Parameter-efficient adaptation of tri-view robot video predictors

YiCheng Pan | KineJing Research Team

Author review draft | 25 October 2026 | Not submitted

Abstract

Adapting robot video predictors on limited hardware requires separating predictive gains from execution feasibility. We develop KineJing Pro Local, a tri-view workflow that adapts the released BWM model using 1.47 million low-rank parameters and compressed weight storage. In a matched local configuration, 96 updates reduce future RGB mean squared error by 71.06% across ten previously explored simulation trajectories (paired-trajectory bootstrap 95% interval, 59.08-77.61%). The complete RGB pathway executes on a 12 GB laptop GPU. Separately, the earlier KineJing-Pro submission ranks first in perspective and sixth in physical/3D and temporal consistency among 60 TriWorldBench entries. The local adapter has not received an official benchmark score and still trails first-frame persistence. These findings establish a limited, measurable adaptation capability and an executable local system, while identifying action-sensitive prediction on unseen data as the next validation target.

Robot video prediction has two distinct experimental demands: a model must run under an explicit resource budget, and its predicted transitions must improve under a meaningful evaluation. A successful execution alone establishes neither accurate dynamics nor useful robot control. This distinction becomes particularly important for three-camera manipulation, where an apparently stable head view can coexist with incorrect local interactions or disagreement at the wrists [1].

Our research question is whether a small, explicitly trained adaptation can improve a released robot video predictor while retaining an executable RGB pathway on a consumer laptop. We study this question through two evidence streams: the earlier KineJing-Pro benchmark submission, and a subsequent, separately versioned local adaptation experiment. They share upstream model lineage but are different systems. The benchmark result is not assigned to the local adapter, and local pixel improvements are not converted into a predicted benchmark score.

The KineJing contribution is an implemented adaptation and execution workflow. It includes checked conversion of bimanual trajectory inputs, a validation-selected query/value adapter, a loading and compressed-storage path, and a strict initial-image-to-RGB interface. The released BWM model supplies the pretrained action-conditioned core; BWM itself initializes from Wan2.2-TI2V-5B [2]. We do not claim to have trained that core, invented low-rank adaptation, or established a new world-model architecture.

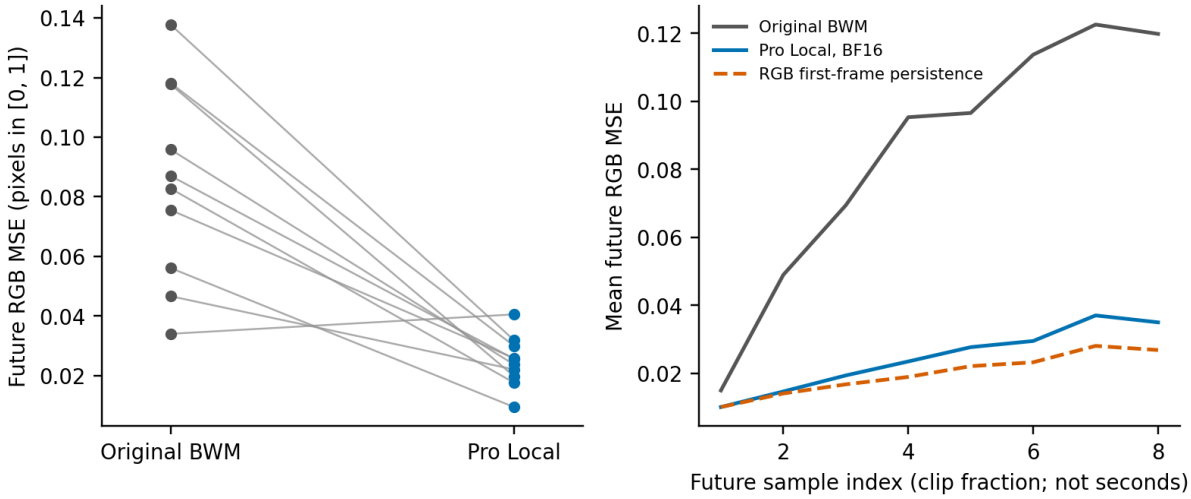
Our evidence supports three specific contributions. First, the local adapter improves decoded future RGB prediction relative to the unchanged BWM checkpoint under matched inputs, noise and integration steps. Second, the complete compressed core and VAE execute together on the tested 12 GB GPU, with clearly scoped memory and timing measurements. Third, a full benchmark profile documents where the earlier submission performs well and where it fails. These contributions are deliberately scoped to measured adaptation and implementation, rather than general robot competence.

