognizable branded intellectual property, and other content categories. Rather than rejecting prompts outright, the system rewrites user intent into a safe environmental description suitable for generation. For example, a prompt referencing

a specific copyrighted work is transformed into a description that captures the intended aesthetic without producing derivative content.

On the structural side, as described in Section 3.4, the sanitizer enriches under-specified prompts into the structured signal format the world model expects: a) a seed image, b) a short synthetic video extension, and c) aligned control signals derived from the inverse dynamics model. This multi-feature approach means that the safety layer is not simply an optional filter but an integral component of the generation pipeline.

Early versions of the sanitizer returned structured information about the elements removed alongside the sanitized prompt, which was useful for debugging but introduced unnecessary complexity. The current system returns only the sanitized prompt, reducing latency and surface area for failure.

9.5 Mid-Generation Classification & Redirection

Because world models generate content continuously and interactively, safety cannot rely solely on input filtering and output review. A prompt that passes all safety checks may nonetheless lead to unsafe visual states as the generated environment evolves in response to user actions.

9.6 Output Classification & Transformation

In addition to input and mid-generation safeguards, we apply post-generation analysis to completed outputs. For the hosted Overworld.stream service, generated content passes through output classifiers before delivery to users.

9.7 Prompt-Output Audit Dashboard

Monitoring deployed systems requires visibility into what users are requesting and what the model is generating. We are building moderation dashboards for Overworld.stream that track prompt distributions, flagged content rates, and generation outcomes over time.

This monitoring serves multiple purposes: detecting emerging misuse patterns, identifying gaps in the filtering pipeline, and informing the prompt distribution analysis that guides future training data curation (see Section 7).

9.8 Red-Teaming & Independent Audit

Structured adversarial testing is a critical component of safety evaluation for generative systems. We conduct internal red-teaming before each major release, with team members attempting to elicit disallowed content through prompt manipulation, multi-step interaction sequences, and edge-case inputs.

As a small team, we recognize that internal perspectives are limited. We actively seek external input from researchers, artists, and community members, and maintain public channels for reporting problematic generations.

9.9 Model Cards, Model Licenses & Release Qualification

Each release of the Waypoint model family includes documentation of capabilities, known limitations, and intended use cases in a published model card. We encourage downstream users and researchers building on Waypoint to publish model cards for their derivative work as well.

Our release strategy reflects a commitment to open research alongside responsible deployment. Waypoint releases include:

- An Apache-licensed base model for broad research and commercial use.
- A larger non-commercial research model for academic experimentation.
- WorldEngine, our GPL-licensed inference library.

This tiered licensing structure supports open experimentation while discouraging closed, unattributed derivatives. The GPL license on WorldEngine ensures that modifications to the inference pipeline — including any safety-relevant changes — remain open.

Terms of Service apply to hosted services (Overworld.stream, Biome) and to partners hosting Waypoint models. These terms define prohibited content categories, usage restrictions, and reporting obligations. Licensing and product design are treated as safety mechanisms in their own right: they shape how the model is used in practice and establish accountability for downstream deployment.

10 Conclusion

We have presented Waypoint-1.5, a 1.28B parameter single-stream causal Diffusion Transformer for real-time interactive video generation on consumer hardware. The system integrates frame-causal attention with a static rolling KV cache, Diffusion Forcing pretraining, Self-Forcing DMD post-training distillation, and the TAEHV 1.5 compact autoencoder, trained on over 100,000 hours of controller-synchronized gameplay from the Owl-Control dataset. Together, these components enable stable long-horizon autoregressive rollout at 4 denoising steps per frame, with classifier-free guidance baked in during distillation rather than applied at inference.

Benchmarked across ten consumer NVIDIA GPUs, the INT8-quantized 720P variant meets the 30 latent FPS real-time threshold on all GPUs from the RTX 5060 Ti upward, and the 360P variant achieves this on all tested hardware including the RTX 3060. These results demonstrate that interactive, controllable video generation at interactive frame rates is achievable on the hardware end users already own, without relying on enterprise level infrastructure or streamed delivery. We hope Waypoint-1.5 serves as a practical foundation for future research in real-time world models and interactive video generation.

References

[1] Anmol Agarwal, Pranay Meshram, Sumer Singh, Saurav Suman, Andrew Lapp, Shahbuland Matiana, Louis Castricato, and Spencer Frazier. Combat: Conditional world models for behavioral agent training. *arXiv preprint arXiv:2603.00825*, 2026.

