SCALEWORLD: HIGH-RESOLUTION WORLD GENERATION THROUGH SHARED PREDICTION AND REFINEMENT

Anonymous authors
Paper under double-blind review

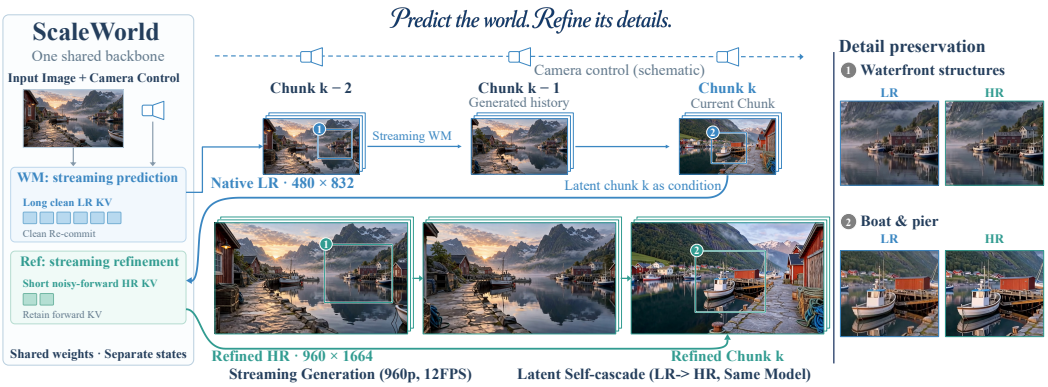

Figure 1: SCALEWORLD **uses one shared backbone for streaming world prediction and high-resolution refinement.** Each completed low-resolution chunk conditions its high-resolution counterpart, while separate causal states retain a long clean prediction history and a short noisy-forward refinement history. Matched crops highlight the recovered structures and textures.

ABSTRACT

High-resolution generation is crucial for camera-controlled world models, yet direct high-resolution rollout is costly and often degrades visual details. We present SCALEWORLD, a causal video model that learns world prediction and high-resolution refinement in shared parameters. This allows the model to refine its own predictions for improved visual detail while reducing the cost of high-resolution generation. Specifically, each low-resolution chunk is completed through four-step camera-controlled prediction and then conditions a second invocation of the same backbone for one-step refinement. Realizing this process requires addressing two core challenges: (i) **Efficiency.** Separate causal states maintain a long prediction context at low resolution and a short refinement context at high resolution, reducing context cost. (ii) **Capability integration.** We separate specialist optimization from consolidation and introduce task-asymmetric on-policy distillation (TA-OPD), which transfers prediction and refinement behaviors into a shared student under their respective sampling and history policies. Extensive experiments show that SCALEWORLD effectively unifies prediction and refinement while reducing the context-storage cost of high-resolution world generation.

1 INTRODUCTION

Camera-controlled world models generate visual futures from an initial observation and camera commands (Sun et al., 2025; Gao et al., 2026; Qian et al., 2026). High-resolution world generation requires fine detail to remain consistent with scene evolution and control. Although causal video diffusion enables incremental generation (Yin et al., 2025; Huang et al., 2025), increasing spatial

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

resolution raises both denoising cost and key–value (KV) storage, restricting the temporal context available under a fixed memory budget.

Coarse-to-fine pipelines address this cost by predicting at low resolution and refining the result with a dedicated model (Luxury et al., 2026; Zhu et al., 2026a). Prediction and refinement share visual priors for structure, appearance, and motion, suggesting that both capabilities could reside in one backbone. Cross-resolution reuse has been explored by Self-Cascade, Pyramid Flow, and HiStream (Guo et al., 2024; Jin et al., 2024; Qiu et al., 2025). We therefore ask: *can high-resolution refinement become a world model's own capability while retaining its native camera-controlled prediction?*

We present SCALEWORLD, a self-cascading world model that first completes a camera-controlled low-resolution (LR) prediction, then conditionally refines it at high resolution (HR) using the same causal backbone. This design poses two challenges. **(i) Efficiency.** Combining long prediction histories with spatially expensive HR refinement requires careful allocation of computation and temporal context. **(ii) Capability integration.** Sharing visual representations does not ensure that prediction and refinement can be effectively distilled together. In our mixed-DMD experiments, jointly optimizing prediction and refinement exhibits negative cross-task gradient alignment and competing quality trends: improvements in SR reconstruction can accompany degraded temporal stability in world prediction (Appendix C.1). The challenge is therefore to consolidate both capabilities while retaining the sampling and temporal-context policies required by each task.

To address efficiency, we exploit the roles' different information requirements through asymmetric temporal contexts. World prediction retains a longer clean history at low resolution to infer unobserved scene evolution. Refinement receives the current LR structure, content, and motion, and uses a shorter *noisy-forward KV* history to support detail continuity without an additional HR re-encoding pass. Both roles reindex temporal rotary positional embeddings (RoPE) within their windows to avoid positional extrapolation beyond the training range. The LR output remains independently usable, and HR refinement does not feed back into prediction history. Sharing the backbone also avoids storing a separate refinement model.

To address capability integration, we separate specialist optimization from capability consolidation. Starting from a jointly trained and causally adapted model, we train prediction and refinement specialists with distribution matching distillation (DMD) (Yin et al., 2024), allowing each capability to be optimized under its own sampling and history policy. Building on existing on-policy distillation methods (Li et al., 2026; Zhou et al., 2026), we then consolidate the frozen specialists into a shared student through *task-asymmetric on-policy distillation* (TA-OPD). As updates to shared parameters change the student's generation trajectories and recurrent states, supervision on fixed expert trajectories may become misaligned with the states encountered by the student. TA-OPD instead matches expert velocities at student-visited denoising queries for prediction, and at matched conditional noise queries with network-specific recurrent histories for refinement. This formulation transfers both specialist behaviors while respecting their different execution policies.

2 RELATED WORK

Due to space constraints, we defer a detailed discussion of related work to the Appendix E.

3 METHOD

Figure 2 summarizes how SCALEWORLD learns to predict and refine with one backbone. We first define the shared generation process, then describe how specialists learn under asymmetric temporal contexts and how TA-OPD consolidates their behaviors. Streaming inference composes the two capabilities using the same context policies.

3.1 PREDICTION–REFINEMENT FACTORIZATION

Given an initial observation I, text c, and a camera trajectory a, we generate a high-resolution world through two successive computations: camera-controlled prediction at low resolution, followed by conditional refinement of the completed prediction. Let $\widehat{z}_k^L$ and $\widehat{z}_k^H$ denote the generated low- and

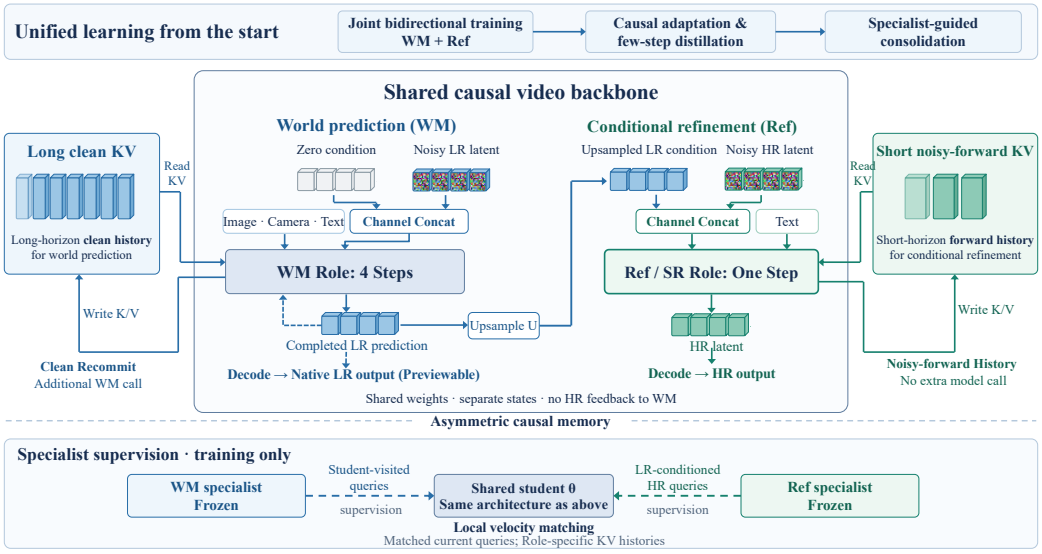

Figure 2: **Overview of** SCALEWORLD**. Top:** Joint prediction and refinement training precedes causal adaptation, few-step specialist training, and TA-OPD. **Center:** One shared backbone completes each camera-controlled LR prediction and then refines it using the upsampled latent as a condition. WM maintains longer clean history; refinement retains shorter noisy-forward history. Both roles reindex temporal RoPE within their attention windows to avoid positional extrapolation. HR outputs do not enter the WM history. **Bottom:** During consolidation, frozen specialists supervise the shared student through local velocity matching, with matched current queries and network-specific KV histories.

high-resolution latents of chunk k. The two samplers share a velocity network v_θ:

$$\hat{z}_k^L = S_\theta^{\mathrm{WM}}(\epsilon_k^L; I, c, a_{\le k}, M_{k-1}^{\mathrm{WM}}), \tag{1}$$

$$\hat{z}_k^H = S_\theta^{\mathrm{Ref}}(\epsilon_k^H; c, U(\hat{z}_k^L), M_{k-1}^{\mathrm{Ref}}). \tag{2}$$

Here U spatially upsamples the LR latent, ϵ_k^L and ϵ_k^H collect each sampler's random draws, and M^{WM} and M^{Ref} are separate causal states. We denote prediction by WM and refinement by Ref; the latter is learned through video super-resolution (SR).