Results

A small trained adapter improves the matched local predictor

The adapter contains 1,474,560 trainable parameters, approximately 0.0293% of the 5,025,700,032-parameter upstream core. Rank-four updates are installed on the query and value projections of 30 self-attention blocks. We train for four epochs over 24 complete trajectories, select the checkpoint using four separate validation trajectories, and evaluate ten further trajectories. All three views remain in the same trajectory partition. The public development data had been explored previously; this is a local holdout comparison rather than a new blind benchmark.

Under identical evaluation inputs and noise, future RGB MSE decreases by 71.06%, from 0.08517601 to 0.02464687. Mean episode PSNR increases from 11.050 to 16.355 dB, and SSIM from 0.54946 to 0.63926. The recorded 10,000-resample paired whole-trajectory bootstrap gives a 95% interval of 59.08-77.61% for relative MSE reduction. This interval describes variation over the evaluated trajectories; it does not include training-seed variability, unseen tasks or population-wide robot performance.



Matched local prediction and horizon errors. Left, lines join the original and adapted errors for each of ten complete evaluation trajectories; points are trajectories, not independent frames. Right, mean errors across the eight future sample indices. Inputs contain three 96 x 96 views and supplied future EEF states. Samples are clip fractions, not elapsed seconds. Persistence is included to expose the remaining dynamic-prediction gap. Source: retained PRO-REAL-0001 RGB evaluation.

Compressed storage retains the observed local quality

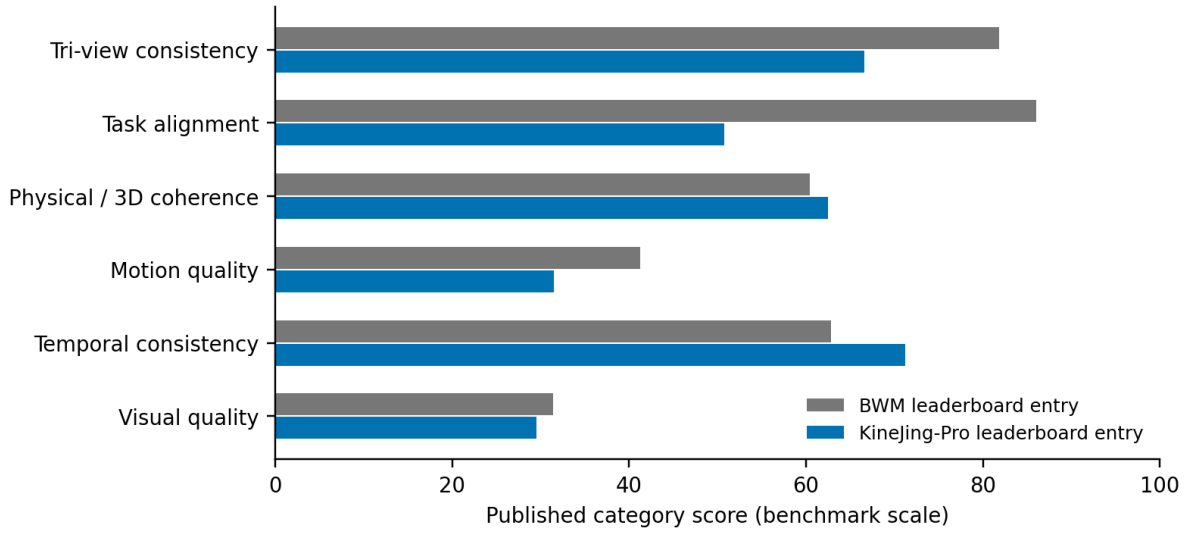
The compressed path stores linear weights in row-scaled FP8 E4M3FN representation and dequantizes them for BF16 matrix multiplication. It is a storage implementation, not native FP8 GEMM. Its mean future RGB MSE is 0.02439503, compared with 0.02464687 for the adapted BF16 model. The recorded paired interval for this small relative difference spans -0.93% to 2.86%; the data do not establish an additional quality gain from quantization. The supported result is that the compressed representation meets the experiment's quality-retention criterion on these inputs.