[2] Joshua Ainslie, James Lee-Thorp, Michiel De Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901, 2023.

[3] Philip J. Ball, Jakob Bauer, Frank Belletti, Bethanie Brownfield, Ariel Ephrat, Shlomi Fruchter, Agrim Gupta, Kristian Holsheimer, Aleksander Holynski, Jiri Hron, Christos Kaplanis, Marjorie Limont, Matt McGill, Yanko Oliveira, Jack Parker-Holder, Frank Perbet, Guy Scully, Jeremy Shar, Stephen Spencer, Omer Tov, Ruben Villegas, Emma Wang, Jessica Yung, Cip Baetu, Jordi Berbel, David Bridson, Jake Bruce, Gavin Buttimore, Sarah Chakera, Bilva Chandra, Paul Collins, Alex Cullum, Bogdan Damoc, Vibha Dasagi, Maxime Gazeau, Charles Gbadamosi, Woohyun Han, Ed Hirst, Ashyana Kachra, Lucie Kerley, Kristian Kjems, Eva Knoepfel, Vika Koriakin, Jessica Lo, Cong Lu, Zeb Mehring, Alex Moufarek, Henna Nandwani, Valeria Oliveira, Fabio Pardo, Jane Park, Andrew Pierson, Ben Poole, Helen Ran, Tim Salimans, Manuel Sanchez, Igor Saprykin, Amy Shen, Sailesh Sidhwani, Duncan Smith, Joe Stanton, Hamish Tomlinson, Dimple Vijaykumar, Luyu Wang, Piers Wingfield, Nat Wong, Keyang Xu, Christopher Yew, Nick Young, Vadim Zubov, Douglas Eck, Dumitru Erhan, Koray Kavukcuoglu, Demis Hassabis, Zoubin Gharamani, Raia Hadsell, Aäron van den Oord, Inbar Mosseri, Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 3: A new frontier for world models. 2025.

[4] Ollin Boer Bohan. TAEHV: Tiny AutoEncoder for Hunyuan Video. https://github.com/madebyollin/taehv, 2025. Accessed: 2026-06-10.

[5] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. URL https://openai.com/research/video-generation-models-as-world-simulators.

[6] Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*,

volume 37, pages 24081–24125. Curran Associates, Inc., 2024. doi: 10.52202/079017-0759. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/2aee1c4159e48407d68fe16ae8e6e49e-Paper-Conference.pdf.

[7] Justin Cui, Jie Wu, Ming Li, Tao Yang, Xiaojie Li, Rui Wang, Andrew Bai, Yuanhao Ban, and Cho-Jui Hsieh. Self-forcing++: Towards minute-scale high-quality video generation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=DzvPiqh23f.

[8] Xingtong Ge, Xin Zhang, Tongda Xu, Yi Zhang, Xinjie Zhang, Yan Wang, and Jun Zhang. Senseflow: Scaling distribution matching for flow-based text-to-image distillation. *arXiv preprint arXiv:2506.00523*, 2025.

[9] Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, et al. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.

[10] Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=mSiN7iOBYH.

[11] Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL https://kellerjordan.github.io/posts/muon/.

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

[13] Zichong Li, Liming Liu, Chen Liang, Weizhu Chen, and Tuo Zhao. Normuon: Making muon more efficient and scalable, 2025. URL https://arxiv.org/abs/2510.05491.

[14] Yixin Liu, Kai Zhang, Yuan Li, Zhiling Yan, Chujie Gao, Ruoxi Chen, Zhengqing Yuan, Yue Huang, Hanchi Sun, Jianfeng Gao, et al. Sora: A review on background, technology, limitations, and opportunities of large vision models. *arXiv preprint arXiv:2402.17177*, 2024.

[15] Munan Ning, Bin Zhu, Yujia Xie, Bin Lin, Jiaxi Cui, Lu Yuan, Dongdong Chen, and Li Yuan. Video-bench: A comprehensive benchmark and toolkit for evaluating video-based large language models. *Computational Visual Media*, 2025.