Each LR chunk is a completed, independently usable world prediction. It establishes the scene layout, content, and camera-induced motion that condition HR synthesis. Refinement learns to enrich this realization with fine detail while preserving its structure and dynamics. This division gives the two roles different information: prediction must infer an unobserved future, while refinement observes the current scene through its LR condition. Refinement uses no future LR chunks, and HR outputs do not enter the prediction history.

For fixed parameters, a supplied camera trajectory, and independent sampling streams, this construction induces

$$p_\theta(z^L, z^H \mid I, c, a) = p_\theta^{\mathrm{WM}}(z^L \mid I, c, a) \cdot p_\theta^{\mathrm{Ref}}(z^H \mid z^L, c). \tag{3}$$

These sequence distributions marginalize internal sampling variables and states; refinement is executed chunk causally as in Eq. 2. This factorization allows prediction context to remain at low resolution while the same model produces high-resolution output.

3.2 SHARED CAUSAL VIDEO MODEL

Shared visual interface. We implement both samplers with a single Wan2.2 TI2V-5B backbone, sharing the transformer and output head. A 48-channel upsampled LR latent is concatenated with the noisy target before patch embedding. WM sets these additional channels to zero and uses image

Table 1: Prediction and refinement interfaces in the shared model.

Property	Prediction (WM)	Refinement (Ref)
Output resolution	480×832	960×1664
Conditions	image, camera, text	LR latent, text
Denoising evaluations	4	1

anchoring, text, and camera control. We inject the camera poses using Unified Camera Positional Encoding (UCPE) (Zhang et al., 2026). Refinement supplies the LR latent and text, with no direct camera input or clean HR anchor. The input projection is expanded by copying its original weights and zero-initializing the added condition weights. The resulting zero-conditioned interface preserves the original WM function at initialization (Appendix A.6).

Sampling. Table 1 summarizes the two invocations. For $m \in \{\mathrm{WM}, \mathrm{Ref}\}$, let c_m collect the corresponding conditions. The shared network uses the flow parameterization $z_t = (1-\sigma_t)z_0 + \sigma_t\epsilon$ and predicts

$$\widehat{z}_0 = z_t - \sigma_t v_\theta(z_t, t; c_m, M^m). \tag{4}$$

WM uses four denoising queries. Between queries, its current clean prediction is re-noised at the next noise level:

$$z^{\theta,\mathrm{WM}}_{k,j+1} = (1-\sigma_{j+1})\widehat{z}^{\theta,\mathrm{WM}}_{k,j} + \sigma_{j+1}\epsilon_{k,j+1}. \tag{5}$$

Here j indexes denoising queries and $\widehat{z}^{\theta,\mathrm{WM}}_{k,j}$ is the clean prediction at query j. The last prediction is the completed LR chunk. Refinement uses a single noise query conditioned on the LR chunk, directly producing the HR prediction.

The two roles share parameters while maintaining distinct temporal states. Each learns to generate under its own context policy during specialist training, and uses the same policy during consolidation and inference, as described next.

3.3 Specialist Training and Consolidation

Learning specialists for causal generation. Our training procedure separates learning a shared representation, optimizing specialist behaviors, and consolidating them for deployment. Joint bidirectional training first equips the model with prediction and refinement interfaces, followed by causal adaptation and consistency distillation. Directly optimizing task-specific distillation and reconstruction objectives in this shared model can introduce conflicting gradient directions and imbalanced gradient magnitudes (Appendix C.1), motivating separate specialist optimization. Section 4.3 describes the training-recipe comparisons used to assess this choice. From the shared foundation, we train a prediction specialist ϕ_{WM} and a refinement specialist ϕ_{Ref} using distribution matching distillation (DMD) (Yin et al., 2024). Each generator is optimized through causal rollouts under the context policy of its task. The resulting few-step generators become the frozen teachers for TA-OPD. This separates learning effective specialist behaviors from consolidating them into one model. Three complementary choices govern their temporal context: attention extent, temporal position indexing, and history construction.

Asymmetric context windows. Prediction must infer scene evolution beyond the observed view, so its history supports persistent world structure and motion. Refinement already receives the current LR realization of that world; its temporal context primarily supports the continuity of newly synthesized detail. We therefore allocate a longer attention window to WM and a shorter window to Ref. Keeping the longer history at low resolution concentrates context on world prediction while limiting the cost of HR attention. We use this asymmetry during DMD specialist training and retain it during TA-OPD and streaming inference. The reference consolidated model uses capacities of 25 and 8 latent frames for WM and Ref, respectively, including the current chunk and any retained anchor.

RoPE reindexing for temporal extrapolation. Training observes finite temporal spans, whereas streaming generation can continue far beyond them. A bounded attention window alone does not prevent absolute temporal indices from growing, exposing the model to RoPE positions outside its training range. We address this source of training–inference mismatch by reindexing temporal RoPE within the active window of *both WM and Ref*. As the window advances, retained history and the current chunk are expressed in local temporal coordinates. Cached keys are transformed to the

corresponding RoPE positions, allowing history to be reused without another model evaluation. The same indexing rule is used in specialist training, TA-OPD, and inference, keeping temporal positions within the task's training range as the generated sequence grows. The treatment of retained anchors is detailed in Appendix A.3.

Asymmetric history construction. The information available to each role also determines how it forms history. WM re-encodes each completed LR prediction at zero noise, producing clean KV features for subsequent prediction. This *clean recommit* makes the history represent the completed world state after multi-step denoising. Refinement has a different balance of information and cost: the LR condition already establishes the scene structure, content, and motion, while re-encoding a completed HR chunk requires an additional, expensive evaluation over its spatial tokens. For one-step refinement, this would add a second HR model call solely to construct history.

We instead retain *noisy-forward KV* from the refinement evaluation itself. These features combine the noisy HR input, the LR condition, and preceding temporal context, providing a history for detail synthesis without HR clean recommit. Crucially, the refinement specialist learns under this history during DMD: each new chunk is generated using features retained from earlier noisy forwards. TA-OPD subsequently preserves the same history construction when combining the two specialists. This asymmetry uses the LR scene condition together with temporal features for detail synthesis, requiring only one HR evaluation per chunk. The quality–cost trade-off is analyzed in Section 4.3, with configuration details in Appendix C.3.

Task-asymmetric on-policy distillation (TA-OPD). We initialize the shared student from the WM specialist and regress its velocities to both frozen experts under their task-specific sampling and history policies. Following on-policy distillation (Li et al., 2026; Zhou et al., 2026), supervision is constructed along the student's evolving rollout.

For WM, the teacher supervises all four student-visited denoising queries. After each chunk, both networks separately encode the same completed student LR prediction at zero noise to build clean KV. For Ref, both networks process the same conditional HR noise query using their own histories and retain their noisy-forward KV, without HR clean recommit. The teacher also processes unsupervised history chunks to construct its recurrent context.

Within each task, teacher and student match the noisy latent, timestep, conditions, context window, and RoPE rule, while keeping separate KV states. Both tasks use the velocity-matching loss below.

Local velocity matching. For chunk k and query j, let $z_{k,j}^{\theta,m}$ be the student's current noisy latent and t_j^m its query timestep. For $\psi \in \{\theta, \phi_m\}$, write

$$q_{\psi,k,j}^m = v_\psi\big(\mathrm{sg}[z_{k,j}^{\theta,m}], t_j^m; c_{m,k}, \mathrm{sg}[M_{\psi,k,j}^m]\big). \tag{6}$$

Here $c_{m,k}$ collects the conditions for the current chunk, $M_{\psi,k,j}^m$ is network ψ's state before the query, and sg stops gradients through earlier sampling and history construction.

