Hierarchical Denoising For Multi-Step Visual Reasoning

Zezhong Qian[1,6], Xiaowei Chi[2,6], Chak-Wing Mak[1,6], Tianze Zhou[3,6],
Ruibin Yuan[2,5], Yuhan Rui[1,6], Hengzhe Sun[1], Zhuoqun Wu[4],
Yuming Li[1], Siyuan Qian[1], Sirui Han[2], Shanghang Zhang[1]

[1]State Key Laboratory of Multimedia Information Processing, School of Computer Science, Peking University
[2]The Hong Kong University of Science and Technology
[3]Beihang University
[4]Fuzhou University
[5]Multimodal Art Projection
[6]Muka Robotics

Abstract

Video models are recently evolving into vision foundation models, but they still lack human-like, multi-step reasoning. Existing streaming autoregressive diffusion models are efficient but lack the reasoning ability, whereas bidirectional diffusion allows for global revision but incurs high inference cost due to the dense frames in fixed-sequence denoising. Consequently, both paradigms struggle to maintain logical consistency with low-latency streaming in complex reasoning tasks. Bridging this gap, we propose **HDR** (*Hierarchical Denoising for Visual Reasoning*), a unified framework for multi-step reasoning by integrating hierarchical latents into the causal video generation process. HDR organizes video latents into a tree-structured hierarchy to perform coarse-to-fine reasoning before streaming output. Coarse denoising layers maintain uncertain hypotheses for global planning, while finer denoising layers progressively refine them into concrete visual states. A sparse hierarchical attention pattern (SHAP) further reduces temporal attention cost. We construct a level-stratified multi-step video reasoning benchmark with out-of-distribution cases, covering six tasks: *maze navigation, Tower of Hanoi, one-line drawing, sliding puzzle, Sokoban, and water pouring*. Compared with the streaming autoregressive diffusion baseline, HDR improves overall success from 34.22 to 60.29 (76.2% relative gain) in multi-step reasoning accuracy, and improves average progress from 76.00 to 89.56, indicating more consistent intermediate reasoning trajectories. For deployment efficiency, HDR maintains low-latency streaming at 0.70s per latent, $54.2\times$ faster than bidirectional diffusion during streaming. HDR also demonstrates strong data efficiency, retaining 82.9% of its full-data success score using only 2% of the training data, compared with 52.0% for bidirectional diffusion. Further experiments on real-world robots showcase the potential of HDR in physical interaction, providing a new paradigm for physical world modeling. Project demo page is available at `https://hierarchical-diffusion-reasoning.github.io/`.

1 Introduction

Video models are evolving from realistic video synthesizers into generalist visual foundation models capable of visual reasoning [27, 26, 25, 28, 39, 37, 13]. Recent work such as Veo3 [27] showed that large video generation models can solve tasks such as maze navigation, symmetry completion,

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

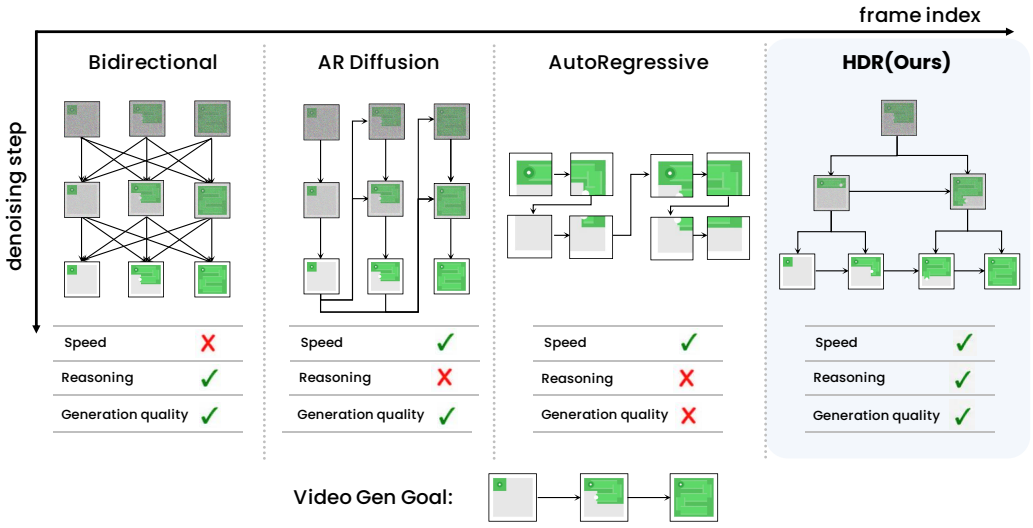


Figure 1: Comparison of video generation paradigms. Streaming autoregressive diffusion models are efficient but commit too early for reliable multi-step reasoning, while bidirectional diffusion supports global revision but requires costly dense fixed-sequence denoising. HDR performs hierarchical denoising before streaming output: coarse layers maintain high-level hypotheses, finer layers refine them into visual states, and sparse hierarchical attention keeps generation efficient.

physical reasoning, and tool-use simulation through generated visual trajectories. Complementarily, recent analysis of diffusion-based video reasoning suggests that such reasoning is closely tied to intermediate denoising states, where the model can maintain and refine candidate solutions over multiple steps [26]. These findings raise a key question: how can video models support reliable multi-step reasoning while still enabling low-latency streaming generation?

Existing video generation paradigms struggle to satisfy both multi-step reasoning and low-latency streaming. Streaming autoregressive diffusion models, such as CausVid [34] and CausalForcing [41] generate efficiently by conditioning each step only on past context, but this left-to-right commitment limits their ability to revise previous decisions and perform multi-step reasoning [30, 4, 9, 16]. In contrast, bidirectional diffusion models jointly denoise a fixed video sequence, allowing global information flow and revision across time [20, 24, 27]. However, this requires dense full-sequence updates at every denoising step, leading to high deployment cost and poor compatibility with streaming generation [34, 5, 2, 21]. These limitations reveal a fundamental tension: current models either generate efficiently but struggle with logical consistency over multiple steps, or reason globally at the cost of high-latency fixed-sequence denoising.

Motivated by this observation, we propose **HDR** (*Hierarchical Denoising for Visual Reasoning*), a unified framework that integrates hierarchical latents into the streaming autoregressive diffusion process. As illustrated in Figure 1, HDR organizes video latents into a tree-structured hierarchy and performs coarse-to-fine multi-step reasoning before streaming output. This hierarchy gives the model an explicit intermediate space for high-level planning before it commits to frame-level generation.

A key design of HDR is to match denoising strength to hierarchy level. Instead of fully denoising every layer, HDR keeps coarse layers at higher noise levels so they can preserve multiple possible global plans, while finer layers receive stronger denoising and lower residual noise to instantiate these plans into concrete visual states. HDR further introduces **SHAP** (*Sparse Hierarchical Attention Pattern*), which lets each token communicate only with fixed local and parent-level contexts for fast retrieval. This enables multi-scale information flow without dense full-sequence attention, reducing temporal attention cost while preserving streaming generation.

We construct a level-stratified multi-step video reasoning benchmark with out-of-distribution cases, covering six tasks: *maze navigation, Tower of Hanoi, one-line drawing, sliding puzzle, Sokoban, and water pouring*. These tasks require models to maintain logical consistency across multi-step reasoning trajectories rather than merely generating locally plausible motion. Experiments show

that HDR substantially improves both final task completion and intermediate reasoning consistency: it improves the overall success score from 34.22 to 60.29 (76.2% relative gain), and increases the overall average progress score from 76.00 to 89.56. HDR also preserves efficient streaming behavior, achieving 0.70s per latent during streaming, $54.2\times$ faster than bidirectional diffusion. In addition, HDR demonstrates strong data efficiency, retaining 82.9% of its full-data success score when trained with only 2% of the data, compared with 52.0% for bidirectional diffusion. Finally, real-world robot maze experiments show that HDR can transfer its hierarchical reasoning ability to physical interaction, suggesting its potential as a new paradigm for physical world modeling.

Our contributions can be summarized as follows:

- We identify the core tension in multi-step video reasoning: models must maintain logical consistency across long trajectories while also supporting low-latency streaming generation.
- We propose **HDR** (*Hierarchical Denoising for Visual Reasoning*), a unified framework that integrates hierarchical latents into streaming autoregressive diffusion and performs coarse-to-fine reasoning before streaming output.
- We introduce a hierarchy-matched denoising schedule and **SHAP** (*Sparse Hierarchical Attention Pattern*). The former preserves high-level hypotheses at coarse layers and refines them at finer layers, while the latter reduces temporal attention cost through local and parent-level contexts.
- We construct a level-stratified multi-step video reasoning benchmark with OOD cases, and demonstrate HDR's advantages in reasoning accuracy, data efficiency, low-latency streaming, and physical-world robot interaction.

2 Related Work

Video Model Reasoning. Recent studies show that video generation models can exhibit reasoning behaviors beyond visual realism. Benchmarks such as VBVR, V-ReasonBench, and VR-Bench evaluate these capabilities across structured problem solving, spatial cognition, physical dynamics, maze navigation, and multi-step planning [25, 18, 35]. Unlike video understanding, where the input video is fixed, video generation requires the model to construct a coherent future trajectory that satisfies both local visual plausibility and global task constraints. This makes generated videos a natural interface for world-model-style reasoning, but also makes early visualized mistakes difficult to correct [28]. Recent analyses further suggest that reasoning in video diffusion models is closely tied to iterative denoising dynamics, where intermediate latents maintain and refine uncertain hypotheses, rather than arising solely from frame-by-frame prediction [26]. However, existing work mainly benchmarks or analyzes emergent reasoning in bidirectional generators, rather than designing architectures that preserve reasoning ability under streaming generation constraints [25, 18, 26].

Autoregressive Video Diffusion Models. Streaming autoregressive diffusion models replace dense full-sequence denoising with sequential temporal computation, enabling low-latency video generation and efficient KV-cache reuse. Recent work improves this paradigm through chunk-wise rollout, queue-based denoising, AR-guided diffusion, causal attention, training–inference alignment, distillation, cache sharing, and sliding-window KV-cache acceleration [7, 12, 15, 34, 5, 9, 41, 31, 22, 11, 10, 29]. These properties make autoregressive video models attractive for closed-loop robot interaction [14, 33], where low latency is critical, and their context-as-memory structure allows previously generated visual states to serve as a persistent temporal memory [36, 8]. Moreover, because generation proceeds autoregressively, sliding-window KV-cache mechanisms can extend rollout length and support effectively unbounded video generation [31, 16, 5]. However, the same sequential commitment makes earlier decisions difficult to revise: once a frame or latent chunk has been produced, later predictions condition on this committed history. HDR preserves the low-latency structure of streaming autoregressive diffusion while introducing a tree-structured latent hierarchy for coarse-to-fine denoising, enabling revisable high-level planning before committing to fine-grained visual details.

3 Method

We present HDR (*Hierarchical Denoising for Visual Reasoning*), a hierarchical framework for multi-step video reasoning. HDR preserves the low-latency streaming behavior of streaming autoregressive diffusion while introducing a structured intermediate process for global planning and revision.

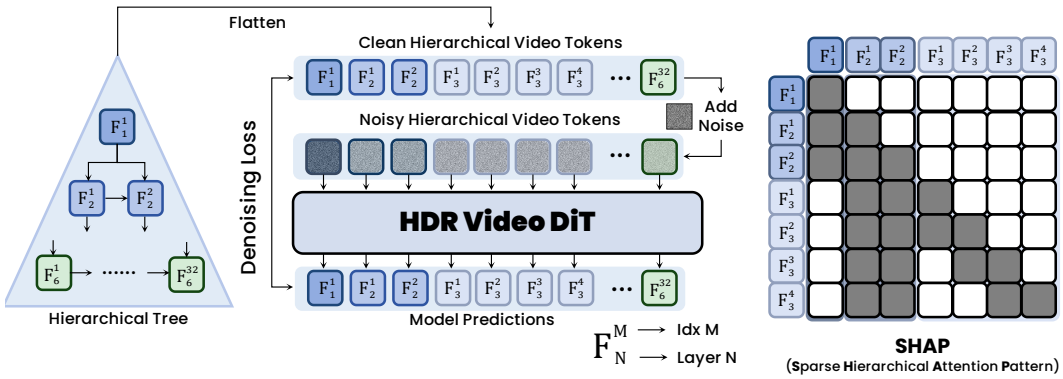


Figure 2: Overview of HDR. Video latents are organized into a tree-structured hierarchy across multiple temporal resolutions. During training, each hierarchy token is corrupted along a flow-matching path and HDR Video DiT is optimized with a layer-wise objective. During inference, all tree tokens are flattened into a coarse-to-fine autoregressive order. **SHAP** (*Sparse Hierarchical Attention Pattern*) defines a structured attention mask over this flattened sequence: each token attends only to fixed local, parent-level, and first-frame contexts rather than the full video sequence. Generated tokens are written into a shared KV cache and reused by later tokens across hierarchy levels, enabling multi-scale information propagation with low temporal attention cost.

We first revisit why streaming autoregressive generation struggles with multi-step reasoning, then introduce the hierarchical latent representation, layer-wise flow-matching objective, and **SHAP** (*Sparse Hierarchical Attention Pattern*), which enables efficient inference over flattened tree tokens.

3.1 Rethinking Streaming Autoregressive Generation for Multi-step Reasoning

We start by comparing bidirectional diffusion and streaming autoregressive diffusion. Let $\mathbf{z}^t = \{z_1^t, \ldots, z_N^t\}$ denote a video latent sequence at denoising step t, where N is the number of temporal latent tokens and z_i^t is the token at temporal position i. Let c denote the conditioning signal, such as text, image, or the first frame. A bidirectional video diffusion model updates the entire sequence jointly at every denoising step [24, 20]:

$$\mathbf{z}^{t-1} = D_\theta(\mathbf{z}^t, t, c), \tag{1}$$

where D_θ is the denoising network. Because all temporal tokens remain noisy during intermediate denoising steps, information can propagate across the whole sequence before video is committed. This allows uncertain hypotheses to be maintained and refined over multiple denoising steps [26]. However, the same global revision ability comes with high cost: each step repeatedly updates a dense fixed-length sequence, making bidirectional diffusion poorly aligned with low-latency streaming.