[16] NVIDIA, :, Aditi, Niket Agarwal, Arslan Ali, Jon Allen, Martin Antolini, Adeline Aubame, Alisson Azzolini, Junjie Bai, Maciej Bala, Yogesh Balaji, Josh Bapst, Aarti Basant, Mukesh Beladiya, Mohammad Qazim Bhat, Zaid Pervaiz Bhat, Dan Blick, Vanni Brighella, Han Cai, Tiffany Cai, Eric Cameracci, Jiaxin Cao, Yulong Cao, Mark Carlson, Carlos Casanova, Ting-Yun Chang, Yan Chang, Yu-Wei Chao, Prithvijit Chattopadhyay, Roshan Chaudhari, Chieh-Yun Chen, Junyu Chen, Ke Chen, Qizhi Chen, Wenkai Chen, Xiaotong Chen, Yu Chen, An-Chieh Cheng, Click Cheng, Xiu Chia, Jeana Choi, Chaeyeon Chung, Wenyan Cong, Yin Cui, Magdalena Dadela, Nalin Dadhich, Wenliang Dai, Joyjit Daw, Alperen Degirmenci, Rodrigo Vieira Del Monte, Robert Denomme, Sameer Dharur, Marco Di Lucca, Ke Ding, Wenhao Ding, Yifan Ding, Yuzhu Dong, Nicole Drumheller, Yilun Du, Aigul Dzhumamuratova, Aleksandr Efitorov, Hamid Eghbalzadeh, Naomi Eigbe, Imad El Hanafi, Hassan Eslami, Benedikt Falk, Jiaojiao Fan, Jim Fan, Amol Fasale, Sergiy Fefilatyev, Liang Feng, Francesco Ferroni, Sanja Fidler, Xiao Fu, Vikram Fugro, Prashant Gaikwad, TJ Galda, Katelyn Gao, Yihuai Gao, Wenhang Ge, Sreyan Ghosh, Arushi Goel, Vivek Goel, Akash Gokul, Rama Govindaraju, Jinwei Gu, Miguel Guerrero, Elfie Guo, Aryaman Gupta, Siddharth Gururani, Hugo Hadfield, Song Han, Ankur Handa, Zekun Hao, Mohammad Harrim, Ali Hassani, Nathan Hayes-Roth, Yufan He, Chris Helvig, Cyrus Hogg, Madison Huang, Michael Huang, Sophia Huang, Yufan Huang, Jacob Huffman, DeLesley Hutchins, Suneel Indupuru, Boris Ivanovic, Arihant Jain, Joel Jang, Ryan Ji, Yanan Jian, Dongfu Jiang, Jingyi Jin, Atharva Joshi, Nikhilesh Joshi, Pranjali Joshi, Andy Ju, Jaehun Jung, Weiwei Kang, Scott Kassekert, Jan Kautz, Ashna Khetan, Julia Kiczka, Slawek Kierat, Gwanghyun Kim, Kuno Kim, Sunny Kim, Kezhi Kong, Xin Kong, Zhifeng