Let d_θ^m denote the joint distribution of queries and network states constructed above, including sampled conditions and noise, and let τ collect a realization of this process. We optimize

$$\begin{aligned} \mathcal{L}_{\mathrm{con}} &= \sum_m p_m \, \mathbb{E}_{\tau\sim d_\theta^m}[\mathcal{L}_m(\tau)], \\ \mathcal{L}_m(\tau) &= \frac{1}{|\mathcal{B}_m|} \sum_{k\in\mathcal{B}_m} \sum_{j=1}^{J_m} w_j^m \, \ell_{k,j}^m, \\ \ell_{k,j}^m &= \mathrm{MSE}\big(q_{\theta,k,j}^m, \mathrm{sg}[q_{\phi_m,k,j}^m]\big). \end{aligned} \tag{7}$$

Here $\mathcal{B}_m$ contains supervised blocks, $J_{\mathrm{WM}} = 4$, $J_{\mathrm{Ref}} = 1$, $w_j^m = 1/J_m$, and p_m is the task-update frequency. MSE is averaged over latent elements, normalizing for spatial resolution. Each term differentiates only the current student prediction, with the rollout and cached history detached.

Updates alternate in a deterministic WM:Ref ratio of 2:1. TA-OPD uses this regression loss without additional reconstruction or perceptual-reward terms. Its refinement conditions are derived from degraded real videos; evaluation additionally tests refinement of generated LR inputs and the complete self-cascade. Rollout lengths, supervised-block selection, and specialist training details are given in Appendix A.

3.4 Streaming Inference and Context Cost

Inference composes the learned prediction and refinement behaviors on one shared backbone. WM completes an LR chunk and updates its clean history; the upsampled latent then conditions one-step refinement, whose noisy-forward KV is retained for subsequent chunks. Both roles advance their temporal windows with the RoPE reindexing used during training, and the resulting HR chunk is decoded for output. This self-cascade produces HR output incrementally, with each refinement depending only on the LR prefix already available. With fixed parameters, independent sampling streams, and unchanged WM execution settings, enabling refinement leaves the native LR sample path unchanged. Frame layout and startup are detailed in Appendix A.

The factorization allocates a longer prediction window at low resolution and a shorter refinement window at high resolution. Let n_L, n_H be the spatial tokens per latent frame and w, s the respective window capacities. Compared with direct-HR prediction using the same backbone, cache precision, and prediction-window capacity w, the ratio of self-attention KV payloads is

$$\frac{M_{\mathrm{KV}}^{\mathrm{cascade}}}{M_{\mathrm{KV}}^{\text{direct-HR}}} = \frac{n_L w + n_H s}{n_H w} = \frac{n_L}{n_H} + \frac{s}{w}. \tag{8}$$

With s fixed, each added prediction-window slot costs only n_L/n_H as much KV storage as a direct-HR slot. At our reference resolutions, $n_H = 4n_L$; with $w = 25$ and $s = 8$, the payload ratio is 0.57. This analytical comparison concerns KV storage; allocation and peak-memory accounting are described in Appendix D.2.

Parameter sharing additionally removes the duplicate backbone storage of a two-specialist cascade. Both stages still contribute to inference compute: the reference path uses four LR denoising evaluations, an LR clean-history forward, and one HR refinement forward per generated content chunk. Relative to clean-recommit refinement, noisy-forward history removes one HR model evaluation per content chunk. End-to-end resource measurements therefore include history construction, latent transfer, and VAE decoding alongside denoising.

4 Experiments

4.1 Experimental Setup

We evaluate three aspects of shared prediction and refinement: complete HR generation, native prediction retention, and refinement of fixed generated inputs. The backbone is Wan2.2 TI2V-5B, with 480×832 prediction and 960×1664 refinement. W, R, and U denote the WM specialist, refinement specialist, and unified model. WM training uses camera-annotated navigation video; refinement additionally uses natural HR video with synthetic spatial degradation.

We use 158 navigation cases from WBench (Ying et al., 2026), reporting its Average and five dimensions: Video Quality, Setting Adherence, Interaction Adherence, Consistency, and Physical. Methods receive the same observations, camera commands, and temporal intervals while retaining their supported generation settings. Native-WM baselines are HY-World 1.5, LingBot-World 2.0 (fast), and Matrix-Game 3.5; HR baselines combine WorldPlay, LingBot-World 2.0, and Matrix-Game 3.5 with Ultra Flash or FlashVSR (Sun et al., 2025; Gao et al., 2026; Qian et al., 2026; Luxury et al., 2026; Zhuang et al., 2025).

Fixed-input refinement uses identical decoded LR videos, re-encoded for our model. We report MUSIQ (Ke et al., 2021), CLIP-IQA+ (Wang et al., 2023), LR-consistency PSNR, and motion-compensated residuals (MC L1 and its high-frequency variant, MC HF L1). These diagnostics measure perceptual quality, content preservation, and temporal variation; generated futures have no unique HR target. Bold marks the best measured value for each metric within its evaluation protocol, including ties. Evaluation details are in Appendix B.

4.2 Generation Quality and Capability Retention

Shared versus separate generation. Table 2 compares $W \to R$ with $U \to U$. The noisy-forward unified model scores 81.06, close to the separate cascade's 80.96, while improving Interaction and Consistency by 0.28 and 0.18 points. In the measured resident deployments, sharing reduces peak

GPU memory from 45.42 to 35.84 GiB (21.1%; Appendix D). These results demonstrate comparable overall HR quality with a smaller resident model footprint.

Table 2: Complete HR generation: separate specialists versus a shared model. All rows use 480×832 prediction and 960×1664 refinement. Bold marks the highest score per column.

System	Avg.↑	Qual.↑	Set.↑	Inter.↑	Cons.↑	Phys.↑
Cascade ($W \rightarrow R$)	80.96	**84.02**	77.15	86.80	86.54	**71.28**
$U \rightarrow U$ (clean-recommit KV)	80.99	83.83	76.81	86.98	**86.81**	70.53
$U \rightarrow U$ (noisy-forward KV)	**81.06**	83.43	**77.73**	**87.08**	86.72	70.32

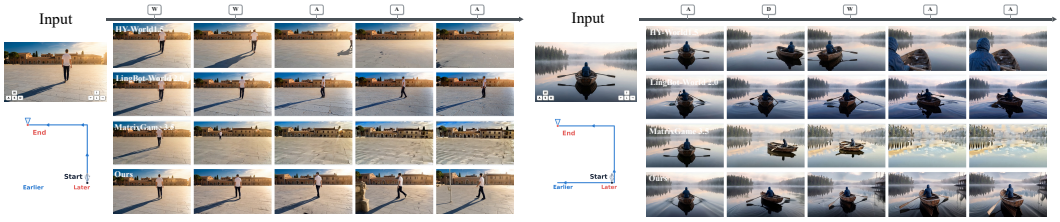

Figure 3: **Native world prediction.** Initial observations, camera trajectories, and generated frames from HY-World 1.5, LingBot-World 2.0, Matrix-Game 3.5, and our WM role.

Native prediction retention. The unified model retains a native WBench Average of 80.308, close to its WM specialist's 80.522 (Table 3). It also achieves the highest Quality score among the evaluated native WMs (83.04). This supports retaining the overall prediction capability while incorporating refinement into the same backbone. Figure 3 illustrates its camera-controlled sequences alongside the baselines.

Table 3: Native WM comparison on the 158-case WBench navigation split, without refinement.

World model	Avg.↑	Qual.↑	Set.↑	Inter.↑	Cons.↑	Phys.↑
Matrix-Game 3.5	74.760	78.46	72.10	83.12	79.63	60.51
HY-World 1.5	79.150	79.82	72.79	**87.38**	**87.52**	68.25
LingBot-World 2.0 (fast)	**81.460**	82.23	**80.50**	84.92	84.45	**75.22**
WM-only specialist W	80.522	83.02	77.14	85.91	86.52	70.02
Unified WM U	80.308	**83.04**	77.28	86.05	85.64	69.53

Complete HR systems. Our system scores 81.056, close to the highest Average of 81.070, and leads in Quality (83.43) and Interaction (87.08) (Table 4). The results demonstrate competitive HR world generation with strong visual quality and control adherence. Since upstream WMs differ, refiner-specific effects are assessed separately on fixed inputs below.

Table 4: WBench scores on final HR videos. Each external WM supplies the same frozen predictions to its two refiners. Bold marks the highest score per column.