AR diffusion improves deployment efficiency by factorizing generation from left to right:

$$p(\mathbf{z} \mid c) = \prod_{i=1}^{N} p(z_i \mid z_{<i}, c), \tag{2}$$

where $\mathbf{z} = \{z_1, \ldots, z_N\}$ is the clean latent sequence and $z_{<i}$ denotes all previously generated temporal tokens. With an autoregressive attention mask, token z_i attends only to past tokens and the condition c, enabling streaming inference and KV-cache reuse [34, 41, 9]. However, this structure also creates an irreversible rollout: once z_i is generated, future predictions follow $z_i \to p(z_{i+1} \mid z_{\leq i}, c) \to p(z_{i+2} \mid z_{\leq i+1}, c) \to \cdots$. If an early token encodes an incorrect decision, later tokens must condition on this committed history and cannot revise it, which weakens logical consistency across multi-step reasoning trajectories.

Thus, the central challenge is not simply choosing between bidirectional and streaming autoregressive generation. Bidirectional diffusion supports global revision but incurs high deployment cost, while streaming autoregressive diffusion is efficient but lacks a mechanism for revisable multi-step reasoning. HDR addresses this tension by introducing a hierarchy of latent variables: coarse levels preserve noisy high-level hypotheses for global planning, while finer levels progressively refine them into concrete visual states before streaming output.

3.2 Hierarchical Denoising for Visual Reasoning

HDR represents a video using a tree-structured hierarchy of latent tokens:

$$\mathcal{T} = \{\mathcal{V}^1, \mathcal{V}^2, \ldots, \mathcal{V}^L\}, \qquad \mathcal{V}^\ell = \{v_{\ell,1}, \ldots, v_{\ell,N_\ell}\}. \tag{3}$$

Here, L is the number of hierarchy levels, $\mathcal{V}^\ell$ is the set of latent tokens at level ℓ, N_ℓ is the number of tokens at that level, and $v_{\ell,i}$ denotes the i-th token. We use $\ell = 1$ for the coarsest level and $\ell = L$ for the finest level. Coarse tokens summarize global temporal structure and represent high-level plans, while fine tokens encode local visual details and instantiate the final video dynamics. Each non-root token has a parent in the previous coarser level, denoted as $\pi(\ell, i) = (\ell - 1, p_\ell(i))$ for $\ell > 1$, where $p_\ell(i)$ maps token $v_{\ell,i}$ to its parent index at level $\ell - 1$. The parent token represents the coarse temporal segment that the current token refines.

As shown in Figure 2, HDR constructs hierarchical tokens from the input video latent sequence and performs coarse-to-fine reasoning before streaming output. During training, each hierarchy token is corrupted along a flow-matching path, and the model learns to predict the corresponding velocity target under hierarchical context. During inference, HDR generates tokens from coarse to fine: upper layers form noisy but revisable global hypotheses, while lower layers refine these hypotheses into concrete visual states. Within each level, generation proceeds autoregressively from left to right, preserving the streaming structure of autoregressive diffusion.

3.3 Layer-wise Flow-Matching Objective

We now describe the training objective over hierarchical tokens. For a clean hierarchy token $v^0_{\ell,i}$, we sample a noise token $\epsilon \sim \mathcal{N}(0, I)$ and construct an interpolated token at continuous time $t \in [0, 1]$ as $v^t_{\ell,i} = (1 - t)v^0_{\ell,i} + t\epsilon$. Under this linear probability path, the target velocity field is $u^t_{\ell,i} = \epsilon - v^0_{\ell,i}$. The HDR network is trained to predict this flow velocity from the interpolated token, the timestep, the hierarchical context $h_{\ell,i}$, and the condition c. The layer-wise flow-matching objective sums the velocity regression loss over all hierarchy levels and tokens:

$$\mathcal{L}_{\text{HDR}} = \sum_{\ell=1}^{L} \lambda_\ell \frac{1}{N_\ell} \sum_{i=1}^{N_\ell} \mathbb{E}_{t,\epsilon} \left[\left\| (\epsilon - v^0_{\ell,i}) - u_\theta \left(v^t_{\ell,i}, t, h_{\ell,i}, c \right) \right\|_2^2 \right]. \tag{4}$$

Here, $h_{\ell,i}$ denotes the sparse hierarchical context available to token $v_{\ell,i}$, and λ_ℓ balances the contribution of different hierarchy levels. This objective encourages each level to learn a velocity field appropriate to its temporal abstraction: coarse levels model global planning structure, while fine levels model concrete visual states and motion details.

A key design of HDR is to match denoising strength to hierarchy level. Instead of assigning the same inference budget to every layer, HDR uses a level-dependent sampling budget K_ℓ, with $K_1 < K_2 < \cdots < K_L$. Equivalently, the residual noise level decreases from coarse to fine, i.e., $\rho_1 > \rho_2 > \cdots > \rho_L$. Coarse layers are intentionally stopped at higher noise levels, so their predictions remain partially stochastic and can preserve multiple possible global plans. Finer layers receive stronger denoising and lower residual noise, progressively instantiating these hypotheses into visual states. We provide the entropy-matched derivation of K_ℓ and its ablation in Appendix C.

3.4 Sparse Hierarchical Attention Pattern

HDR implements hierarchical reasoning with **SHAP** (*Sparse Hierarchical Attention Pattern*), a structured attention mask over flattened tree tokens. Each hierarchy token is indexed by (ℓ, i), where ℓ is the hierarchy level and i is the temporal position within that level. To perform inference autoregressively, we flatten all tree tokens into a coarse-to-fine order using $\phi(\ell, i) = i + \sum_{r<\ell} N_r$, with $s = \phi(\ell, i)$ denoting the flattened token index. Thus, all tokens in $\mathcal{V}^1$ are generated first, followed by $\mathcal{V}^2$, and so on until the finest level $\mathcal{V}^L$. This ordering matches the inference path shown in Figure 2: higher-level tokens are generated before the lower-level tokens that refine them.

For token $v_{\ell,i}$, SHAP first defines its sparse context in tree coordinates:

$$\begin{aligned} \mathcal{A}_{\text{SHAP}}(\ell, i) = \mathbf{1}[i > 1]\{(\ell, i-1)\} \cup \mathbf{1}[i = 1]\{x_{\text{ref}}\} \\ \cup \mathbf{1}[\ell > 1]\{\pi(\ell, i), \text{left}(\pi(\ell, i)), \text{right}(\pi(\ell, i))\}. \end{aligned} \tag{5}$$

Here, x_{ref} is the clean first-frame condition, $(\ell, i-1)$ provides same-level autoregressive continuity, and $\pi(\ell, i)$ is the parent token in the coarser level. The neighboring parent-level tokens $\mathrm{left}(\pi(\ell, i))$ and $\mathrm{right}(\pi(\ell, i))$ provide boundary information from adjacent coarse segments. Invalid boundary indices are omitted. For root-level tokens, which have no parent, the hierarchical context reduces to the first-frame condition and same-level autoregressive context.

The tree context is then converted into a binary attention mask over the flattened sequence. Let $S = \sum_{\ell=1}^{L} N_\ell$ be the total number of hierarchy tokens and $M_{\mathrm{SHAP}} \in \{0,1\}^{S \times S}$ be the SHAP mask. For $s = \phi(\ell, i)$ and $r = \phi(\ell', j)$, we define:

$$M_{\mathrm{SHAP}}[s, r] = \mathbf{1}\left[(\ell', j) \in \mathcal{A}_{\mathrm{SHAP}}(\ell, i)\right]. \tag{6}$$

This mask is sparse by construction: each row contains only a constant number of valid attention targets, independent of video length. It is also autoregressive under the flattened order, because every valid context token is either a previous same-level token, a coarser-level token that has already been generated, or the first-frame condition.

SHAP naturally induces cross-level KV-cache sharing. During inference, once token $v_{\ell,i}$ is generated, its key-value state is written into a global hierarchy cache, denoted by $\mathcal{C}_s = \mathcal{C}_{s-1} \cup \{(k_{\ell,i}, v_{\ell,i}^{\mathrm{KV}})\}$, where $s = \phi(\ell, i)$. When generating a later token $v_{\ell',j}$, HDR retrieves only the cached states selected by the SHAP mask, i.e., $h_{\ell',j} = \mathrm{Attn}(q_{\ell',j}, \{(k_{\ell,i}, v_{\ell,i}^{\mathrm{KV}}) : M_{\mathrm{SHAP}}[\phi(\ell', j), \phi(\ell, i)] = 1\})$. Thus, information generated at coarse levels is reused by lower levels through the shared KV cache, while local temporal continuity is preserved by same-level autoregressive links. Because each token attends to a fixed-size SHAP context rather than dense full-sequence tokens, HDR reduces temporal attention cost while maintaining multi-scale information flow before streaming output.

4 Experiments

We evaluate HDR along two axes: multi-step reasoning ability and low-latency streaming generation efficiency. Section 4.1 introduces the benchmark, metrics, baselines, and implementation setup. Section 4.2 compares HDR with full-attention and streaming baselines. Section 4.3 analyzes the hierarchy layers impact, Section 4.4 tests robustness under reduced denoising budgets and limited data, and Section 4.5 evaluates transfer to real-world robot interaction.

4.1 Benchmarks, Baselines, and Implementation Details

Benchmarks and metrics. Existing video reasoning benchmarks are not fully suitable for evaluating streaming multi-step generation: many do not release complete evaluation scripts, and most emphasize short clips or perceptual consistency rather than logical consistency across multiple reasoning steps. We therefore construct a controlled benchmark suite with six tasks: Tower of Hanoi, maze navigation, one-line drawing, sliding puzzle, Sokoban, and water pouring. The benchmark is stratified by difficulty and includes OOD cases to test rule transfer beyond frequent training patterns. The scored evaluation set contains 370 held-out videos. Each task is evaluated with *success*, which measures exact task completion, and *average progress*, which measures partial progress and reflects intermediate logical consistency. We report task results as *Success / Avg. Progress* pairs and compute overall performance as an unweighted average across the six tasks. Full benchmark construction and task-specific evaluation details are provided in Appendix B.

Baselines and implementation. We compare HDR with full-attention baselines, including bidirectional diffusion and VideoMAE [23], and streaming baselines, including CausalForcing [41] and VideoGPT [30]. For a controlled comparison, the main diffusion baselines and HDR are built on Wan2.2-5B-TI2V [24]; HDR uses six latent hierarchy levels with the entropy-matched denoising schedule $[5, 8, 13, 20, 32, 50]$. All methods are trained on the same 18,000-video reasoning dataset, conditioned on the first frame, and optimized with the same flow-matching training setup. Complete baseline definitions, training details, and implementation settings are provided in Appendix I.

4.2 Main Results: Bridging Reasoning Precision and Streaming Efficiency

HDR improves both the final success of multi-step reasoning trajectories and their intermediate logical consistency. We support this conclusion through quantitative comparisons in Table 1 and

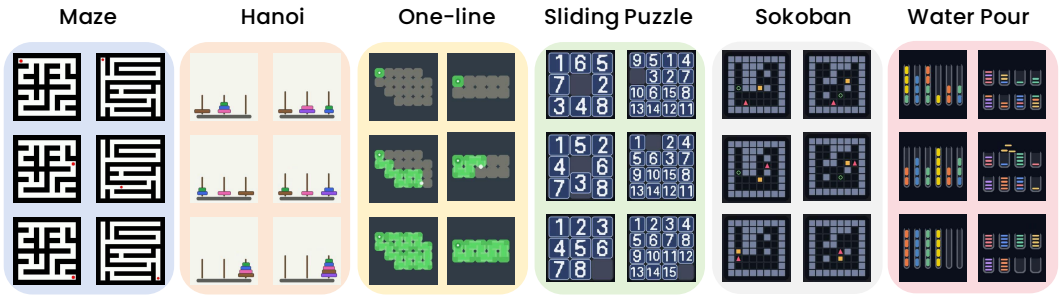


Figure 3: Visualization of the six multi-step video reasoning benchmarks: maze navigation, Tower of Hanoi, one-line drawing, sliding puzzle, Sokoban, and water pouring. Each task requires logical consistency across multiple reasoning steps rather than only local visual plausibility.

Table 1: Main comparison on multi-step video reasoning benchmarks. Scores are reported as mean $\pm$ standard deviation for success and average progress. The best and second-best means are shown in **bold** and underlined, respectively. Full-attention baselines are grayed out. Compared with CausalForcing, HDR improves overall success from 34.22 to 60.29 and average progress from 76.00 to 89.56.

Method	Full Attn.	Metric	Hanoi	Maze	One-line	Sliding	Sokoban	Water	Overall
VideoMAE [23]	✓	Success	22.50±6.18	52.00±8.96	58.33±9.67	1.67±0.00	63.33±9.42	**43.33±4.94**	40.53±6.53
		Avg. Progress	58.23±3.80	93.40±1.32	91.92±1.71	49.93±0.00	**100.00±0.00**	**73.26±2.48**	77.79±1.55
Bidirectional [24]	✓	Success	45.00±6.78	**90.00±3.12**	**81.67±7.60**	31.67±6.98	**90.00±4.21**	21.67±6.83	60.00±5.92
		Avg. Progress	73.58±2.95	**99.89±0.11**	**99.26±0.27**	93.00±1.56	**100.00±0.00**	57.08±3.66	87.13±1.42
VideoGPT [30]	✗	Success	18.75±5.70	22.00±12.83	31.67±7.73	10.00±5.57	15.00±5.46	8.33±3.00	17.57±6.71
		Avg. Progress	25.82±5.46	26.32±12.30	81.74±2.34	27.81±4.01	23.50±4.91	38.03±3.56	36.88±5.43
CausalForcing [41]	✗	Success	45.00±6.73	12.00±8.08	48.33±7.66	21.67±5.16	40.00±10.69	38.33±5.25	34.22±7.26
		Avg. Progress	70.47±2.76	55.04±8.36	95.15±1.28	86.13±2.62	82.25±5.79	66.94±3.03	76.00±3.97
HDR	✗	Success	**58.75±4.50**	78.00±5.25	70.00±10.12	**33.33±7.93**	78.33±9.67	**43.33±4.79**	**60.29±7.04**
		Avg. Progress	**79.62±3.04**	97.18±1.07	97.84±0.61	**93.69±1.47**	99.69±0.31	69.34±2.84	**89.56±1.56**

qualitative examples in Figure 4. We further evaluate streaming efficiency in Table 2, showing that HDR maintains comparable latency to AR diffusion baseline [41].