Kong, Tomasz Kornuta, Egor Krivov, Hui Kuang, Saurav Kumar, Chia-Wen Kuo, George Kurian, Wojciech Kutak, JF Lafleche, Himangshu Lahkar, Omar Laymoun, Jayjun Lee, Sanggil Lee, Gabriele Leone, Boyi Li, Freya Li, Jiajun Li, Jinfeng Li, Ling Li, Pengcheng Li, Shangru Li, Tingle Li, Xiaolong Li, Xuan Li, Zhaoshuo Li, Zhiqi Li, Hao Liang, Maosheng Liao, Chen-Hsuan Lin, Tsung-Yi Lin, Ming-Yu Liu, Sifei Liu, Zihan Liu, Hai Loc Lu, Xiangyu Lu, Alice Luo, Ruipu Luo, Wenjie Luo, Jiangran Lyu, Martin Ding Ma, Nic Ma, Qianli Ma, Dawid Majchrowski, Louis Marcoux, Miguel Martin, Qing Miao, Ashkan Mirzaei, Shreyas Misra, Kaichun Mo, Durra Mohsin, Hyejin Moon, Pawel Morkisz, Saeid Motiian, Kirill Motkov, Seungjun Nah, Yashraj Narang, Deepak Narayanan, Thabang Ngazimbi, Julian Ouyang, Shubham Pachori, David Page, Yatian Pang, Sehwi Park, Mahesh Patekar, Mostofa Patwary, Marco Pavone, Trung Pham, Wei Ping, Soha Pouya, Shrimai Prabhumoye, Varun Praveen, Delin Qu, Hesam Rabeti, Morteza Ramezanali, Marilyn Reeb, Xuanchi Ren, Kristen Rumley, Wojciech Rymer, Jun Saito, Yeongho Seol, John Shao, Piyush Shekdar, Tianwei Shen, Humphrey Shi, Min Shi, Stella Shi, Kevin Shih, Mohammad Shoeybi, Mateusz Sieniawski, Shuran Song, Alexander Sotelo, Amir Sotoodeh, Sunil Srinivasa, Vignesh Srinivasakumar, Bartosz Stefaniak, Rahul Heinrich Steiger, Shangkun Sun, Jiaxiang Tang, Shitao Tang, Yangyang Tang, Yue Tang, Tolou Tavakkoli, Kayley Ting, Krzysztof Tomala, Wei-Cheng Tseng, Jibin Varghese, Sergei Vasilev, Thomas Volk, Raju Wagwani, Roger Waleffe, Andrew Z. Wang, Boxiang Wang, Haoxiang Wang, Qiao Wang, Shihao Wang, Shijie Wang, Ting-Chun Wang, Yan Wang, Yu Wang, Rohit Watve, David Wehr, Fangyin Wei, Xinshuo Weng, Jay Zhangjie Wu, Kedi Wu, Hongchi Xia, Summer Xiao, Tianjun Xiao, Kevin Xie, Daguang Xu, Jiashu Xu, Mengyao Xu, Ruqing Xu, Xingqian Xu, Yao Xu, Dinghao Yang, Dong Yang, Hans Yang, Xiaodong Yang, Xuning Yang, Yichu Yang, Yurong You, Zhiding Yu, Hao Yuan, Simon Yuen, Xiaohui Zeng, Pengcuo Zeren, Cindy Zha, Haotian Zhang, Jenny Zhang, Jing Zhang, Liangkai Zhang, Paris Zhang, Shun Zhang, Xuanmeng Zhang, Zhizheng Zhang, Ann Zhao, Yilin Zhao, Yuliya Zhautouskaya, Charles Zhou, Fengzhe Zhou, Shilin Zhu, Yuke Zhu, Dima Zhylko, and Artur Zolkowski. Cosmos 3: Omnimodal world models for physical ai, 2026. URL https://arxiv.org/abs/2606.02800.

[17] Georgy Savva, Oscar Michel, Daohan Lu, Suppakit Waiwitlikhit, Timothy Meehan, Dhairya Mishra, Srivats Poddar, Jack Lu, and Saining Xie. Solaris: Building a multiplayer video world model in minecraft, 2026. URL https://arxiv.org/abs/2602.22208.

[18] Kiwhan Song, Boyuan Chen, Max Simchowitz, Yilun Du, Russ Tedrake, and Vincent Sitzmann. History-guided video diffusion. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=j8Vr3E3vhy.

[19] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[20] DreamX Team, Yancheng Bai, Rui Chen, Xiangxiang Chu, Rujing Dang, Hao Dou, Bingjie Gao, Qiwen Gu, Siyu Hong, Jiachen Lei, Geng Li, Jifan Li, Ruimin Lin, Qingfeng Shi, Bingze Song, Lei Sun, Jing Tang, Ruitian Tian, Jun Wang, Jiahong Wu, Pengfei Zhang, Shen Zhang, and Jiashu Zhu. Dreamx-world 1.0: A general-purpose interactive world model, 2026. URL https://arxiv.org/abs/2606.16993.

[21] Kairos Team, Fei Wang, Shan You, Qiming Zhang, Tao Huang, Zuoyi Fu, Zhisheng Zheng, Yunlong Xi, Feng Lv, Xiaoming Wu, Zeyu Liu, Cong Wan, Pu Li, Ruiqing Yang, Xiaoou Li, Wei Wang, Kangkang Zhu, Yuwei Zhang, Shi Fu, Zheng Zhang, Xiaoning Wu, Xuzeng Fan, Dacheng Tao, and Xiaogang Wang. Kairos: A native world model stack for physical ai, 2026. URL https://arxiv.org/abs/2606.16533.

[22] Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Fvd: A new metric for video generation. 2019.

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

[24] Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Yingcong Chen, Yao Lu, et al. Longlive: Real-time interactive long video generation. *arXiv preprint arXiv:2509.22622*, 2025.

[25] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems*, 37:47455–47487, 2024.

[26] Haoyi Zhu, Haozhe Liu, Yuyang Zhao, Tian Ye, Junsong Chen, Jincheng Yu, Tong He, Song Han, and Enze Xie. Sana-wm: Efficient minute-scale world modeling with hybrid linear diffusion transformer, 2026. URL https://arxiv.org/abs/2605.15178.