World model	Refiner	Avg.↑	Qual.↑	Set.↑	Inter.↑	Cons.↑	Phys.↑
WorldPlay	Ultra Flash	78.630	79.69	72.27	85.62	**87.43**	68.13
WorldPlay	FlashVSR	79.160	80.08	73.09	86.52	87.09	69.01
LingBot-World 2.0	Ultra Flash	**81.070**	81.94	79.29	84.73	84.36	**75.03**
LingBot-World 2.0	FlashVSR	81.030	81.98	**79.86**	84.93	84.44	73.96
Matrix-Game 3.5	Ultra Flash	73.990	78.56	70.81	83.43	79.48	57.69
Matrix-Game 3.5	FlashVSR	75.060	78.72	72.53	83.14	79.51	61.41
SCALEWORLD WM (ours)	Shared Ref (noisy-forward KV)	81.056	**83.43**	77.73	**87.08**	86.72	70.32

Refinement of fixed predictions. On the full-frame protocol, our noisy-forward refiner improves MUSIQ over Ultra Flash by 5.23 and LR-consistency PSNR by 3.96 dB, while reducing MC L1 from 0.0330 to 0.0212 (35.8%; Table 5). Thus, its perceptual gain accompanies better LR content preservation and lower temporal residuals relative to that baseline. Compared with bicubic, however, detail synthesis raises temporal residuals; the method does not dominate simple upsampling on every metric. FlashVSR uses a separate 896×1664 crop protocol and is compared only with its matching bicubic baseline. Figure 4 illustrates the texture and structure differences. Qualitative inspection of the generated videos also reveals inter-frame flickering in Ultra Flash outputs.

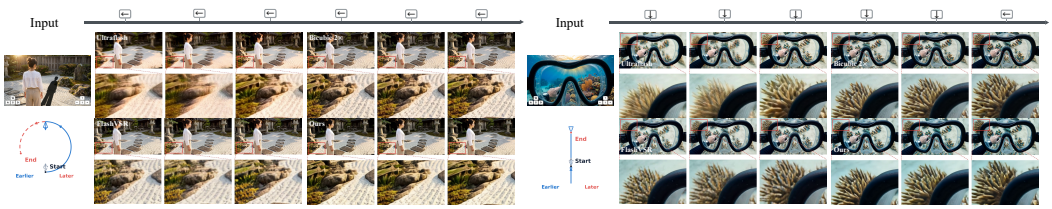

Figure 4: **High-resolution refinement.** Ultra Flash, bicubic 2×, FlashVSR, and our model on two examples; enlarged regions expose differences in structure and texture.

Table 5: Refinement of fixed generated LR videos (158 cases). PSNR measures LR consistency, not HR reconstruction. MC residuals use grayscale in $[0, 1]$. Only rows within each spatial protocol are directly comparable.

Refiner	MUSIQ↑	CLIP-IQA+↑	LR PSNR↑	MC L1↓	MC HF L1↓
FlashVSR center-crop protocol: 896×1664 output					
Bicubic (center crop)	46.3522	0.5037	**44.0321**	**0.0154**	**0.0053**
FlashVSR Full	**61.9861**	**0.5261**	23.4840	0.0327	0.0140
Full frame: 480×832 → 960×1664					
Bicubic	46.8674	0.5175	**45.2736**	**0.0160**	**0.0055**
Ultra Flash v1.1	53.7366	0.4417	22.2844	0.0330	0.0090
Ours (clean-recommit KV)	56.7657	0.5156	25.3340	0.0208	0.0081
Ours (noisy-forward KV)	**58.9656**	**0.5185**	26.2409	0.0212	0.0086

4.3 Training and Efficiency Analysis

Training recipes. TA-OPD improves native WBench Average by 8.308 points over Mixed DMD and 4.430 over CD + mixed DMD (Table 6). Against the latter, Setting improves by 12.75 points and Quality by 7.32. These gains support the specialist-and-consolidation recipe for integrating the two capabilities. The comparison evaluates complete training recipes; the following diagnostics examine task interference.

Table 6: Training-recipe comparison on native WM outputs. These complete recipes differ in preparation and do not isolate on-policy supervision.

Recipe	Avg.↑	Qual.↑	Set.↑	Inter.↑	Cons.↑	Phys.↑
Mixed DMD	72.000	79.82	63.94	76.79	79.96	59.49
CD + mixed DMD	75.878	75.72	64.53	**87.98**	82.44	68.72
TA-OPD	**80.308**	**83.04**	**77.28**	86.05	**85.64**	**69.53**

Task interference in mixed distillation. At fixed model parameters, we separately probe WM and SR DMD gradients. Their global cosine changes from $+0.0969$ at step 400 to -0.3421 at step 800, with negative alignment in all 30 blocks (Table 7). This provides direct evidence of conflicting task gradients and motivates separate specialist optimization. Detailed settings and output diagnostics remain in Appendix C.1.

Table 7: WM/SR DMD gradient alignment during mixed training. Negative blocks have negative gradient cosine; probe settings are in Appendix C.1.

Training step	Global cosine	Negative blocks
400	$+0.0969$	2/30
600	-0.1748	29/30
700	-0.1775	25/30
800	-0.3421	30/30

Noisy-forward refinement history. Table 8 compares clean recommit with noisy-forward history. Retaining the denoising KV removes one HR history forward: refinement takes 300 rather than 600 ms per 16-frame content chunk. Including 192 ms of decoding, latency decreases from 792 to

492 ms (37.9%), and throughput increases from 20.20 to 32.52 FPS. The noisy-forward configuration also increases fixed-input MUSIQ from 56.77 to 58.97 and LR-consistency PSNR from 25.33 to 26.24 dB, with MC L1 changing from 0.0208 to 0.0212. The higher LR consistency indicates that the perceptual gain accompanies better preservation of the supplied scene content. These measurements support one-forward refinement as an effective quality–cost choice. The configurations differ in training and checkpoint age, so their quality difference cannot be attributed solely to KV construction. Details are in Appendix C.3.

Table 8: Refinement-history configurations on fixed LR inputs. HR forwards include history construction; latency and FPS cover refinement plus decoding of 16 frames. PSNR measures LR consistency; HR Q is MUSIQ and TE is MC L1. Training settings and checkpoint ages differ.

Refinement KV	HR forwards	HR Q↑	PSNR↑	TE↓	ms/chunk↓	FPS↑
Clean-recommit	2	56.7657	25.3340	**0.0208**	792	20.20
Noisy-forward (reference)	1	**58.9656**	**26.2409**	0.0212	**492**	**32.52**

Shared-backbone deployment. All deployments use four GPUs, BF16, batch size one, four-step 480×832 prediction, and one-step 960×1664 refinement. Steady-state FPS is computed from the median end-to-end latency over eight 16-frame chunks after six warm-up chunks, including four-GPU halo-8 MG decoding. First-chunk latency uses warm kernels and empty streaming caches. Peak memory is the maximum allocated across GPUs with the backbone, VAE, text encoder, and caches resident.

Table 9 separates weight-sharing savings from changes in model quality using one-copy and two-copy deployments of the same unified checkpoint. Sharing halves resident DiT weight storage from 19.16 to 9.58 GiB per GPU and reduces peak memory from 45.42 to 35.84 GiB, while preserving 11.94 FPS throughput. KV allocation remains 1.77 GiB because both temporal states are still required. First-chunk latency is similar (1.33 versus 1.37 s). Thus, sharing reduces the resident footprint without adding a second model's weight storage or sacrificing steady-state throughput. The matched two-copy control directly supports the deployment advantage of unifying prediction and refinement.

Table 9: Measured deployment costs on four GPUs (BF16, batch size one). Weight/KV are per-GPU allocations in GiB; Peak is the maximum allocated memory across GPUs. First is first-chunk latency; FPS includes decoding. Memory accounting: Appendix D.

Deployment	Copies	Params (B)	Weight GiB	KV GiB	Peak GiB↓	First (s)↓	FPS↑
$W \to R$, resident	2	10.28	19.16	1.77	45.42	1.38	10.77
$U \to U$, two copies	2	10.28	19.16	1.77	45.42	**1.33**	**11.94**
$U \to U$, shared copy	1	5.14	9.58	1.77	**35.84**	1.37	**11.94**

The memory reduction is attributable to removing a duplicate backbone: both unified deployments preserve the same task-specific histories and sampling schedules. It is distinct from the analytical 43% KV-payload reduction relative to direct-HR prediction in Eq. 8. Together, these comparisons quantify two complementary benefits: sharing model weights and retaining the longer prediction history at low resolution.

5 Conclusion

We presented ScaleWorld, a shared causal model for camera-controlled prediction and high-resolution refinement. Its asymmetric temporal contexts allocate long history to low-resolution prediction, while specialist training and TA-OPD integrate the two capabilities. Experiments show comparable overall generation quality to a separate specialist cascade, retained native prediction capability, and lower resident deployment memory. The resulting model produces HR worlds incrementally with one backbone. Limitations are discussed in Appendix F.

AI USE STATEMENT

Large language models (LLMs) assisted with translation, language editing, manuscript organization, and the presentation and consistency checking of author-provided experimental results. The authors are responsible for verifying all AI-assisted text, analyses, and claims and for the final content of the manuscript.