Overall comparison. Table 1 presents the main comparison on our multi-step video reasoning benchmark. Compared with CausalForcing, the streaming autoregressive diffusion baseline, HDR improves overall success from 34.22 to 60.29, corresponding to a 76.2% relative gain. Average progress also increases from 76.00 to 89.56, indicating stronger intermediate logical consistency. Full-attention baselines, shown in gray, enable dense global interaction but are less aligned with low-latency streaming. Despite not using full temporal attention, HDR achieves the best overall success and average progress, showing that hierarchical denoising can recover strong reasoning precision while preserving the streaming structure of autoregressive diffusion.

Qualitative evidence of revisable planning. Figure 4 compares CausalForcing and HDR on Maze and One-line cases. CausalForcing commits to an early local decision that leads to failure: taking the wrong branch in Maze or missing the top block in One-line. HDR instead resolves the ambiguous decision point through hierarchical planning before committing to fine-grained outputs, leading to successful task completion.

Table 2: Inference speed comparison. Latency is the average time required for each streaming generation step after KV-cache initialization.

Method	Latency
Bidirectional	37.92s
CausalForcing [41]	0.72s
HDR	**0.70s**

Streaming efficiency. We further report inference speed in Table 2. Latency measures the time required for each streaming generation step after KV-cache initialization. HDR achieves comparable latency to CausalForcing, 0.70s versus 0.72s, while being substantially faster than bidirectional diffusion at 37.92s. This shows that HDR improves multi-step reasoning while preserving low-latency streaming behavior.

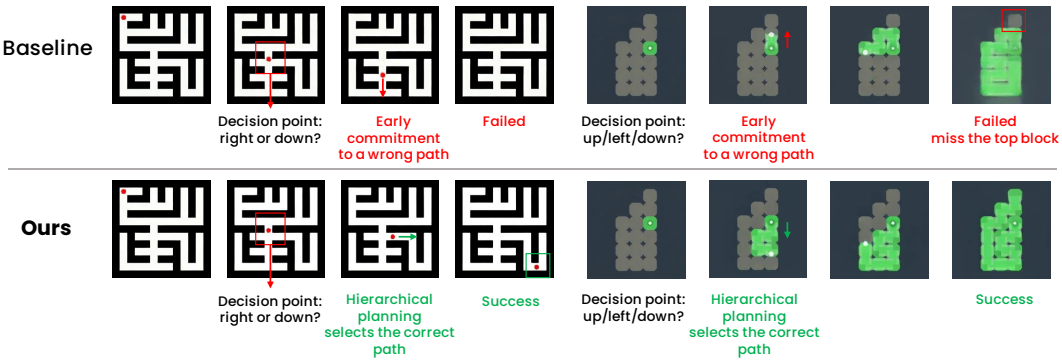


Figure 4: Qualitative comparison between Baseline (CausalForcing) and HDR on Maze and One-line tasks. CausalForcing makes an early local commitment that leads to failure, while HDR performs hierarchical planning before committing to the final trajectory.

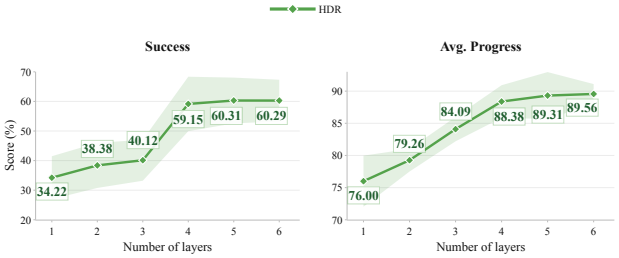


Figure 5: Hierarchical layer importance. Curves show mean performance with one-standard-deviation bands. Increasing active hierarchy layers from 1 to 6 consistently improves HDR over CausalForcing [41], showing that each layer contributes to the full coarse-to-fine reasoning process.

4.3 Mechanism Analysis: Layer-wise Analysis

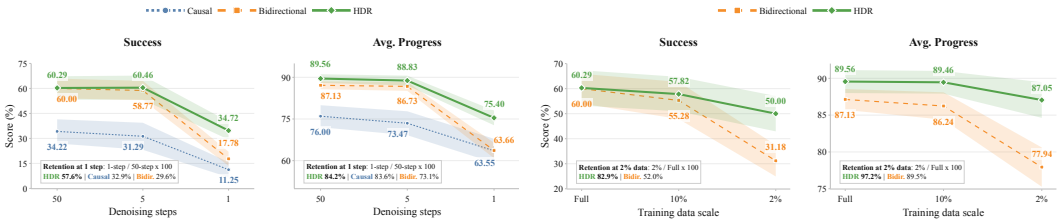
Hierarchical layer importance. We study how performance changes as the number of active hierarchical layers increases. Figure 5 reorganizes the variants by layer count, from a single-layer setting equivalent to CausalForcing to the full six-layer HDR hierarchy. The one-layer setting lacks the coarse-to-fine reasoning process enabled by the hierarchy. As additional layers are introduced, the model gains progressively richer high-level planning and stronger multi-step reasoning behavior. This trend shows that HDR's advantage does not come only from the final frame-level refinement: each hierarchy level contributes useful structure, and deeper hierarchies lead to better performance.

4.4 Robustness: Performance under Reduced Budgets and Limited Data

Denoising-step reduction. We first study robustness under reduced denoising budgets. Figure 6(a) compares CausalForcing, bidirectional diffusion, and HDR as the number of inference denoising steps decreases. Both baselines degrade substantially under aggressive step reduction: bidirectional diffusion drops from 60.00 to 17.78 in overall success with one step, while CausalForcing drops from 34.22 to 11.25. In contrast, HDR remains substantially more robust, achieving 34.72 success with one denoising step and preserving 57.6% of its full-step performance, compared with 29.6% and 32.9% for the bidirectional and CausalForcing baselines, respectively.

Data reduction. We test whether HDR can learn reasoning rules from limited data. Figure 6(b) compares HDR and bidirectional diffusion using the full training set, 10%, and 2% of the data, with task-level results in Appendix Table 10. As data decreases, HDR degrades more gracefully: with only 2% of the training data, it retains 82.9% of its full-data success and 97.2% of its full-data average progress, compared with 52.0% and 89.5% for bidirectional diffusion. This suggests that the hierarchy provides a useful inductive bias for learning transferable task rules rather than simply memorizing training examples.

Together, these ablations show that HDR's robustness comes from its hierarchical architecture. The model remains effective under limited denoising budgets because coarse layers provide stable global guidance, while lower layers refine this guidance into detailed visual states. Conversely, when coarse levels are absent or training data is limited, the hierarchical reasoning process becomes the key factor that determines whether the model can preserve multi-step logical consistency.



(a) Denoising-step reduction. HDR remains more robust than both CausalForcing [41], the streaming AR diffusion baseline, and bidirectional diffusion.

(b) Data reduction. HDR degrades more gracefully than bidirectional diffusion as the training data scale decreases from the full set to 10% and 2%.

Figure 6: Robustness ablations of HDR. Curves show mean performance, and shaded bands denote one standard deviation. The left plot evaluates reduced denoising budgets, and the right plot evaluates reduced training-data scales. Full task-level data-reduction results are provided in Appendix Table 10.

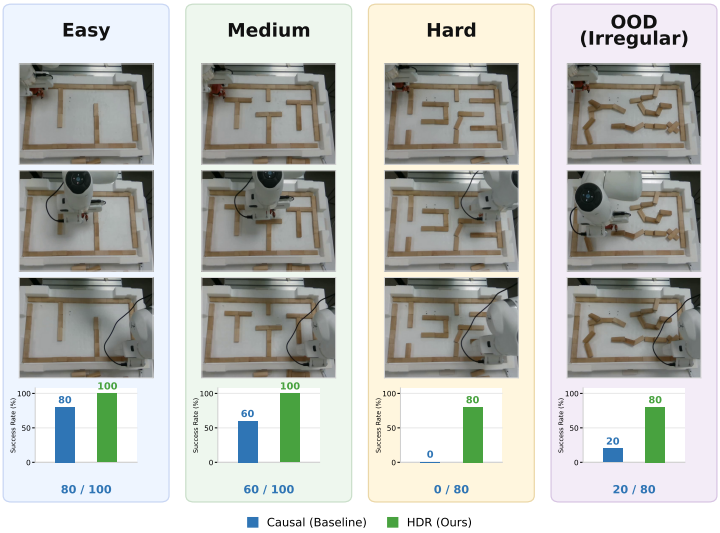


Figure 7: Physical-world robot maze experiment. We fine-tune both CausalForcing [41] and HDR using only 50 real-world robot maze videos, then use an inverse dynamics model (IDM) to convert generated videos into executable robot actions. HDR maintains strong success rates across easy, medium, hard, and OOD mazes, where OOD mazes contain diagonal or stacked wooden blocks.

4.5 Physical World Modeling: Transfer to Robot Interaction

To evaluate whether HDR transfers beyond synthetic benchmarks, we conduct a physical-world robot maze experiment. The task requires a robot arm to pick up a plush toy and move it through a maze built from toy blocks. This setting introduces visual domain shifts, imperfect object localization, occlusion, lighting variation, and irregular maze boundaries. We pretrain HDR on 3,000 virtual maze videos, fine-tune both HDR and CausalForcing using only 50 real-world robot videos, and convert generated videos into executable robot actions using an inverse dynamics model (IDM).

We categorize physical-world mazes by difficulty. Easy, medium, and hard mazes are defined by maze complexity, including the number of branches and turns required by the solution path. We also include an OOD setting, where wooden blocks are placed diagonally or stacked together, producing layouts that differ from regular grid-like training mazes. As shown in Figure 7, HDR maintains strong success rates across all settings, indicating that hierarchical denoising can transfer multi-step reasoning ability to physical interaction.

4.6 World Action Modeling on RoboDojo

We also evaluate whether hierarchical denoising transfers to embodied world-action modeling. We instantiate **HDR-WAM** by combining episode-level visual context with a local action-conditioned rollout, while keeping the detailed token layout, sampling procedure, and attention mask in Appendix A.

Table 3 reports results on the RoboDojo simulation benchmark, which contains 42 robot interaction tasks grouped into five capability dimensions. Without robot-domain or embodied-interaction pretraining, HDR-WAM reaches an overall score of 5.47 and an average success rate of 3.00%. Among

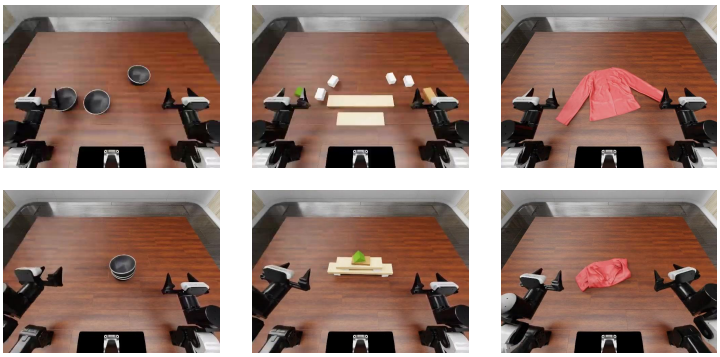


Figure 8: RoboDojo environments and HDR-WAM executions. The figure presents representative tabletop manipulation environments from the RoboDojo simulation benchmark together with executions produced by HDR-WAM. The model achieves particularly strong performance on Long-Horizon tasks, reaching a score of 9.85 and a success rate of 4.75%, demonstrating its ability to maintain coherent task structure over extended interactions.

Table 3: RoboDojo simulation benchmark leaderboard. Each cell reports score / success rate for a capability dimension. Within each pretraining group, the best and second-best scores in each column are shown in **bold** and underlined, respectively, with ties included. The Pretrain column indicates whether the method uses robot-domain or embodied-interaction pretraining beyond task-specific benchmark training. HDR-WAM adapts HDR to World Action Models through hierarchical episode-conditioned visual planning and local action prediction.

Alg.\Dim	Pretrain	Generalization	Precision	Long-Horizon	Memory	Open	Average
X-WAM [6]	Yes	**7.39 / 3.33%**	**6.72 / 1.83%**	**17.47 / 9.08%**	**6.32 / 4.67%**	0.57 / 0.25%	**7.69 / 3.83%**
GigaWorld-Policy [32]	Yes	5.34 / 2.89%	6.15 / 1.83%	15.51 / 8.92%	3.46 / 2.22%	0.54 / 0.50%	6.20 / 3.27%
LDA-1B [19]	Yes	0.71 / 0.17%	3.21 / 0.50%	1.92 / 0.08%	2.08 / 1.78%	0.00 / 0.00%	1.58 / 0.51%
RDT-1B [17]	Yes	0.56 / 0.33%	0.38 / 0.00%	1.13 / 0.00%	0.49 / 0.33%	0.00 / 0.00%	0.51 / 0.13%
H-RDT [1]	Yes	0.49 / 0.22%	0.41 / 0.00%	2.23 / 0.17%	0.12 / 0.11%	0.08 / 0.08%	0.67 / 0.12%
HDR-WAM (Ours)	**No**	4.98 / 3.17%	**5.90 / 2.25%**	**9.85 / 4.75%**	**6.65 / 4.67%**	0.50 / 0.50%	**5.47 / 3.00%**
AHA-WAM [3]	No	**5.79 / 3.28%**	5.86 / 2.42%	8.61 / 2.67%	2.97 / 2.78%	**0.88 / 0.83%**	4.82 / 2.39%
Fast-WAM [38]	No	2.34 / 1.11%	1.96 / 0.00%	9.14 / 5.17%	3.55 / 3.44%	0.42 / 0.42%	3.48 / 2.03%
ACT [40]	No	0.69 / 0.56%	0.85 / 0.00%	1.73 / 0.92%	1.65 / 0.13%	0.00 / 0.00%	0.98 / 0.32%

no-pretraining World Action Model baselines, it improves over AHA-WAM ($4.82/2.39\%$) and Fast-WAM ($3.48/2.03\%$), establishing a new no-pretraining WAM state of the art on the benchmark.

The strongest gains appear in the Long-Horizon and Memory dimensions, where HDR-WAM reaches $9.85/4.75\%$ and $6.65/4.67\%$, respectively. This pattern matches the intended role of hierarchical denoising: sparse episode-level landmarks help preserve coherent task structure over extended interactions, while the local action-conditioned rollout keeps the actor responsive to the current observation. Figure 8 visualizes representative RoboDojo environments and HDR-WAM executions, further illustrating the model's strong long-horizon task capability.

5 Conclusion

We introduced HDR, a framework for long-horizon video reasoning. By organizing video latents into a coarse-to-fine temporal tree, equipping the hierarchy with sparse structured attention, and allocating denoising budgets according to an entropy-matched principle, the method recovers much of the reasoning ability of bidirectional diffusion without relying on full temporal attention.

Empirically, HDR outperforms the causal baseline and remains competitive with bidirectional diffusion across six benchmarks. Ablations show both ingredients matter: removing coarse hierarchy levels hurts, and fully denoising every level is inferior to preserving uncertainty at upper levels.

More broadly, our results suggest that global reasoning in generative video modeling does not necessarily require dense all-to-all temporal computation. Structured multi-scale latent planning provides a promising alternative that preserves causal efficiency while preserving reasoning ability.

References

[1] Hongzhe Bi, Lingxuan Wu, Tianwei Lin, Hengkai Tan, Zhizhong Su, Hang Su, and Jun Zhu. H-rdt: Human manipulation enhanced bimanual robotic manipulation, 2025.

[2] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets, 2023.

[3] Jisong Cai, Long Ling, Shiwei Chu, Zhongshan Liu, Jiayue Kang, Zhixuan Liang, Wenjie Xu, Yinan Mao, Weinan Zhang, Xiaokang Yang, Ru Ying, Ran Zheng, and Yao Mu. Aha-wam:asynchronous horizon-adaptive world-action modeling with observation-guided context routing, 2026.

[4] Haoge Deng, Ting Pan, Haiwen Diao, Zhengxiong Luo, Yufeng Cui, Huchuan Lu, Shiguang Shan, Yonggang Qi, and Xinlong Wang. Autoregressive video generation without vector quantization, 2025.

[5] Kaifeng Gao, Jiaxin Shi, Hanwang Zhang, Chunping Wang, Jun Xiao, and Long Chen. Ca2-vdm: Efficient autoregressive video diffusion model with causal generation and cache sharing, 2025.

[6] Jun Guo, Qiwei Li, Peiyan Li, Zilong Chen, Nan Sun, Yifei Su, Heyun Wang, Yuan Zhang, Xinghang Li, and Huaping Liu. Unified 4d world action modeling from video priors with asynchronous denoising, 2026.

[7] Roberto Henschel, Levon Khachatryan, Hayk Poghosyan, Daniil Hayrapetyan, Vahram Tadevosyan, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Streamingt2v: Consistent, dynamic, and extendable long video generation from text, 2025.

[8] Yicong Hong, Yiqun Mei, Chongjian Ge, Yiran Xu, Yang Zhou, Sai Bi, Yannick Hold-Geoffroy, Mike Roberts, Matthew Fisher, Eli Shechtman, Kalyan Sunkavalli, Feng Liu, Zhengqi Li, and Hao Tan. Relic: Interactive video world model with long-horizon memory, 2025.

[9] Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion, 2025.

[10] Dongya Jia, Zhuo Chen, Jiawei Chen, Chenpeng Du, Jian Wu, Jian Cong, Xiaobin Zhuang, Chumin Li, Zhen Wei, Yuping Wang, et al. Ditar: Diffusion transformer autoregressive modeling for speech generation. In *International Conference on Machine Learning*, pages 27255–27270. PMLR, 2025.

[11] Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong MU, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. In *The Thirteenth International Conference on Learning Representations*, 2025.

[12] Jihwan Kim, Junoh Kang, Jinyoung Choi, and Bohyung Han. Fifo-diffusion: Generating infinite videos from text without training, 2024.

[13] Dan Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, Grant Schindler, Rachel Hornung, Vighnesh Birodkar, Jimmy Yan, Ming-Chang Chiu, Krishna Somandepalli, Hassan Akbari, Yair Alon, Yong Cheng, Josh Dillon, Agrim Gupta, Meera Hahn, Anja Hauth, David Hendon, Alonso Martinez, David Minnen, Mikhail Sirotenko, Kihyuk Sohn, Xuan Yang, Hartwig Adam, Ming-Hsuan Yang, Irfan Essa, Huisheng Wang, David A. Ross, Bryan Seybold, and Lu Jiang. Videopoet: A large language model for zero-shot video generation, 2024.

[14] Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control, 2026.

[15] Zongyi Li, Shujie Hu, Shujie Liu, Long Zhou, Jeongsoo Choi, Lingwei Meng, Xun Guo, Jinyu Li, Hefei Ling, and Furu Wei. Arlon: Boosting diffusion transformers with autoregressive models for long video generation, 2025.

[16] Kunhao Liu, Wenbo Hu, Jiale Xu, Ying Shan, and Shijian Lu. Rolling forcing: Autoregressive long video diffusion in real time, 2025.

[17] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation, 2025.

[18] Yang Luo, Xuanlei Zhao, Baijiong Lin, Lingting Zhu, Liyao Tang, Yuqi Liu, Ying-Cong Chen, Shengju Qian, Xin Wang, and Yang You. V-reasonbench: Toward unified reasoning benchmark suite for video generation models, 2025.

[19] Jiangran Lyu, Kai Liu, Xuheng Zhang, Haoran Liao, Yusen Feng, Wenxuan Zhu, Tingrui Shen, Jiayi Chen, Jiazhao Zhang, Yifei Dong, Wenbo Cui, Senmao Qi, Shuo Wang, Yixin Zheng, Mi Yan, Xuesong Shi, Haoran Li, Dongbin Zhao, Ming-Yu Liu, Zhizheng Zhang, Li Yi, Yizhou Wang, and He Wang. Lda-1b: Scaling latent dynamics action model via universal embodied data ingestion, 2026.

[20] NVIDIA, :, Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, Daniel Dworakowski, Jiaojiao Fan, Michele Fenzi, Francesco Ferroni, Sanja Fidler, Dieter Fox, Songwei Ge, Yunhao Ge, Jinwei Gu, Siddharth Gururani, Ethan He, Jiahui Huang, Jacob Huffman, Pooya Jannaty, Jingyi Jin, Seung Wook Kim, Gergely Klár, Grace Lam, Shiyi Lan, Laura Leal-Taixe, Anqi Li, Zhaoshuo Li, Chen-Hsuan Lin, Tsung-Yi Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Alice Luo, Qianli Ma, Hanzi Mao, Kaichun Mo, Arsalan Mousavian, Seungjun Nah, Sriharsha Niverty, David Page, Despoina Paschalidou, Zeeshan Patel, Lindsey Pavao, Morteza Ramezanali, Fitsum Reda, Xiaowei Ren, Vasanth Rao Naik Sabavat, Ed Schmerling, Stella Shi, Bartosz Stefaniak, Shitao Tang, Lyne Tchapmi, Przemek Tredak, Wei-Cheng Tseng, Jibin Varghese, Hao Wang, Haoxiang Wang, Heng Wang, Ting-Chun Wang, Fangyin Wei, Xinyue Wei, Jay Zhangjie Wu, Jiashu Xu, Wei Yang, Lin Yen-Chen, Xiaohui Zeng, Yu Zeng, Jing Zhang, Qinsheng Zhang, Yuxuan Zhang, Qingqing Zhao, and Artur Zolkowski. Cosmos world foundation model platform for physical ai, 2025.

[21] Zezhong Qian, Xiaowei Chi, Yuming Li, Shizun Wang, Zhiyuan Qin, Xiaozhu Ju, Sirui Han, and Shanghang Zhang. Wristworld: Generating wrist-views via 4d world models for robotic manipulation, 2025.

[22] Robbyant Team, Zelin Gao, Qiuyu Wang, Yanhong Zeng, Jiapeng Zhu, Ka Leong Cheng, Yixuan Li, Hanlin Wang, Yinghao Xu, Shuailei Ma, Yihang Chen, Jie Liu, Yansong Cheng, Yao Yao, Jiayi Zhu, Yihao Meng, Kecheng Zheng, Qingyan Bai, Jingye Chen, Zehong Shen, Yue Yu, Xing Zhu, Yujun Shen, and Hao Ouyang. Advancing open-source world models, 2026.

[23] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training, 2022.

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

[25] Maijunxian Wang, Ruisi Wang, Juyi Lin, Ran Ji, Thaddäus Wiedemer, Qingying Gao, Dezhi Luo, Yaoyao Qian, Lianyu Huang, Zelong Hong, Jiahui Ge, Qianli Ma, Hang He, Yifan Zhou, Lingzi Guo, Lantao Mei, Jiachen Li, Hanwen Xing, Tianqi Zhao, Fengyuan Yu, Weihang Xiao, Yizheng Jiao, Jianheng Hou, Danyang Zhang, Pengcheng Xu, Boyang Zhong, Zehong Zhao,

Gaoyun Fang, John Kitaoka, Yile Xu, Hua Xu, Kenton Blacutt, Tin Nguyen, Siyuan Song, Haoran Sun, Shaoyue Wen, Linyang He, Runming Wang, Yanzhi Wang, Mengyue Yang, Ziqiao Ma, Raphaël Millière, Freda Shi, Nuno Vasconcelos, Daniel Khashabi, Alan Yuille, Yilun Du, Ziming Liu, Bo Li, Dahua Lin, Ziwei Liu, Vikash Kumar, Yijiang Li, Lei Yang, Zhongang Cai, and Hokin Deng. A very big video reasoning suite, 2026.

[26] Ruisi Wang, Zhongang Cai, Fanyi Pu, Junxiang Xu, Wanqi Yin, Maijunxian Wang, Ran Ji, Chenyang Gu, Bo Li, Ziqi Huang, Hokin Deng, Dahua Lin, Ziwei Liu, and Lei Yang. Demystifing video reasoning, 2026.

[27] Thaddäus Wiedemer, Yuxuan Li, Paul Vicol, Shixiang Shane Gu, Nick Matarese, Kevin Swersky, Been Kim, Priyank Jaini, and Robert Geirhos. Video models are zero-shot learners and reasoners, 2025.

[28] Jialong Wu, Xiaoying Zhang, Hongyi Yuan, Xiangcheng Zhang, Tianhao Huang, Changjing He, Chaoyi Deng, Renrui Zhang, Youbin Wu, and Mingsheng Long. Visual generation unlocks human-like reasoning through multimodal world models, 2026.

[29] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*, 2024.

[30] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers, 2021.

[31] Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Yingcong Chen, Yao Lu, Song Han, and Yukang Chen. Longlive: Real-time interactive long video generation, 2025.

[32] Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Hao Li, Hengtao Li, Jie Li, Jindi Lv, Jingyu Liu, Min Cao, Peng Li, Qiuping Deng, Wenjun Mei, Xiaofeng Wang, Xinze Chen, Xinyu Zhou, Yang Wang, Yifan Chang, Yifan Li, Yukun Zhou, Yun Ye, Zhichao Liu, and Zheng Zhu. Gigaworld-policy: An efficient action-centered world–action model, 2026.

[33] Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi "Jim" Fan, and Joel Jang. World action models are zero-shot policies, 2026.

[34] Tianwei Yin, Qiang Zhang, Richard Zhang, William T. Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models, 2025.

[35] Jiashuo Yu, Yue Wu, Meng Chu, Zhifei Ren, Zizheng Huang, Pei Chu, Ruijie Zhang, Yinan He, Qirui Li, Songze Li, Zhenxiang Li, Zhongying Tu, Conghui He, Yu Qiao, Yali Wang, Yi Wang, and Limin Wang. Vrbench: A benchmark for multi-step reasoning in long narrative videos, 2025.

[36] Jiwen Yu, Jianhong Bai, Yiran Qin, Quande Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Xihui Liu. Context as memory: Scene-consistent interactive long video generation with memory retrieval, 2025.

[37] Lijun Yu, José Lezama, Nitesh B. Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, Alexander G. Hauptmann, Boqing Gong, Ming-Hsuan Yang, Irfan Essa, David A. Ross, and Lu Jiang. Language model beats diffusion – tokenizer is key to visual generation, 2024.

[38] Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-wam: Do world action models need test-time future imagination?, 2026.

[39] Kevin Zhang, Kuangzhi Ge, Xiaowei Chi, Renrui Zhang, Shaojun Shi, Zhen Dong, Sirui Han, and Shanghang Zhang. Can world models benefit vlms for world dynamics?, 2025.

[40] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware, 2023.

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

A HDR-WAM Details

A.1 Hierarchical Action Modeling

HDR-WAM adapts HDR to embodied world-action modeling by treating control as a joint denoising problem over visual dynamics and actions. The video stream predicts action-conditioned future observations, while the action stream predicts executable action chunks conditioned on language, proprioception, and visual context.