ETHICS STATEMENT

This work follows the ICLR Code of Ethics. The experiments reported in this submission do not involve human subjects or animal experimentation. As with other generative video systems, the proposed model may inherit biases from its training data, produce misleading visual content, or be used to create deceptive media. High-resolution refinement can increase apparent realism without guaranteeing physical correctness or factual accuracy. These limitations should be considered when interpreting generated content or deploying the method, together with appropriate provenance disclosure and safeguards for application-specific risks.

REPRODUCIBILITY STATEMENT

We describe the model architecture, training objectives, and role-specific prediction and refinement procedures in Sections 3 and 4. Implementation details, tensor shapes, sampling schedules, temporal-state management, and the local distillation procedure are provided in Appendix A. The experimental section and appendix specify the evaluation cases, preprocessing, metrics, frame sampling, and paired comparison protocol used for the reported results. Together, these descriptions are intended to make the training, inference, and evaluation procedures reproducible.

REFERENCES

Keyan Ding, Kede Ma, Shiqi Wang, and Eero P. Simoncelli. Image quality assessment: Unifying structure and texture similarity. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(5):2567–2581, 2022.

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

Lanqing Guo, Yingqing He, Haoxin Chen, Menghan Xia, Xiaodong Cun, Yufei Wang, Siyu Huang, Yong Zhang, Xintao Wang, Qifeng Chen, Ying Shan, and Bihan Wen. Make a cheap scaling: A self-cascade diffusion model for higher-resolution adaptation. *arXiv preprint arXiv:2402.10491*, 2024.

Jingwen He, Tianfan Xue, Dongyang Liu, Xinqi Lin, Peng Gao, Dahua Lin, Yu Qiao, Wanli Ouyang, and Ziwei Liu. VEnhancer: Generative space-time enhancement for video generation. *arXiv preprint arXiv:2407.07667*, 2024.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In *Adv. Neural Inform. Process. Syst.*, 2022.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *arXiv preprint arXiv:2506.08009*, 2025.

Junhwa Hur, Charles Herrmann, Saurabh Saxena, Janne Kontkanen, Wei-Sheng Lai, Yichang Shih, Michael Rubinstein, David J. Fleet, and Deqing Sun. High-resolution frame interpolation with patch-based cascaded diffusion. In *AAAI*, 2025.

Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. *arXiv preprint arXiv:2410.05954*, 2024.

Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. MUSIQ: Multi-scale image quality transformer. In *Int. Conf. Comput. Vis.*, pp. 5148–5157, 2021.

Denis Korzhenkov, Adil Karjauv, Animesh Karnewar, Mohsen Ghafoorian, and Amirhossein Habibian. PyramidalWan: On making pretrained video model pyramidal for efficient inference. *arXiv preprint arXiv:2601.04792*, 2026.

Quanhao Li, Junqiu Yu, Kaixun Jiang, Yujie Wei, Zhen Xing, Pandeng Li, Ruihang Chu, Shiwei Zhang, Yu Liu, and Zuxuan Wu. DiffusionOPD: A unified perspective of on-policy distillation in diffusion models. *arXiv preprint arXiv:2605.15055*, 2026.

Hongyu Liu, Chun Wang, Feng Gao, Xuanhua He, Yue Ma, Ziyu Wan, Yong Zhang, Xiaoming Wei, and Qifeng Chen. OPSD-V: On-policy self-distillation for post-training few-step autoregressive video generators. *arXiv preprint arXiv:2607.08766*, 2026. URL https://arxiv.org/abs/2607.08766.

Luxury, Jie Huang, Zihao Fan, Xiaoxiao Ma, Yuming Li, Jun-hao Zhuang, Zeyue Xue, Siming Fu, Haoran Li, Mingchen Zhong, et al. Ultra flash: Scaling real-time streaming video generation to high resolutions. *arXiv preprint arXiv:2606.09150*, 2026.

Meituan LongCat Team. LongCat-Video technical report. *arXiv preprint arXiv:2510.22200*, 2025.

Runjia Qian, Zile Wang, Jihai Zhang, Kai Zou, Wei Yu, Jiaxing Li, Zexiang Liu, Yaokun Li, Fei Kang, Kaichen Huang, Mengyin An, Haobo Zhang, Biao Jiang, Jiahua Wang, Haofeng Sun, Yang Liu, and Yangguang Li. Matrix-Game 3.5: Enhancing real-time streaming interactive world models with patch memory. *arXiv preprint arXiv:2608.29910*, 2026.

Haonan Qiu, Shikun Liu, Zijian Zhou, Zhaochong An, Weiming Ren, Zhiheng Liu, Jonas Schult, Sen He, Shoufa Chen, Yuren Cong, Tao Xiang, Ziwei Liu, and Juan-Manuel Perez-Rua. HiStream: Efficient high-resolution video generation via redundancy-eliminated streaming. *arXiv preprint arXiv:2512.21338*, 2025.

Wenqiang Sun, Haiyu Zhang, Haoyuan Wang, Junta Wu, Zehan Wang, Zhenwei Wang, Yunhong Wang, Jun Zhang, Tengfei Wang, and Chunchao Guo. Worldplay: Towards long-term geometric consistency for real-time interactive world modeling. *arXiv preprint*, 2025.

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

Jianyi Wang, Kelvin C. K. Chan, and Chen Change Loy. Exploring CLIP for assessing the look and feel of images. In *AAAI*, 2023.

Jianyi Wang, Zhijie Lin, Meng Wei, Yang Zhao, Ceyuan Yang, Fei Xiao, Chen Change Loy, and Lu Jiang. SeedVR: Seeding infinity in diffusion transformer towards generic video restoration. *arXiv preprint arXiv:2501.01320*, 2025.

Jianyi Wang, Shanchuan Lin, Zhijie Lin, Yuxi Ren, Meng Wei, Zongsheng Yue, Shangchen Zhou, Hao Chen, Xuefeng Xiao, Chen Change Loy, Lu Jiang, Yang Zhao, and Ceyuan Yang. SeedVR2: One-step video restoration via diffusion adversarial post-training. *arXiv preprint arXiv:2506.05301*, 2026.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T. Freeman, Frédo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2025.

Kaining Ying, Hengrui Hu, Siyu Ren, Jiamu Li, Fengjiao Chen, Ziwen Wang, Xuezhi Cao, Xunliang Cai, and Henghui Ding. WBench: A comprehensive multi-turn benchmark for interactive video world model evaluation. *arXiv preprint arXiv:2605.25874*, 2026.

Cheng Zhang, Boying Li, Meng Wei, Yan-Pei Cao, Camilo Gambardella, Dinh Phung, and Jianfei Cai. Unified camera positional encoding for controlled video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 38027–38037, 2026.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 586–595, 2018.

Wei Zhou, Xiongwei Zhu, Zelin Xu, Bo Dong, Lixue Gong, Yongyuan Liang, Meng Chu, Leigang Qu, Lingdong Kong, Wei Liu, and Tat-Seng Chua. DanceOPD: On-policy generative field distillation. *arXiv preprint arXiv:2606.27377*, 2026.

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. *arXiv preprint arXiv:2605.15178*, 2026a.

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. In *International Conference on Machine Learning*, 2026b.

Junhao Zhuang, Shi Guo, Xin Cai, Xiaohui Li, Yihao Liu, Chun Yuan, and Tianfan Xue. FlashVSR: Towards real-time diffusion-based streaming video super-resolution. *arXiv preprint arXiv:2510.12747*, 2025.

A Implementation Details

A.1 Shared Visual Interface

The reference backbone is Wan2.2 TI2V-5B. Its VAE uses 48 latent channels with spatial stride 16, and the transformer uses 2×2 spatial patches. For SR, a 48-channel interpolated LR latent is concatenated with the noisy target before patch embedding. The added projection weights are initialized to zero when expanding a pretrained input layer. WM uses zero-valued LR channels and its image/camera conditioning; SR uses a zero image-anchor mask and no camera condition. The shared transformer and output head serve both roles.

Table 10: Reference spatial representations.

Role	Pixels	VAE latent	Patch grid
WM	480×832	30×52	15×26
SR	960×1664	60×104	30×52

Pixel-space degradation uses blur, bicubic downsampling, and noise while retaining consecutive frames. Latent conditioning uses nearest-neighbor spatial interpolation in the reference recipe. These operations are distinct: bicubic describes the LR video construction, not the latent upsampler. Additional latent-condition noise/dropout and a learned latent upsampler are disabled during reference consolidation. No SR-specific spatial Swin partition is enabled.

A.2 Sampling and Temporal State

Table 11 describes the raw-SR consolidation variant used as the reference for the Method. Here SR denotes the task used to learn the Ref role. The shared shift parameter is 5.0; nominal timestep indices are warped by the common scheduler. Teacher and student use matching flow/noise parameterizations. The four-step WM transitions re-noise clean student predictions, whereas one-step SR has no additional within-chunk query.