Temporal views. Consider an episode $\mathcal{E} = \{(x_t, q_t, u_t)\}_{t=0}^{T-1}$, where x_t is the multi-view RGB observation, q_t is proprioception, and $u_t \in \mathbb{R}^{d_a}$ is the low-level action. For a training sample starting at frame s, HDR-WAM constructs two visual views. The episode-level view is a uniformly subsampled set of global anchors

$$\mathcal{I}^{\mathrm{epi}} = \left\{ \left\lfloor \frac{i(T-1)}{N_e - 1} \right\rfloor \right\}_{i=0}^{N_e-1}, \qquad X^{\mathrm{epi}} = \{x_i : i \in \mathcal{I}^{\mathrm{epi}}\},$$

which summarizes task phase and long-range progress. The local action-conditioned view contains a local rollout window plus future visual landmarks. With local length N_ℓ, sampling stride Δ, and N_f future landmarks, we first take

$$\mathcal{I}^{\mathrm{loc}} = \{s + i\Delta\}_{i=0}^{N_\ell - 1}, \qquad t_{\mathrm{end}} = \min(s + (N_\ell - 1)\Delta, T - 1).$$

The future landmark indices are then uniformly sampled after the local window,

$$\mathcal{I}^{\mathrm{fut}} = \left\{ \left\lfloor t_{\mathrm{end}} + \frac{j}{N_f}(T - 1 - t_{\mathrm{end}}) \right\rfloor \right\}_{j=1}^{N_f},$$

with the last available frame repeated when the remaining suffix is shorter than required. The local visual view is $X^{\mathrm{loc}} = \{x_i : i \in \mathcal{I}^{\mathrm{loc}} \cup \mathcal{I}^{\mathrm{fut}}\}$. In our RoboDojo experiments, $N_e = 9$, $N_\ell = 9$, and $N_f = 4$.

Action alignment and token layout. Actions are aligned only to the $N_\ell - 1$ local visual transitions, not to the episode anchors or future landmarks. Given an action horizon H, we split the action chunk into $N_\ell - 1$ groups,

$$A_s = \{u_s, \ldots, u_{s+H-1}\}, \qquad G_r = \{u_{s+rm}, \ldots, u_{s+(r+1)m-1}\}, \qquad m = \frac{H}{N_\ell - 1},$$

where $r = 0, \ldots, N_\ell - 2$. Thus the additional global and future visual tokens provide context but do not introduce extra action targets. After VAE encoding, the two visual views produce latents $Z^{\mathrm{epi}} \in \mathbb{R}^{C \times L_e \times H' \times W'}$ and $Z^{\mathrm{loc}} \in \mathbb{R}^{C \times L_\ell \times H' \times W'}$. We concatenate them along time and keep the first latent of each view clean:

$$Z = [Z^{\mathrm{epi}}; Z^{\mathrm{loc}}], \qquad \mathcal{C} = \{0, L_e\}.$$

For $t \notin \mathcal{C}$, the training input is noised with the video diffusion process; for $t \in \mathcal{C}$, the clean latent is preserved as an observed visual condition. The joint actor sequence is

$$S = [V^{\mathrm{epi}}, V^{\mathrm{loc}}, A],$$

where V^{epi} and V^{loc} are visual tokens obtained from the two latent views, and A denotes action tokens for the grouped action chunk.

Attention mask. HDR-WAM uses a block attention mask over the joint sequence S to separate visual denoising from action prediction. Let $V = V^{\mathrm{epi}} \cup V^{\mathrm{loc}}$, let A be the action-token set, and let $V_{\mathcal{C}} \subset V$ be tokens belonging to the clean visual latents. The MoT self-attention mask can be written as

$$M_{\mathrm{MoT}} = \begin{bmatrix} M_{V \to V} & \mathbf{0}_{V \times A} \\ M_{A \to V} & \mathbf{1}_{A \times A} \end{bmatrix},$$

where $M_{V \to V}$ is the video self-attention pattern, $\mathbf{1}_{A \times A}$ allows action tokens to model the whole action chunk, and $M_{A \to V}$ satisfies $M_{A \to V}(a, v) = 1$ only when $v \in V_{\mathcal{C}}$. Video tokens do not directly

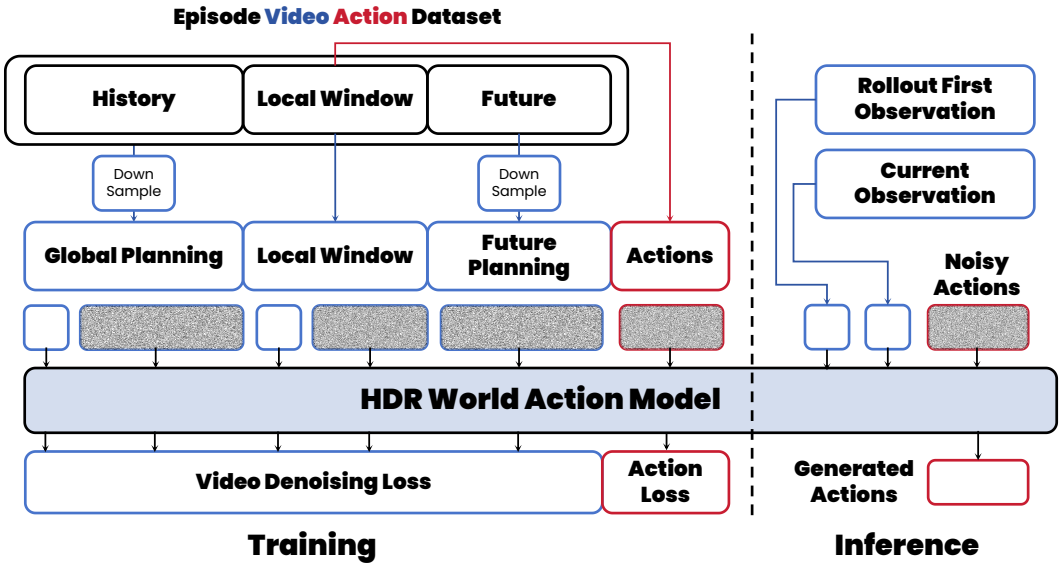


Figure 9: Overview of the HDR-WAM actor. The model combines sparse episode-level visual context with a local action-conditioned rollout, then arranges episode, local visual, and action tokens in a joint sequence. The attention mask separates visual denoising from executable action prediction: action tokens attend to action tokens and clean visual conditions, while visual tokens denoise future observations under language, proprioception, and grouped action context.

attend to action tokens through MoT self-attention; instead, action conditioning enters the video denoiser through cross-attention to grouped action context. For the r-th local visual transition, the causal group mask is

$$M^{\text{cross}}_{r,k} = \mathbf{1}[k \leq r],$$

so denoising a visual transition can use the corresponding and previous action groups while the clean conditioning frames remain unconditioned by future actions. The video loss excludes clean indices $\mathcal{C}$, and the action loss is computed on the action stream with padding masked out.

A.2 RoboDojo Data Processing

For each training sample, we construct both temporal views from the same RoboDojo episode. The global view samples 9 RGB frames uniformly from the entire episode, forming sparse anchors for task-level progress. The local action-conditioned view starts at the current frame, takes 9 consecutive local RGB frames under the dataset stride, and appends 4 future landmarks sampled uniformly between the end of the local window and the end of the episode. If the remaining episode is shorter than required, the final available frame is repeated. Thus each local view contains 13 RGB frames, while action supervision remains aligned to the 8 transitions inside the local 9-frame window.

RoboDojo observations use three camera views. We resize the top camera and the two wrist-side cameras, concatenate the side cameras horizontally, and stack the result under the top view before the standard resize, crop, and normalization pipeline. Both temporal views are encoded with the same video VAE and cached at the episode level, so the global anchors are shared by all samples from the same episode. This avoids repeatedly decoding the same long video while preserving the full episode context needed by HDR-WAM.

B Benchmark and Eval Details

We evaluate all methods on a mixed benchmark containing six long-horizon reasoning tasks: maze navigation, Tower of Hanoi, one-line drawing, sliding puzzle, Sokoban, and water pouring. These tasks cover diverse reasoning patterns, including spatial planning, state transition, object manipulation, and trajectory consistency.

Each generated video is evaluated with task-specific success criteria and reported using two metrics. The success score $S \in \{0, 1\}$ measures exact task completion under the benchmark-specific success criterion. The average progress score $A \in [0, 1]$ measures partial progress toward the target outcome and is therefore more tolerant of minor visual deviations that do not alter the core semantic trajectory. Results in the main paper are reported as Success / Avg. Progress pairs, with both values multiplied by 100.

For a scored set $\mathcal{D}$, the reported task score is computed as the mean over evaluation samples:

$$\text{Success}(\mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} S_i, \qquad \text{Avg. Progress}(\mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} A_i.$$

The overall score is obtained by averaging the task-level scores across the six benchmarks. This gives equal weight to each reasoning task and reflects the model's robustness across different forms of long-horizon video reasoning.

The benchmark uses task-specific difficulty levels. Maze and Hanoi are grouped by native problem size, while the remaining four tasks use level directories. The evaluation split is summarized in Table 4.

Table 4: Difficulty grouping of the mixed reasoning benchmark.

Task	Levels / Sizes	Eval Split	Main Difficulty Factor
Maze	$N = 6, 7, 8, 9, 10$	10 per size	Grid size, path length
Hanoi	$2, 3, 4, 5$ disks	10 ID + 10 OOD per size	Disk count, initial rods
One-line	levels $2, 3, 4$	25 ID + 25 OOD per level	Path length, shape topology
Sliding	levels $2, 3, 4$	30 ID + 20 OOD per level	Board size, optimal moves
Sokoban	levels $2, 3, 4$	30 ID + 20 OOD per level	Layout grammar, action horizon
Water	levels $2, 3, 4$	30 ID + 20 OOD per level	Tubes, capacity, solution length

B.1 Maze Navigation

Maze samples are generated on square grids with fixed start $(0, 0)$ and goal $(N - 1, N - 1)$. The generator constructs a recursive-division maze graph, solves it with breadth-first search, stores the graph edges and solution path, and renders a red ball moving along the path. The default evaluation set contains 50 samples, with 10 samples for each $N \in \{6, 7, 8, 9, 10\}$.

The evaluator tracks the red ball using color segmentation and connected components. The success score requires the decoded route to start correctly, remain in open cells, move only through valid graph edges, and reach the goal. The average progress score is the normalized longest-common-subsequence overlap between the relaxed decoded route $\hat{r}$ and the reference path $r^\star$:

$$A_{\text{maze}} = \frac{\text{LCS}(\hat{r}, r^\star)}{\max(1, |r^\star|)}.$$

B.2 Tower of Hanoi

Hanoi samples use 2–5 disks and three rods. In-domain samples initialize disks on rods $\{0, 1\}$, while OOD samples may initialize disks on $\{0, 1, 2\}$ and require at least one disk to already appear on the goal rod. The goal is always rod 2. For each initial assignment, the generator solves the shortest legal plan with breadth-first search and renders one lifted disk move at a time.

The evaluator decodes disk tracks, infers moves of the form $(\text{disk}, \text{source}, \text{target})$, and simulates them from the ground-truth initial state. The success score requires all inferred moves to be legal and the final state to place every disk on the goal rod. The average progress score measures plan overlap with the reference shortest plan:

$$A_{\text{hanoi}} = \frac{2\,\text{LCS}(\hat{m}, m^\star)}{|\hat{m}| + |m^\star|}.$$

B.3 One-line Drawing

One-line drawing levels $2, 3, 4$ use increasingly larger boards and longer self-avoiding paths. The generator samples a topology family, searches for an adjacent-cell path without revisits, applies random geometric transforms, checks uniqueness and shape constraints, and treats the resulting path as both the occupied shape and the required solution. In the evaluation set, level 2 uses a 9×9 board, level 3 uses 10×10, and level 4 uses 12×12.

The evaluator extracts the drawing head, visited cells, white trace cells, and whether the trace remains on the required shape. The success score requires starting from the highlighted cell, covering every occupied cell, avoiding revisits, avoiding non-adjacent jumps, and never leaving the shape. A small visual bridge tolerance is allowed when the white trace already connects an apparent short jump. The average progress score combines coverage, legal-transition ratio, and on-shape ratio:

$$A_{\text{one}} = 0.5\, C_{\text{cover}} + 0.3\, R_{\text{legal}} + 0.2\, R_{\text{shape}}.$$

B.4 Sliding Puzzle

Sliding puzzle levels 2 and 3 use 3×3 boards, while level 4 uses a 4×4 board. The goal state is the ordered board with the blank tile in the final position. For 3×3 puzzles, states are sampled from an exact solvable pool under bucket constraints; for 4×4 puzzles, states are sampled by backward scrambling from the goal. Each accepted sample is solved optimally and filtered by solution length, family, and diversity constraints.

The evaluator decodes board states from video frames and selects the best legal subsequence starting at the initial state. The success score requires the subsequence to reach the goal, preserve tile inventory, contain no illegal blank moves, and have an unresolved-frame ratio below 0.8. The average progress score combines normalized Manhattan progress, final-state quality, rule adherence, and observability:

$$A_{\text{slide}} = 0.45\, P + 0.20\, Q_{\text{final}} + 0.20\, R_{\text{rule}} + 0.15\, R_{\text{obs}}.$$

B.5 Sokoban

Sokoban samples use an 8×8 grid with walls, one player, one box, and one target. The generator builds layouts from grammar operators, samples valid player and box positions, solves each candidate with shortest-path search, and filters by action length, push count, box-target distance, deadlock-like cells, and diversity. Higher levels use longer and more constrained layouts.

The evaluator decodes player and box positions and checks whether the recovered state sequence can be explained by legal walking and pushing. The success score requires the player and box to start correctly, the box to end on the target, no wall crossing or overlap, no illegal pulling or pushing, and an unresolved-frame ratio below 0.8. The average progress score emphasizes box progress toward the target:

$$A_{\text{sokoban}} = 0.50\, P_{\text{box}} + 0.20\, Q_{\text{final}} + 0.15\, R_{\text{rule}} + 0.15\, R_{\text{obs}}.$$

B.6 Water Pouring

Water pouring samples contain vertical tubes with fixed capacity and colored blocks. The generator constructs a solved goal state, samples a backward chain of legal reverse pours to obtain an initial state, solves or validates the resulting puzzle, and filters candidates by solution length, buried blocks, color runs, branching factor, and state diversity. Higher levels increase tube count, color count, capacity, and solution horizon.

The evaluator extracts tube states from stationary frames. A legal move transfers the complete contiguous top run of one color into an empty tube or onto the same color, limited by remaining destination capacity. The success score requires the final tube state to be solved, all decoded transitions to be legal, block conservation and capacity constraints to hold, and unresolved stationary frames to remain below 0.35. The average progress score combines rule adherence, progress, final-state quality, and observability:

$$A_{\text{water}} = 0.45\, R_{\text{rule}} + 0.35\, P + 0.15\, Q_{\text{final}} + 0.05\, R_{\text{obs}}.$$

C Entropy-Matched versus Fully Denoised Hierarchies

Our final ablation studies how denoising budgets should be allocated across the hierarchy. A naive strategy is to fully denoise every hierarchy level before moving to the next one. In our implementation, this corresponds to assigning 50 denoising steps to all layers. Although intuitive, this strategy ignores the different entropy scales of different hierarchy levels. Coarse layers contain fewer temporal tokens and represent lower-resolution hypotheses, while fine layers contain more tokens and carry more detailed visual entropy. Therefore, forcing all levels to remove the same amount of uncertainty is not well matched to the hierarchical representation.

We instead use an entropy-matched denoising schedule. Let N_ℓ denote the number of tokens at layer ℓ, and let $\tilde{N}_\ell$ denote its effective temporal support. Since a finer layer represents a larger number of temporal degrees of freedom, we assume that the entropy to be removed at layer ℓ should grow with its effective support:

$$\Delta H_\ell \propto \tilde{N}_\ell^\beta,$$

where β is a tunable parameter controlling how aggressively denoising budget increases from coarse to fine layers. Larger β allocates more denoising steps to fine layers, while smaller β makes the schedule more uniform.

We convert this entropy allocation into a layer-wise denoising budget:

$$K_\ell = \left\lceil K_{\max} \left(\frac{\tilde{N}_\ell}{\tilde{N}_L} \right)^\beta \right\rceil,$$

where $K_{\max}$ is the maximum denoising budget used by the finest layer. This rule gives fewer denoising steps to coarse layers and more steps to fine layers, matching the intuition that coarse layers should preserve uncertainty while fine layers should resolve it into concrete visual states.

For the 21-frame setting, our hierarchy has actual layer sizes:

$$N_\ell = [1, 2, 4, 8, 16, 21].$$

The last layer is truncated by the video length, but the underlying binary hierarchy has effective temporal supports:

$$\tilde{N}_\ell = [1, 2, 4, 8, 16, 32].$$

Using $K_{\max} = 50$ and $\beta = 0.66$, the entropy-matched rule gives:

$$K_\ell = \left\lceil 50 \left(\frac{[1, 2, 4, 8, 16, 32]}{32} \right)^{0.66} \right\rceil = [5, 8, 13, 20, 32, 50].$$

This is the default denoising schedule used by HDR in the main experiments.

This schedule can also be interpreted through the lens of uncertainty preservation. Coarse layers remove only a small fraction of their uncertainty, so they act as flexible high-level hypotheses. Fine layers remove much more entropy, allowing them to instantiate these hypotheses into detailed frame-level latents. In contrast, the All-50 strategy sets

$$K_1 = K_2 = \cdots = K_L = 50,$$

which forces every level to remove the same amount of uncertainty regardless of its temporal scale. This prematurely collapses high-level hypotheses and reduces the ability of lower layers to correct or refine them.

Table 5 compares the entropy-matched schedule with fully denoised and alternative schedules. The All-50 variant performs worse overall than the entropy-matched schedule: the overall success score drops from 60.29 to 58.38, and the average progress score drops from 89.56 to 88.21. We also include two additional schedules for completeness: a sparse-to-full schedule $[5, 5, 5, 5, 5, 50]$, which keeps coarse layers minimally denoised before fully denoising the finest layer, and an exponential schedule $[2, 4, 8, 16, 32, 50]$, which increases the denoising budget more aggressively. Their benchmark results are left blank.

Table 5: Entropy-matched versus alternative denoising schedules. The entropy-matched schedule allocates fewer denoising steps to coarse layers and more steps to fine layers according to their entropy scale, achieving the best overall average progress and remaining competitive in overall success. Compared with sparse-to-full and exponential schedules, entropy matching provides a better balance between preserving high-level uncertainty and refining fine-grained video states.

Denoising Strategy	Schedule	Hanoi	Maze	One-line	Sliding	Sokoban	Water	Overall
Entropy-matched	$[5, 8, 13, 20, 32, 50]$	58.75/79.62	78.00/97.18	70.00/97.84	33.33/93.69	78.33/99.69	43.33/69.34	60.29/89.56
All-50	$[50, 50, 50, 50, 50, 50]$	60.00/79.70	76.00/97.18	65.00/97.45	30.00/94.34	80.00/99.60	41.67/65.32	58.38/88.21
Sparse-to-full	$[5, 5, 5, 5, 5, 50]$	55.00/74.58	42.00/92.86	63.33/97.25	41.67/93.76	76.67/99.33	23.33/58.99	50.33/86.13
Exponential	$[2, 4, 8, 16, 32, 50]$	53.75/77.36	58.00/95.53	73.33/98.03	41.67/94.30	86.67/98.73	46.67/68.26	60.02/88.70

D Time Complexity Analysis

We analyze the temporal attention complexity of bidirectional diffusion, Streaming AR Diffusion, and HDR. Let N denote the number of frame-level video tokens and K denote the number of denoising steps. We omit constant factors such as hidden dimension, number of heads, model depth, and the fixed number of tokens attended by HDR.

Bidirectional diffusion. Bidirectional diffusion updates the full video sequence at every denoising step. Each token attends to all N tokens, so one denoising step costs $\mathcal{O}(N^2)$ temporal attention. With K denoising steps, the total complexity is:

$$\mathcal{O}(KN^2).$$

This dense all-to-all interaction supports global reasoning, but it repeatedly recomputes the full temporal attention map throughout inference.

Streaming AR Diffusion. Streaming AR Diffusion generates tokens from left to right. When generating token i, it attends to all previously generated tokens. Although KV caching avoids recomputing previous hidden states, the attention length still grows with time. Thus, the total temporal attention cost is:

$$\mathcal{O}\left(K \sum_{i=1}^{N} i\right) = \mathcal{O}(KN^2).$$

Therefore, Streaming AR Diffusion improves streaming efficiency compared with bidirectional diffusion, but its attention complexity with respect to the number of tokens remains quadratic.

HDR. HDR generates a hierarchical latent tree instead of a flat sequence. For a video with N frame-level tokens, the total number of hierarchy tokens is linear in N; for example, a binary tree contains approximately $2N - 1$ nodes. More importantly, each token attends only to a fixed-size context, including its previous same-layer token, its parent, and neighboring parent tokens. Therefore, the temporal attention cost is linear in the number of hierarchy tokens:

$$\mathcal{O}(K_{\text{avg}}N),$$

where K_{avg} denotes the average denoising budget across hierarchy tokens. Since the hierarchy contains approximately $2N - 1$ tokens for N frame-level tokens, the full attention cost is $\mathcal{O}(K_{\text{avg}}(2N - 1))$, which simplifies to $\mathcal{O}(K_{\text{avg}}N)$. This linear-size hierarchy introduces a longer one-time prefill stage, 16.19s for HDR, compared with 1.48s for bidirectional diffusion and 2.44s for CausalForcing; however, the subsequent streaming latency remains 0.70s per latent and much faster than bidirectional diffusion at 37.92s. In practice, HDR assigns fewer denoising steps to coarse layers, so K_{avg} is smaller than the full denoising budget used by standard diffusion.

This analysis highlights the main computational advantage of HDR. Although HDR introduces additional hierarchy tokens, their number grows only linearly with the video length. Since each hierarchy token attends to a constant-size context, the overall temporal attention cost remains linear in N. In contrast, both bidirectional diffusion and Streaming AR Diffusion have quadratic attention complexity with respect to the number of video tokens.

Table 6: Temporal attention complexity comparison. Bidirectional diffusion uses all-to-all attention, Streaming AR Diffusion attends to the full prefix, while HDR attends to a fixed-size context over a linear-size hierarchy.

Method	Attention Pattern	Token Count	Complexity
Bidirectional	all-to-all	N	$\mathcal{O}(KN^2)$
Streaming AR Diffusion	prefix attention	N	$\mathcal{O}(KN^2)$
HDR	fixed sparse attention	$\approx 2N$	$\mathcal{O}(K_{\text{avg}}N)$

Table 7: Comparison with state-of-the-art closed-source video generation models on video reasoning benchmarks. Scores are reported using the success metric and the average progress metric.

Method	Full Attn.	Fine-tuned	Metric	Hanoi	Maze	One-line	Sliding	Sokoban	Water	Overall
Wan2.6	✓	✗	Success	3.75	6.00	1.67	0.00	0.00	1.67	2.16
			Avg. Progress	20.15	46.93	72.13	61.38	67.56	35.50	49.06
Veo	✓	✗	Success	0.00	0.00	0.00	0.00	0.00	0.00	0.00
			Avg. Progress	0.09	24.42	66.46	0.00	0.33	0.82	14.28
HDR	✗	✓	Success	**58.75**	**78.00**	**70.00**	**33.33**	**78.33**	**43.33**	**60.29**
			Avg. Progress	**79.62**	**97.18**	**97.84**	**93.69**	**99.69**	**69.34**	**89.56**

E Comparison with SOTA Closed-source Models

We further compare HDR with state-of-the-art closed-source video generation models, including Wan2.6 and Veo. Since these models are only accessible through closed-source inference APIs or released inference interfaces, we cannot modify their architectures, apply our hierarchical training strategy, or fine-tune them on the proposed benchmark data. Therefore, we evaluate them in an off-the-shelf setting using the same prompts and evaluation protocol as our method.

As shown in Table 7, off-the-shelf closed-source models achieve limited success under the exact-completion criterion on our reasoning benchmarks. Although they can often generate visually plausible videos, they frequently fail to satisfy the exact multi-step constraints required by these tasks. In contrast, HDR is explicitly trained for HDR and achieves substantially higher scores across all tasks. This comparison highlights that strong general-purpose video generation capability does not necessarily translate into reliable multi-step reasoning under irreversible decision-making.

F Full Table of Ablations

The main paper visualizes the key trends of the denoising-step, hierarchical-layer, and data-reduction ablations in Figure 6a, Figure 5, and Figure 6b. For completeness, we provide the corresponding full numerical results in Table 8, Table 9, and Table 10. Table 8 contains the task-level scores behind Figure 6a, including the full success and average-progress results for the causal baseline, bidirectional baseline, and HDR under different denoising budgets. Table 9 provides the task-level breakdown for Figure 5, organized by the number of active hierarchical layers from 1 to 6, where the 1-layer setting corresponds to the causal baseline. Table 10 reports the task-level results behind Figure 6b, comparing the bidirectional baseline and HDR at different training data scales. All entries are reported as mean $\pm$ standard deviation. For each task-level metric computed over N evaluation samples, we estimate the standard deviation by repeatedly sampling $\lfloor N/2 \rfloor$ examples without replacement, recomputing the mean for 10 trials, and then reporting the population standard deviation across the resulting sample means.

These tables support the same conclusions as the figures while exposing the per-task details. First, the denoising-step ablation shows that HDR retains stronger performance under reduced denoising budgets, especially at the one-step setting. Second, the hierarchical-layer ablation shows that shallower variants stay closer to the causal baseline, while adding more layers generally improves performance and makes HDR's reasoning stronger. Third, the data-reduction ablation shows that HDR degrades

Table 8: Denoising step ablation. Values are reported as mean ± std. HDR remains substantially more robust than the causal baseline under reduced inference budgets.

Method	Step	Metric	Hanoi	Maze	One-line	Sliding	Sokoban	Water	Overall
CausalForcing [41]	50	Success	45.00±6.73	12.00±8.08	48.33±7.66	21.67±5.16	40.00±10.69	38.33±5.25	34.22±7.26
		Avg. Progress	70.47±2.76	55.04±8.36	95.15±1.28	86.13±2.62	82.25±5.79	66.94±3.03	76.00±3.97
	5	Success	38.75±5.14	14.00±11.31	43.33±9.26	20.00±6.00	35.00±10.19	36.67±6.83	31.29±8.12
		Avg. Progress	66.31±2.78	49.67±9.17	94.30±2.00	83.73±2.79	79.81±5.85	66.97±3.26	73.47±4.31
	1	Success	7.50±4.36	10.00±5.63	0.00±0.00	18.33±6.75	28.33±8.92	3.33±1.80	11.25±4.57
		Avg. Progress	39.24±4.57	60.29±9.31	92.82±1.00	79.13±3.56	77.45±5.79	32.40±1.70	63.55±4.32
Bidirectional [24]	50	Success	45.00±6.78	90.00±3.12	81.67±7.60	31.67±6.98	90.00±4.21	21.67±6.83	60.00±5.92
		Avg. Progress	73.58±2.95	99.89±0.11	99.26±0.27	93.00±1.56	100.00±0.00	57.08±3.66	87.13±1.42
	5	Success	41.25±6.91	88.00±2.56	75.00±8.79	35.00±5.74	86.67±4.81	26.67±5.85	58.77±5.78
		Avg. Progress	75.01±3.04	99.75±0.15	98.70±0.72	92.45±1.66	99.11±0.89	55.34±3.16	86.73±1.60
	1	Success	5.00±2.78	0.00±0.00	31.67±9.08	28.33±4.35	41.67±9.17	0.00±0.00	17.78±4.23
		Avg. Progress	43.86±3.13	43.87±4.43	91.32±1.42	94.26±0.62	84.79±5.62	23.86±1.53	63.66±2.79
HDR	Full	Success	58.75±4.50	78.00±5.25	70.00±10.12	33.33±7.93	78.33±9.67	43.33±4.79	60.29±7.04
		Avg. Progress	79.62±3.04	97.18±1.07	97.84±0.61	93.69±1.47	99.69±0.31	69.34±2.84	89.56±1.56
	5	Success	58.75±5.25	74.00±6.93	75.00±8.96	36.67±8.26	75.00±10.62	43.33±3.54	60.46±7.26
		Avg. Progress	79.38±3.56	96.88±1.20	97.88±0.61	94.02±1.59	97.61±1.19	67.18±2.45	88.83±1.77
	1	Success	50.00±6.05	0.00±0.00	45.00±12.03	35.00±6.15	76.67±7.23	1.67±1.00	34.72±5.41
		Avg. Progress	74.03±2.74	57.10±9.21	93.80±1.11	95.30±0.79	99.81±0.19	32.35±1.89	75.40±2.66

Table 9: Hierarchical layer ablation. Values are reported as mean ± std. Results are organized by the number of active hierarchical layers, where the 1-layer setting corresponds to the causal baseline. Adding more layers generally improves reasoning performance, showing that each hierarchical level contributes to HDR's coarse-to-fine reasoning.