Table 11: Reference consolidation settings. Sampling and cache settings are listed separately for each role.

Property	WM	SR
Condition	image, camera, text	LR latent, text
Training layout	$[1, 4, 4, \ldots]$	$[4, 4, \ldots]$
Nominal indices	$[1000, 750, 500, 250]$	$[1000]$
Temporal span	25 latent frames	8 latent frames
Rollout	25 latent frames	8–16 latent frames
Supervised blocks	6 content blocks	final block
Supervised queries	all 4	sole query
History construction	clean recommit	raw noisy-forward KV
Temporal RoPE	window reindexing	window reindexing

For clean WM commitment, both networks separately process the same completed student chunk at context timestep zero. For raw SR, each network retains its own noisy-forward entries; the specialist is also run on unsupervised history blocks. No student KV tensors are copied into a specialist. All preceding states and transitions are detached, so the local velocity loss does not backpropagate through earlier chunks. All six WM content blocks receive supervision at all four queries. For SR, a two-to-four-block rollout provides raw history, and only the final block contributes the regression loss.

ADDITIONAL QUALITATIVE RESULTS

Figure 5 shows additional camera-controlled rollouts and their corresponding high-resolution refinements.

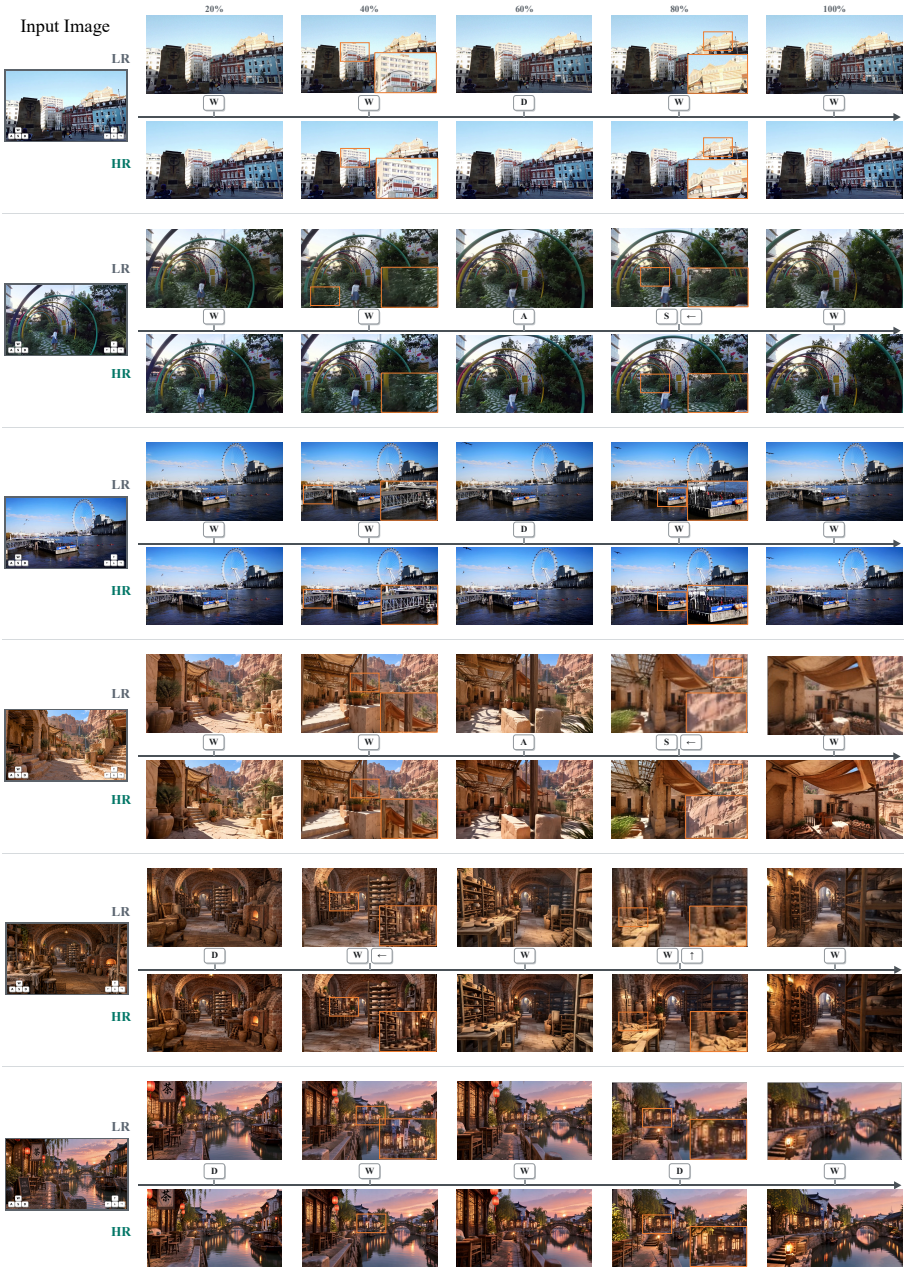

Figure 5: **Additional qualitative self-cascade results.** For each camera-controlled trajectory, the leftmost image is the initial observation, and the remaining columns show frames at 20%, 40%, 60%, 80%, and 100% of the rollout. The LOW rows are obtained by directly decoding the WM task's predicted latents with the VAE for visualization. The corresponding HIGH rows are produced by the SR task conditioned on those WM latents. Labels below the LOW rows indicate camera controls; orange boxes and insets highlight matched regions.

A.3 Temporal RoPE Reindexing

Both WM and Ref use local temporal positions within their respective attention windows. Reindexing addresses positional extrapolation when inference exceeds the finite temporal spans used for training. Let p_i and $\widetilde{p}_i$ be the old and new temporal positions of a retained token, and let $R(p)$ denote the temporal RoPE rotation. If K_i already includes the rotation at p_i, its updated key is

$$\widetilde{K}_i = R(\widetilde{p}_i)R(p_i)^{-1}K_i. \tag{9}$$

The corresponding value is retained unchanged. This updates positional encoding without recomputing the hidden features of the retained history. Spatial RoPE coordinates are unaffected.

For Ref with an eight-latent window and four-latent chunks, the most recent history chunk occupies positions 0–3 and the incoming chunk occupies 4–7 once the window is full. On each subsequent shift, the retained keys are reindexed to 0–3 before the new chunk is processed. For WM, a retained image anchor occupies the initial position, and the remaining window contains the most recent content history and current chunk. The anchor keeps its designated position while the moving history is reindexed. The indexing convention is shared between specialist rollouts, student and teacher histories during TA-OPD, and streaming inference. Reindexing aligns the range of temporal positions; it does not make long generated histories identical to training data.

A.4 Frame Alignment and Startup

For the causal Wan VAE, F latent frames correspond to $4(F-1)+1$ pixel frames. The first latent frame corresponds to the initial observation, while subsequent four-latent content blocks each produce 16 pixel frames. WM starts from a one-latent observation anchor. Refinement processes this anchor before the subsequent content blocks; standalone SR training uses uniform four-latent blocks. Noisy-forward refinement retains the KV entries from its generation pass, whereas clean-recommit refinement adds an HR history forward.

A.5 Optimization and Specialist Training

The causal specialists are trained with DMD after shared-model preparation. In that stage, the real and learned fake score models provide distribution-matching guidance to a causal generator; the resulting generator becomes a TA-OPD teacher. The reference SR specialist learns with noisy-forward history and window-local RoPE positions during its DMD rollouts. Its training includes an auxiliary reconstruction objective, whereas the subsequent TA-OPD stage uses only local velocity regression. Specialist preparation and TA-OPD therefore have distinct objectives while preserving the corresponding generation context policies.

The reference optimizer follows a deterministic WM, WM, SR update cycle with learning rate 2×10^{-6}, effective batch size 16, gradient clipping 0.5, and EMA decay 0.99. Consolidation uses local flow MSE without an extra reconstruction or reward term. Both specialists inherit the jointly trained backbone before task-specific DMD training.

A.6 Function-Preserving Input Expansion

We establish the functional equivalence of the original WM and its zero-conditioned input expansion with the original parameters fixed. Let x contain all original patch-embedding input channels and let r denote the appended LR condition. For a flattened input patch, the original affine projection is $P_{W,b}(x) = Wx + b$; the expanded projection is

$$\widetilde{P}_{W,A,b}(x, r) = Wx + Ar + b, \tag{10}$$

where A contains only the added condition-channel weights. This notation also represents the implemented 3D convolution, whose kernel is linear in the input channels. The expansion copies W and b and initializes A to zero.