Layers	Metric	Hanoi	Maze	One-line	Sliding	Sokoban	Water	Overall
1	Success	45.00±6.73	12.00±8.08	48.33±7.66	21.67±5.16	40.00±10.69	38.33±5.25	34.22±7.26
	Avg. Progress	70.47±2.76	55.04±8.36	95.15±1.28	86.13±2.62	82.25±5.79	66.94±3.03	76.00±3.97
2	Success	31.25±4.19	44.00±7.69	53.33±11.81	20.00±9.03	51.67±6.16	30.00±6.83	38.38±7.62
	Avg. Progress	68.05±2.86	80.60±1.57	97.17±0.57	86.83±1.95	83.72±0.71	59.17±2.91	79.26±1.76
3	Success	38.75±4.58	22.00±8.03	75.00±8.24	30.00±9.75	45.00±7.24	30.00±3.67	40.12±6.92
	Avg. Progress	73.38±3.27	84.76±1.84	98.55±0.46	87.79±2.37	93.72±0.66	66.35±2.72	84.09±1.89
4	Success	61.25±7.69	72.00±10.64	80.00±11.62	33.33±9.57	73.33±10.71	35.00±4.98	59.15±9.20
	Avg. Progress	81.59±3.57	93.33±3.16	98.72±0.60	91.23±2.09	98.71±3.18	66.71±2.65	88.38±2.54
5	Success	57.50±8.22	76.00±9.40	73.33±8.10	33.33±6.57	80.00±8.91	41.67±5.27	60.31±7.74
	Avg. Progress	80.70±5.10	95.40±5.90	98.57±0.93	92.65±1.99	99.25±4.23	69.32±3.73	89.31±3.65
6	Success	58.75±4.50	78.00±5.25	70.00±10.08	33.33±7.93	78.33±9.67	43.33±4.79	60.29±7.04
	Avg. Progress	79.62±3.04	97.18±1.07	97.84±0.46	93.69±1.48	99.69±0.31	69.34±2.90	89.56±1.54

more gracefully than the bidirectional baseline as the amount of training data decreases. Together, the full tables confirm that HDR's robustness comes from both its hierarchical coarse-to-fine structure and its ability to preserve useful reasoning behavior under limited denoising computation and limited data.

G Visualization of Hierarchical Intermediate Predictions

We provide qualitative visualizations of the intermediate predictions produced by HDR during hierarchical reasoning. As shown in Figure 10, the columns labeled L1, L2, and L3 correspond to the model's clean x_0 predictions before noise is added at different hierarchy levels. These intermediate outputs reveal the coarse-to-fine inference process of HDR: higher layers first capture coarse task structure and global plans, while lower layers progressively refine them into concrete video states.

Table 10: Data reduction ablation. Values are reported as mean ± std. We train the bidirectional baseline and HDR using the full training set, 10% of the training set, and 2% of the training set. This experiment evaluates whether the models can learn reasoning rules from limited data.

Method	Data Scale	Metric	Hanoi	Maze	One-line	Sliding	Sokoban	Water	Overall
Bidirectional [24]	Full	Success	45.00±6.78	90.00±3.12	81.67±7.60	31.67±6.98	90.00±4.21	21.67±6.83	60.00±5.92
		Avg. Progress	73.58±2.95	99.89±0.11	99.26±0.27	93.00±1.56	100.00±0.00	57.08±3.66	87.13±1.42
	10%	Success	55.00±5.36	80.00±9.94	80.00±9.13	26.67±7.24	70.00±6.67	20.00±6.15	55.28±7.42
		Avg. Progress	73.94±3.00	95.08±2.03	98.70±0.59	93.45±1.08	99.08±0.90	57.20±3.55	86.24±1.86
	2%	Success	43.75±6.74	20.00±7.76	76.67±9.27	16.67±3.27	23.33±7.42	6.67±3.14	31.18±6.27
		Avg. Progress	72.54±2.59	67.76±6.35	97.69±0.66	92.51±0.71	90.11±3.32	47.03±2.39	77.94±2.67
HDR	Full	Success	58.75±4.50	78.00±5.25	70.00±10.12	33.33±7.93	78.33±9.67	43.33±4.79	60.29±7.04
		Avg. Progress	79.62±3.04	97.18±1.07	97.84±0.61	93.69±1.47	99.69±0.31	69.34±2.84	89.56±1.56
	10%	Success	56.25±5.78	64.00±9.95	68.33±8.34	38.33±5.86	80.00±5.83	40.00±5.57	57.82±6.89
		Avg. Progress	79.26±3.55	93.63±2.05	97.91±0.55	94.40±0.82	99.92±0.08	71.67±2.17	89.46±1.54
	2%	Success	60.00±5.59	50.00±13.43	50.00±7.98	26.67±5.71	70.00±7.20	43.33±2.49	50.00±7.07
		Avg. Progress	77.71±3.75	86.78±6.54	96.40±0.90	93.76±0.68	98.82±0.47	68.84±2.18	87.05±2.42

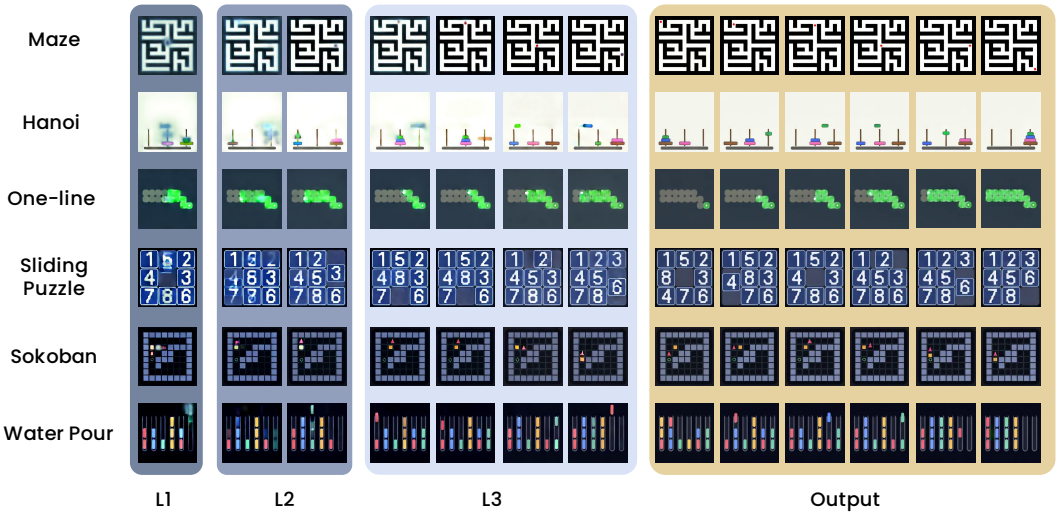


Figure 10: Visualization of HDR's hierarchical intermediate predictions. Columns labeled L1, L2, and L3 show clean x_0 predictions before noise is added at different hierarchy levels, while the rightmost block shows the final generated output. The progression from coarse layers to fine layers illustrates how HDR first forms high-level plans and then refines them into detailed video states.

This supports our claim that HDR enables the model to form high-level hypotheses before committing to final streaming outputs.

H Failure Case Studies

This appendix provides qualitative case studies that complement the quantitative results in the main paper. Section H.1 compares HDR with the streaming autoregressive diffusion baseline and illustrates how hierarchical denoising helps avoid early irreversible mistakes. Section H.2 then examines representative residual failures of HDR, focusing on state-consistency drift in maze navigation, Sokoban, and water pouring.

H.1 Qualitative Comparison with Streaming AR Diffusion

We first provide representative qualitative cases where the streaming autoregressive diffusion baseline, represented by CausalForcing, fails while HDR succeeds. These examples complement the benchmark results in the main paper by showing a recurring failure mode: the baseline makes an early, locally plausible decision, but its irreversible temporal commitment prevents later frames from repairing the

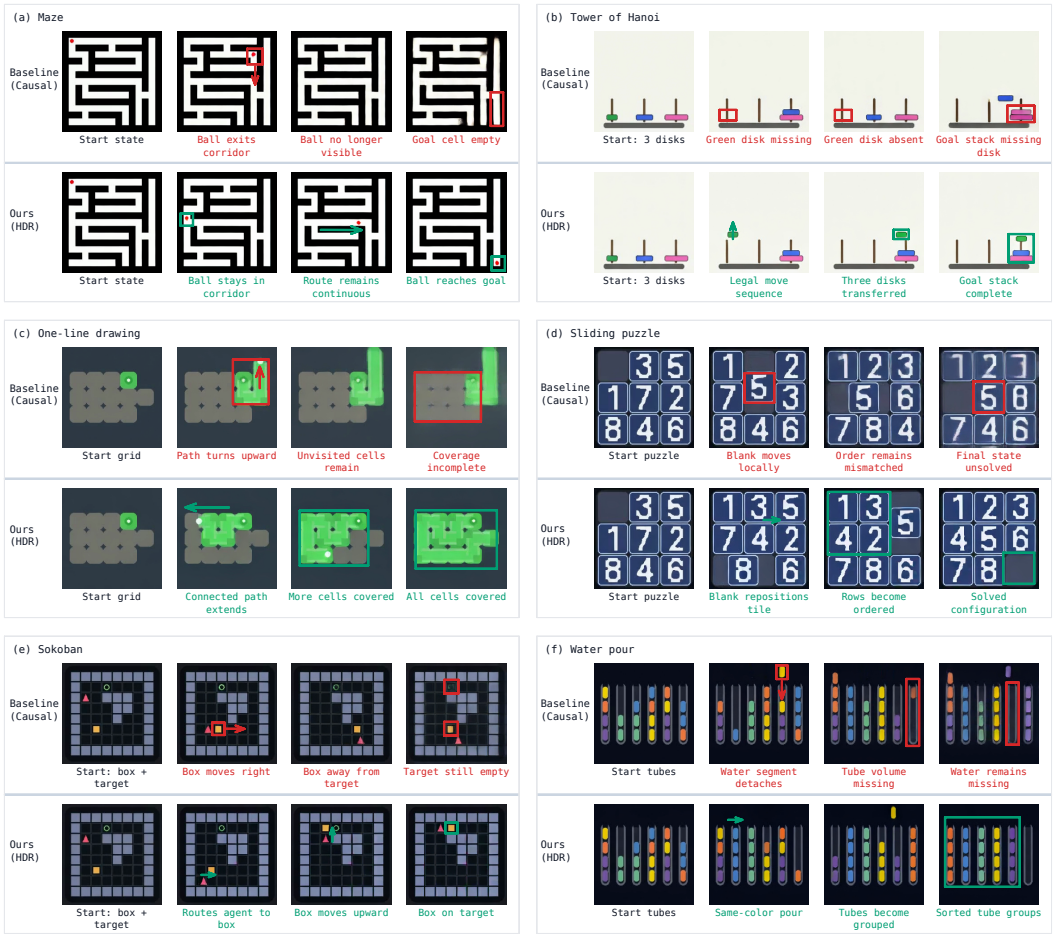


Figure 11: Qualitative comparison across all six reasoning tasks where the streaming autoregressive diffusion baseline fails but HDR succeeds. For each task, the top row shows CausalForcing and the bottom row shows HDR. Red boxes and arrows mark the baseline's failure points, while green boxes and arrows highlight HDR's successful coarse-to-fine refinement. Across tasks, the baseline typically commits to an early local mistake, such as leaving the valid maze corridor, missing a required disk transfer, breaking path continuity, misplacing the blank tile, pushing the box away from the goal, or violating water-pouring constraints. In contrast, HDR maintains globally consistent task structure and reaches the correct final state.

trajectory. Once the rollout deviates from the correct plan, subsequent frames inherit the error and the video ends in a globally inconsistent state.

By contrast, HDR maintains revisable high-level hypotheses at coarse hierarchy levels and refines them through coarse-to-fine denoising before committing to the final streaming output. This hierarchical inference process enables implicit backtracking in latent space: the model can preserve uncertainty about multi-step structure, revise an incorrect partial plan, and instantiate a rule-consistent solution at finer levels. Figure 11 illustrates this behavior across maze navigation, Tower of Hanoi, one-line drawing, sliding puzzle, Sokoban, and water pouring.

H.2 Residual Failure Modes of HDR

Although HDR substantially improves multi-step video reasoning, it can still fail when exact state consistency must be preserved until the final frames. Figures 12, 13, and 14 show three representative residual failure modes. In maze navigation, HDR follows a plausible route but a wall disappears near the end of the rollout, making the final maze state invalid. In Sokoban, the box trajectory largely approaches the target, but a late wall-disappearance artifact breaks physical consistency. In water

pouring, the grouping process is mostly correct, yet one liquid color collapses into another, producing a visually plausible but semantically invalid final arrangement.

These failures suggest that HDR's remaining errors are less often complete planning failures and more often late-stage state-consistency drift. The coarse hierarchy levels typically encode the intended multi-step structure, and the finer levels refine this intent into nearly correct video states. The residual errors appear near the end of rollout, where the model must preserve exact geometry, object identity, or terminal constraints. This suggests that future improvements should focus on constraint-aware refinement or lightweight verification during the final generation stage.

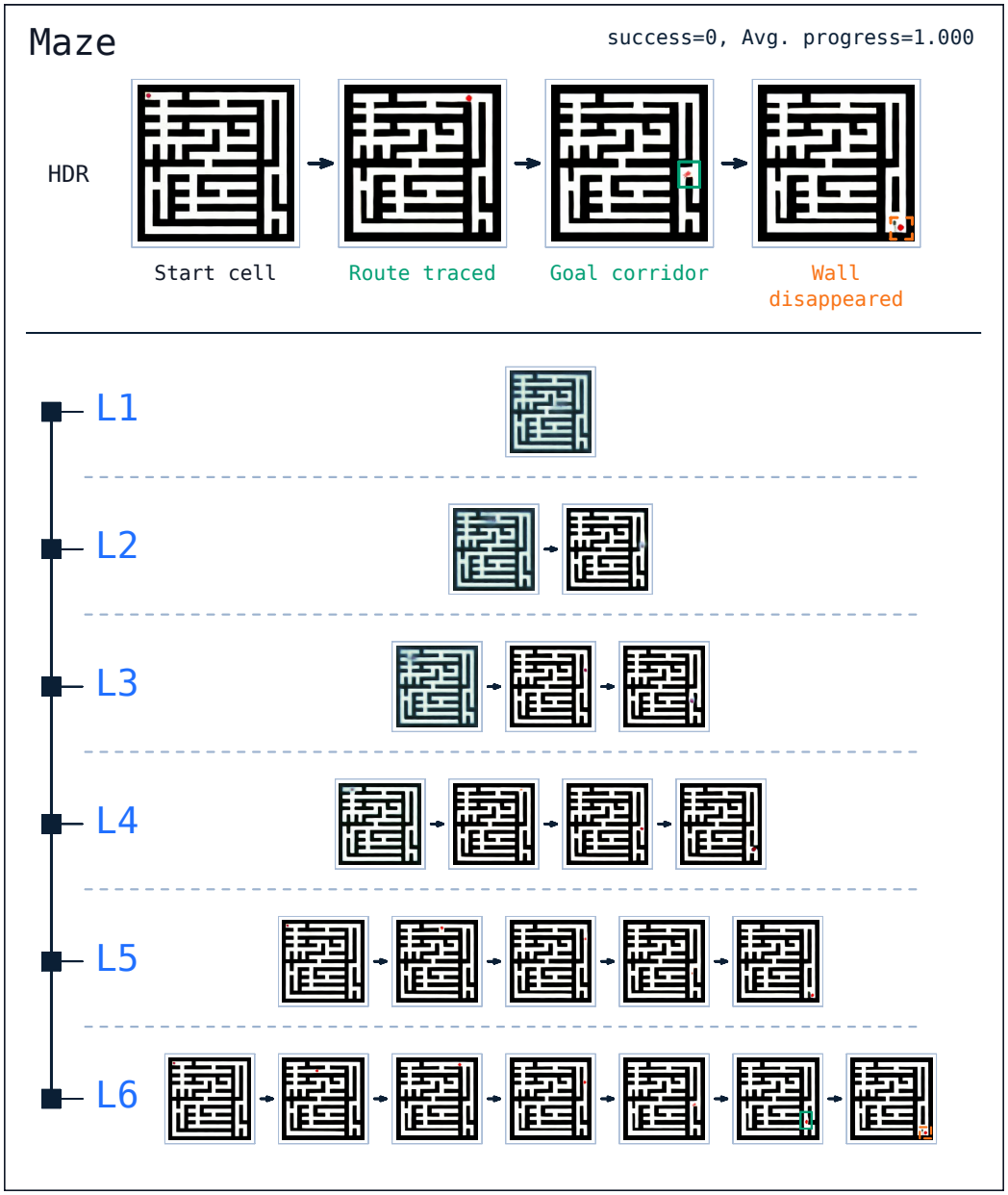


Figure 12: Maze failure case of HDR. The model traces a plausible route and reaches the goal corridor, but a wall disappears near the end of the rollout, producing an invalid maze state despite otherwise correct multi-step planning.

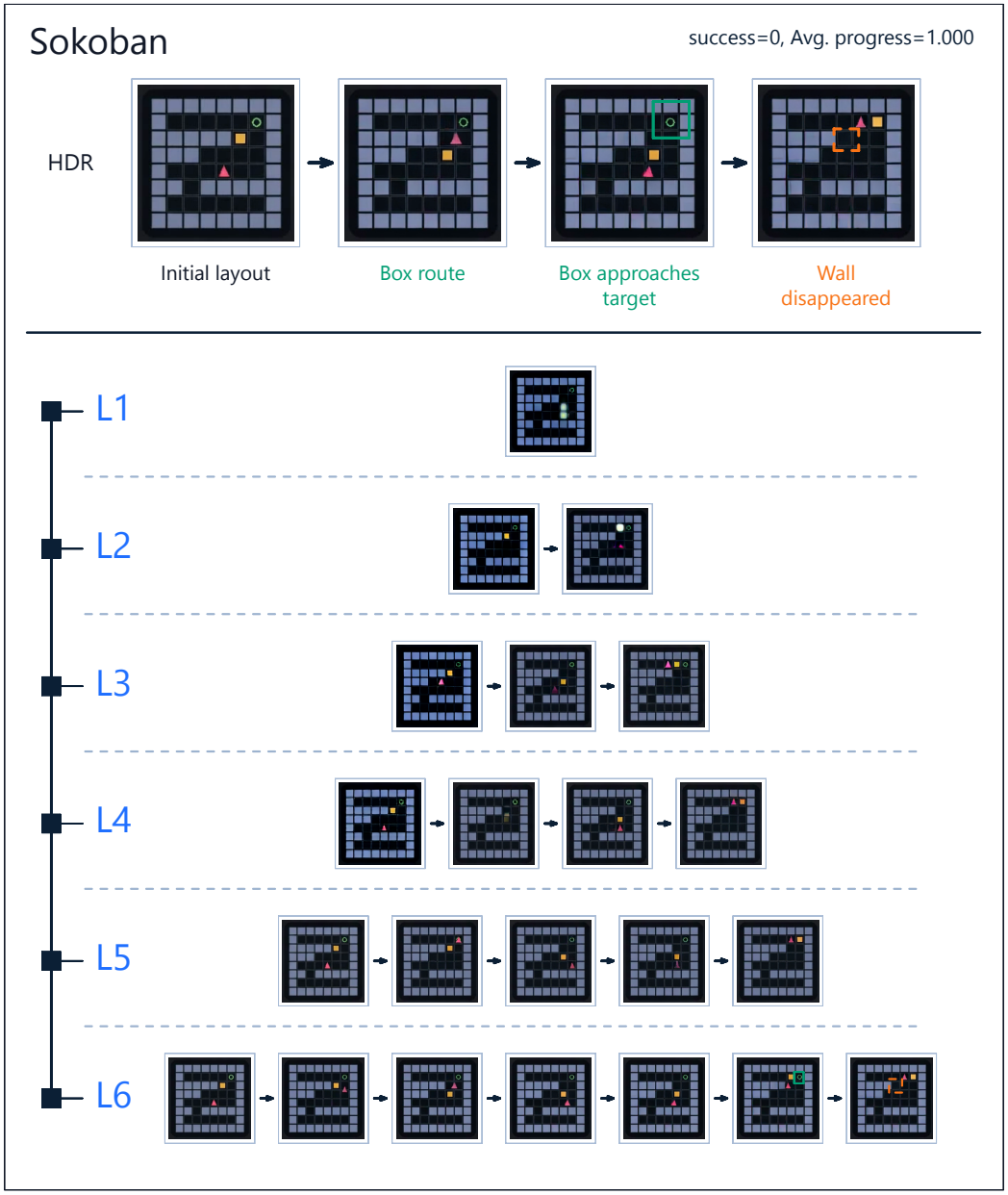


Figure 13: Sokoban failure case of HDR. The box trajectory is largely correct and approaches the target, but a wall disappears near the end of the rollout, making the final scene physically inconsistent.

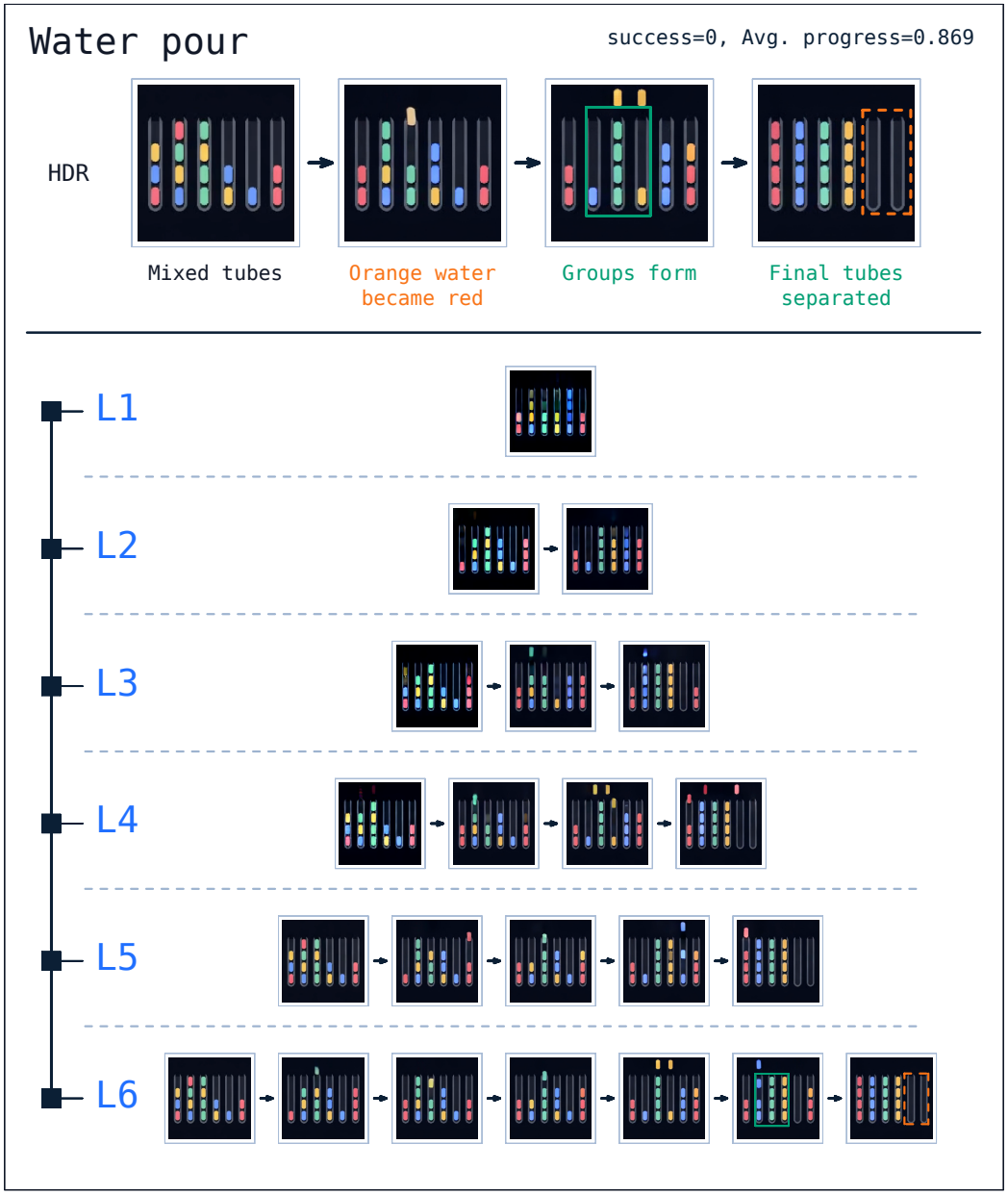


Figure 14: Water-pouring failure case of HDR. The model nearly completes the tube grouping correctly, but orange liquid collapses into red, yielding a visually plausible yet invalid final arrangement.

I Baselines and Implementation Details

Baselines. We compare HDR against three families of baselines. The first family consists of full-attention methods, including bidirectional diffusion and VideoMAE [23]. These methods allow global temporal interactions and therefore provide strong reasoning-oriented comparisons, but they require dense sequence processing. For VideoMAE, we enable image-to-video generation by randomly masking 90%–100% of tokens during training and masking all frames except the first frame at inference. The second family consists of non-full-attention methods, including causal diffusion, autoregressive generation, and VideoGPT [30]. These methods are more efficient and better aligned with streaming generation, but they are more vulnerable to irreversible early mistakes. For a controlled comparison, both the bidirectional diffusion baseline and the causal diffusion baseline are built on Wan2.2-5B-TI2V [24]. The bidirectional baseline follows the original full-attention inference paradigm of the backbone. The causal baseline replaces full temporal attention with a causal attention mask and is trained using the teacher-forcing strategy from Causal Forcing [41]. This setup isolates the effect of temporal attention and causal rollout while keeping the underlying generative backbone comparable.

HDR implementation. Our implementation is also built on Wan2.2-5B-TI2V and augments the backbone with the hierarchical latent tree described in Section 3. For the 125-frame setting used in the main experiments, we employ six hierarchy levels in latent space with sizes $1, 2, 4, 8, 16, 32$ and the entropy-matched denoising schedule $[5, 8, 13, 20, 32, 50]$. Unless otherwise stated, this is the default setting for HDR. We provide the derivation and ablation of this entropy-matched schedule in Appendix C. For training, we use a mixed reasoning dataset containing 18,000 video clips in total, consisting of 3,000 synthetic videos for each of the six tasks. All videos are resized to 224×224. We train from raw videos rather than precomputed latents: videos are decoded online and encoded by the Wan2.2 TI2V VAE during training. We condition on the first frame and allow this condition to be visible to all hierarchy levels. The model is initialized from a pretrained Wan2.2-5B-TI2V checkpoint and optimized with the flow-matching objective for 100,000 training steps. We use AdamW with learning rate 2×10^{-6}, betas $(0, 0.999)$, weight decay 0.01, bfloat16 mixed-precision training, gradient checkpointing, and hybrid full-sharding FSDP. The per-device batch size is 1 and the total batch size is 8. During inference, we use classifier-free guidance with scale 3.0. Each main training run uses 8 GPUs and takes approximately 48 hours. All baselines are trained and evaluated under the same configuration for fair comparison.