Proposition (zero-condition equivalence). Suppose the added condition enters only through Eq. 10, with no joint normalization over the original and added channels before projection. Keep all original weights, conditioning paths, token geometry, and downstream computations unchanged.

For identical timesteps, WM conditions, and causal states, setting $r = 0$ makes the expanded model's prediction equal to that of the original model, for any finite A, in exact arithmetic. With $A = 0$, the same equality holds for any finite r, provided all other inputs and execution settings remain unchanged.

Proof. By linearity,

$$\widetilde{P}_{W,A,b}(x, 0) = Wx + A0 + b = P_{W,b}(x). \tag{11}$$

Thus both networks receive identical patch embeddings. Since all remaining operations, parameters, and conditions agree, their predictions and corresponding cache updates agree. Starting from equal causal states, induction over denoising queries and clean-history commitments extends the equality to the WM rollout when the sampler, random draws, and state-management policies are identical. The $A = 0$ case follows from $0r = 0$. □

Gradient implication. For a WM loss whose dependence on A is only through this projection, zero conditioning gives

$$\nabla_A \mathcal{L}_{\mathrm{WM}} = \sum_p \frac{\partial \mathcal{L}_{\mathrm{WM}}}{\partial h_p} r_p^\top = 0, \tag{12}$$

where p indexes projected patches and h_p is their embedding. This is a statement about the loss gradient, not the optimizer update: stored momentum or weight decay can still change A.

Numerical precision. The equalities above hold in exact arithmetic. Floating-point implementations may differ across kernels; numerical validation should use matched executions, a declared tolerance, and isolated WM random streams.

B Evaluation Details

B.1 World-Generation Evaluation

WBench evaluator. All WBench results reported in this paper are evaluated with Doubao as the multimodal judge. We use the same evaluator, prompt templates, output parsing, dimension definitions, and aggregation procedure for every method. The reported Average is the arithmetic mean of the five unrounded WBench dimension scores.

Matched prediction paths. The native WM and self-cascade should use identical WM random draws, conditions, and mutable runtime settings. Separate per-role random streams prevent SR sampling from changing later WM noise. Any discrepancy in native LR outputs must first be checked for cache or runtime-state contamination. This execution check is distinct from comparing the trained unified model against the WM specialist.

B.2 Refinement Inputs

Controlled refinement interface. For fixed-input comparisons, the refinement role can receive an encoded external LR video through the same condition channels. When comparing pixel-input refiners, all methods receive the same decoded LR video, which our model re-encodes for conditioning. Native latent handoff is evaluated separately as part of the complete self-cascade.

Generated and paired inputs. For generated LR, we evaluate refinement without assuming a unique HR target. For held-out real-video pairs, PSNR, SSIM, temporal PSNR, and perceptual distances provide reconstruction diagnostics (Zhang et al., 2018; Ding et al., 2022). Bicubic upsampling provides a reference for refinement, alongside content and temporal analysis.

B.3 Measured Fixed-Input Refinement Protocol

The two refinement variants in Table 5 are named by their refinement history construction. Ours (clean-recommit KV) uses EMA weights at step 400 and re-encodes completed HR chunks at context timestep zero. Ours (noisy-forward KV) uses EMA weights at step 2000 and retains the KV entries

from conditional noisy forwards. The runs differ in training configuration and checkpoint age; their comparison does not isolate the causal effect of KV policy. These training durations apply to the fixed-input refinement evaluation.

Table 5 uses all 158 cases with matched case IDs, frame counts, and 24 FPS timestamps. The videos contain 193, 289, 385, 481, or 577 frames; all decoded frames are retained. MUSIQ and CLIP-IQA+ are computed at frames $0, 8, 16, \ldots$ using `pyiqa` 0.1.16 (default MUSIQ weights and the CLIP-IQA+ RN50 learned-prompt model). We average sampled image-quality scores within each video and then weight videos equally.

LR-consistency PSNR uses OpenCV AREA downsampling of floating-point RGB outputs and the mean squared error over all pixels, channels, and frames of one video, with peak value 1. For MC L1, we estimate forward and backward Farnebäck flow from the corresponding LR view, with maximum side length 640. The backward flow warps the preceding HR grayscale frame to the current frame; forward–backward consistency and image bounds define a shared validity mask for all methods within a protocol. The residual excludes a four-pixel HR border. MC HF L1 uses grayscale minus a Gaussian blur with standard deviation one HR pixel. These are LR-guided motion-compensated diagnostics, not official WBench consistency scores.

The full-frame Bicubic baseline uses OpenCV INTER_CUBIC. The FlashVSR baseline follows its PIL Bicubic resize-then-crop implementation; resizing and cropping are performed in memory without re-encoding the output videos. Both of our variants and Ultra Flash retain the full frame. External chunk layouts are not used to infer chunk-boundary measurements.

Interpretation of temporal metrics. Motion-aware temporal residuals are interpreted together with motion magnitude and detail, since static or blurred output can exhibit low temporal error. For WBench, official scores retain their original preprocessing; additional common-resolution and native-crop analyses are identified separately.

B.4 Evaluation Accounting and Computational Costs

Evaluation cases and failures. Multiple turns or sampling seeds from one case do not count as independent scenes. Generation failures and evaluator failures are recorded separately, with fixed exclusion rules. Human comparisons, when conducted, randomize presentation order, allow ties, and report both case and rater counts.

Cost accounting. We separate denoising, cache construction, interface conversion, and VAE decoding. Peak memory includes both role caches and any resident modules or preallocated output buffers. Finite cache windows bound history storage, not necessarily the memory required when retaining a complete output video. Training-cost comparisons include specialist preparation in addition to consolidation, and do not equate optimizer steps across different objectives with equal compute.

C Additional Experimental Analysis

C.1 Gradient and Output Diagnostics of Mixed DMD

We examine historical mixed-DMD runs to characterize task interference. At each diagnostic step, WM and SR generator gradients are computed separately at the same model parameters, without an optimizer update between the two probes. Dot products and squared norms are aggregated across FSDP shards to obtain global and per-block cosine similarities. The native probes use each task's sampling policy; their batches and random draws need not be identical across training steps.

Table 7 follows a run initialized from consistency-distillation step 1200, with WM:SR updates at 3:1, SR high-noise DMD weight 0.5, and wavelet-reconstruction weight 0.25. The reported cosines compare the two DMD gradients only. Negative alignment persists at the sampled steps 600–800 and extends to all 30 transformer blocks at step 800. In a separate two-block SR run with reconstruction weight 0.25 and no high-noise reweighting, the step-200 cosine is -0.1734 between WM DMD and SR reconstruction, and -0.1037 between WM DMD and the total SR objective. The latter SR gradient has approximately 3.29 times the WM gradient norm.

Table 12: Reference student consolidated from two specialist teachers. WM Inter./Cons. are native WBench Interaction/Consistency; HR Q and TE are fixed-input MUSIQ and MC L1 for noisy-forward refinement.

WM teacher	SR teacher	WM Inter.↑	WM Cons.↑	HR Q↑	TE↓
T_{WM}	T_{SR}	86.05	85.64	58.9656	0.0212

Table 13: Separate specialist cascade versus unified generation at 480×832 LR and 960×1664 HR. WBench Avg. is the mean of its five official dimensions. FPS includes complete inference. WBench scores are measured for all three systems. MUSIQ reports the corresponding refiners on the fixed decoded LR inputs of Table 5, rather than native latent-handoff outputs. Bold marks the highest reported value in each column.

System	Avg.↑	Qual.↑	Set.↑	Inter.↑	Cons.↑	Phys.↑	MUSIQ↑	FPS↑
Cascade ($W \to R$)	80.96	**84.02**	77.15	86.80	86.54	**71.28**	**59.2371**	9.41
$U \to U$ (clean-recommit KV)	80.99	83.83	76.81	86.98	**86.81**	70.53	56.7657	9.41
$U \to U$ (noisy-forward KV)	**81.06**	83.43	**77.73**	**87.08**	86.72	70.32	58.9656	**11.94**

Output diagnostics also show competing quality trends. In a WM:SR=1:1 run, mean SR PSNR over 32 validation cases increases from 26.446 to 27.444 dB between steps 0 and 400. Over 32 WM cases with matched sample IDs, frame indices, and noise seeds, the median prediction-to-GT luminance-jump ratio rises from 1.815 to 2.515, and the corresponding color-jump ratio rises from 1.538 to 2.237. These ratios measure adjacent-frame changes in global luminance and color, not comprehensive world consistency. Together, the diagnostics establish observed gradient conflicts and competing output trends in these runs; they do not establish gradient conflict as the sole cause of degradation or show that TA-OPD eliminates it.

C.2 Reference Specialist Configuration

Specialist teachers for consolidation. Table 12 summarizes the current student trained with prediction and refinement specialists. Task-specific SFT supplies score-model supervision for specialist DMD, whose few-step generators then serve as frozen TA-OPD teachers. Each teacher constructs its own temporal state under the corresponding sampling and history policy. The table reports the resulting student's native WM and fixed-input refinement scores.

C.3 Refinement-History Configuration Comparison

Comparison of refinement KV configurations. Table 8 compares clean-recommit KV with the noisy-forward KV used in Figures 1 and 2. Both variants use TA-OPD and keep WM history clean. We reuse the fixed-input quality measurements in Table 5, obtained on identical frozen LR videos. The variants differ in training settings and training duration, so this comparison does not isolate the effect of KV construction. Clean recommit adds one HR forward per content chunk to construct history; noisy-forward KV retains entries from the denoising forward. We report detail, temporal error, decoded refinement latency, and throughput to measure the quality–cost trade-off. Refinement takes 0.600 and 0.300 seconds per 16-frame content chunk for clean-recommit and noisy-forward, respectively. Adding 0.192 seconds for decoding gives 0.792 and 0.492 seconds per chunk; throughput is computed as 16 divided by these totals. These timings exclude WM prediction.

C.4 Additional System Metrics

Table 13 retains fixed-input refinement quality and throughput alongside system-level WBench scores. Measured deployment results are reported separately in Table 9.

D Efficiency and Memory Analysis

D.1 Model Storage and Deployment Configuration

Table 9 compares separate specialists with one or two resident copies of the unified model under matched hardware, precision, output size, and batch size. The two-copy control uses identical U weights and inference settings to the shared-copy run, isolating duplicated weight storage from model quality. Backbone parameters, actual resident weight memory, allocated self-attention KV memory, and full-pipeline peak memory are reported separately. Peak memory also includes the VAE, text encoder, other caches, activations, temporary buffers, and retained outputs; it is not the sum of separately measured peaks.

D.2 KV Storage across Resolutions

Prediction and refinement need not retain equally long temporal windows at the same spatial resolution. Let n_L and n_H denote the spatial tokens per latent frame, and let w and s be the allocated WM and SR window capacities in latent frames, including current-chunk and sink slots. For a dense self-attention cache with L layers, batch size B, KV width d_{KV} per layer, and b bytes per element, one cache bank requires $2LBd_{\mathrm{KV}}b$ bytes per stored token. The factor two counts keys and values. We assume identical backbones and cache precision, without cache sharding, quantization, or token compression.

Writing $\kappa = 2LBd_{\mathrm{KV}}b$, the two-role self-cascade and a direct-HR WM with the same prediction-window capacity have

$$M^{\mathrm{KV}}_{\mathrm{cascade}} = \kappa(n_L w + n_H s), \tag{13}$$

$$M^{\mathrm{KV}}_{\mathrm{direct}} = \kappa n_H w. \tag{14}$$

For $\rho = n_H/n_L$, their ratio is

$$\frac{M^{\mathrm{KV}}_{\mathrm{cascade}}}{M^{\mathrm{KV}}_{\mathrm{direct}}} = \frac{1}{\rho} + \frac{s}{w}. \tag{15}$$

Thus the self-cascade uses less KV storage whenever $s < (1 - 1/\rho)w$. For fixed s, each additional WM window slot costs only $1/\rho$ as much KV storage as a direct-HR slot. These are storage identities; they do not assume equal prediction quality at the two resolutions.

Reference configuration. For 480×832 prediction and 960×1664 refinement, Table 10 gives $n_L = 390$, $n_H = 1560$, and $\rho = 4$. With $w = 25$ and $s = 8$, Eq. 15 is 0.57, corresponding to a 43% reduction in self-attention KV payload. For $L = 30$, $d_{\mathrm{KV}} = 3072$, $B = 1$, and $b = 2$, the analytical sizes are shown below.

Table 14: Analytical self-attention KV payload in GiB (2^{30} bytes), with one bank per role. These values are not measured peak GPU memory.

Cache	GiB
LR prediction ($w = 25$)	3.35
HR refinement ($s = 8$)	4.28
Self-cascade total	7.63
Direct-HR prediction ($w = 25$)	13.39

Allocation and interpretation. The current inference allocator reserves positive and negative cache banks for each role, although the cascade forwards use only the positive bank. Retaining both banks doubles the above payloads: 15.26 GiB for the cascade and 26.78 GiB for a direct-HR path using the same allocator. An optimized single-branch comparison should omit unused banks on both sides. Cross-attention caches, weights, activations, temporary buffers, and VAE state are additional costs. The largest executable window must therefore be measured using complete inference, including window transitions and decoding.

The reduction in Eq. 15 comes from retaining the prediction context at low resolution with a shorter HR refinement window, and also applies to a separate-model cascade with the same cache geometry.

Sharing weights additionally avoids storing a second backbone. Whether the shorter HR window preserves sufficient detail consistency is evaluated through refinement and temporal results.

E Extended Related Work

Controllable world generation. Video diffusion models learn rich appearance and motion priors (Ho et al., 2022; Team Wan et al., 2025). CausVid and Self Forcing combine few-step generation with autoregressive execution (Yin et al., 2025; Huang et al., 2025); Causal Forcing further studies causal diffusion distillation (Zhu et al., 2026b). WorldPlay, LingBot-World 2.0, and Matrix-Game 3.5 extend controlled generation with long-horizon modeling and memory (Sun et al., 2025; Gao et al., 2026; Qian et al., 2026). SANA-WM combines camera-controlled generation with a separate long-video refiner (Zhu et al., 2026a). These developments motivate treating visual quality, control, and temporal persistence together. Our work focuses on adding high-resolution refinement to a shared world-model backbone, rather than introducing a new camera or geometric memory representation.

Shared coarse-to-fine generation. Self-Cascade reuses a low-resolution diffusion prior for higher-resolution adaptation (Guo et al., 2024). Pyramid Flow jointly learns resolution stages within a unified flow, and PyramidalWan adapts pretrained models to pyramidal generation (Jin et al., 2024; Korzhenkov et al., 2026). HiStream applies low-resolution early denoising and high-resolution late denoising with dual-resolution caches (Qiu et al., 2025). Its final high-resolution prediction also refreshes the low-resolution history. In contrast, our prediction role completes its native output before conditional refinement, and the refinement does not feed back into the prediction recurrence.

Other forms of parameter reuse include LongCat-Video's refinement-specific LoRA and HiFI's shared interpolation and spatial upsampling model (Meituan LongCat Team, 2025; Hur et al., 2025). These are important precedents for reducing independently stored parameters. Our study concerns the coexistence of camera-controlled prediction and conditional refinement in the same causal backbone under different sampling budgets; it does not claim that full sharing universally dominates adapter-based alternatives.

Generative video refinement. VEnhancer improves generated video through space-time enhancement, while SeedVR and SeedVR2 study diffusion-based restoration (He et al., 2024; Wang et al., 2025; 2026). FlashVSR targets single-step streaming super-resolution (Zhuang et al., 2025). Ultra Flash combines an autoregressive low-resolution generator with a dedicated causal refiner, including latent upsampling, single-step distillation, and generated-context preference optimization (Luxury et al., 2026). It also exploits different computation requirements across stages. Our distinction is therefore not single-step refinement or asymmetric step counts alone, but their consolidation with native world prediction in one shared model.

Distillation and capability composition. Distribution matching distillation transfers generative behavior to few-step students (Yin et al., 2024), and student-rollout training addresses the exposure bias of autoregressive video (Huang et al., 2025). DiffusionOPD consolidates specialist capabilities on student trajectories; DanceOPD studies capability composition through routed velocity-field matching (Li et al., 2026; Zhou et al., 2026). For autoregressive video, OPSD-V supervises student-visited denoising states using a teacher with cleaner, data-assisted temporal context (Liu et al., 2026). We build on local velocity matching to consolidate prediction and refinement specialists in a jointly trained video model. Our study centers on composing these capabilities within one causal backbone for high-resolution world generation, evaluating native prediction retention, conditional refinement quality, and the resource cost of their combined execution.

F Limitations

Our formulation targets camera-controlled visual prediction, not an explicit physical state simulator. High-resolution refinement may introduce unsupported structures or alter appearance even when it does not feed back into the native prediction recurrence. The SR role is trained on degraded real-video conditions, whereas self-cascading uses generated LR conditions; TA-OPD alone does

not remove this distribution gap. Capability retention and temporal robustness must therefore be interpreted within the evaluated controls, domains, and horizons.

Sharing a backbone reduces duplicated weight storage relative to a two-expert cascade, but retains sequential prediction and refinement computation as well as separate temporal states. Specialist acquisition also contributes to training cost. Our comparisons do not establish universal superiority over adapter-based reuse or smaller dedicated refiners, and bounded KV windows do not by themselves imply bounded total application memory or real-time throughput.