The complete pipeline accepts CPU initial RGB images and a supplied trajectory, encodes the initial images with the Wan VAE, predicts with the compressed core, decodes future RGB and returns the result to CPU. On the tested RTX 5070 Ti Laptop GPU, peak CUDA allocated memory during prediction is 6.291 GiB after resetting the counter following model loading. The first prediction takes 3.901 s; three subsequent repetitions on one fixed input have a median of 1.501 s. The full process, including loading, quantization and four prediction repetitions, takes 86.640 s. These measurements establish feasibility for the specified small configuration, not real-time, production-resolution or hardware-independent performance.

The earlier benchmark submission has specific geometric strengths

In the 25 October 2026 TriWorldBench snapshot, KineJing-Pro scores 54.96 and is listed 44th among 60 published model entries. Perspective is 91.43, ranked first; physical/3D coherence is 62.50 and temporal consistency 71.22, both ranked sixth. These are strengths of the submitted system in that dated benchmark, rather than evidence of a newly trained foundation model. The official table and its embedded JSON are two representations of one response; agreement between them checks extraction consistency, not independent evaluation replication.

The same snapshot lists BWM at 65.54 overall. KineJing-Pro has higher published perspective, physical/3D and temporal scores, but lower overall, tri-view, task-alignment and motion scores. We report the six-category profile in full rather than selecting only winning metrics. Because checkpoint identity, sampling settings and per-episode scores are not jointly available for these entries, their difference is a descriptive leaderboard comparison. It is not an ablation proving that KineJing post-processing caused the geometric or temporal advantages.



Complete six-category benchmark comparison. Scores are copied from the same dated public leaderboard snapshot. Blue denotes the earlier KineJing-Pro submission and grey the published BWM entry. The local LoRA adapter is not evaluated in this figure. Category values have benchmark-specific meanings; bar lengths do not measure physical error or robot success. No uncertainty intervals are available for these aggregate entries. Source: TriWorldBench, accessed 25 October 2026.

Persistence remains a stronger local pixel baseline

A useful positive comparison must survive an inexpensive reference. Exact repetition of the permitted initial RGB frame yields future MSE 0.02005718, PSNR 17.281 dB and SSIM 0.69815. The BF16 adapter therefore remains 22.88% worse in MSE than RGB persistence. Repeating the initial latent and decoding it also outperforms the predictor, with MSE 0.01720288. Access to future reference latents yields a VAE-reconstruction MSE of 0.00059897; this last result is an oracle reconstruction diagnostic and is never treated as a predictor.

This comparison narrows the finding: the adapter repairs a substantial part of the original predictor's error in the local configuration, but has not established a useful dynamic advantage over a static baseline. Nor does a low pixel error alone prove correct contact, action response or task completion. Related evaluation work separates perceptual dimensions from task and functional utility [3-5]. The next decisive test is consequently an action-sensitive, persistence-beating result on untouched trajectories, rather than another reduction relative only to a weak local starting point.

Discussion

The central result is practical adaptation under a constrained execution budget. Training a small parameter subset produces a large matched-configuration error reduction, and a compressed loading path supports the complete input-to-RGB computation on the tested laptop. These findings support further research on local adaptation; they do not establish that limited hardware alone can reproduce the breadth of a large benchmark or physical-robot study.

Two distinctions govern interpretation. The first is lineage versus contribution: using BWM provides a strong pretrained starting point, while KineJing's adapter, input checks and runtime are separate engineering and experimental additions. The second is benchmark profile versus causal mechanism: favorable perspective and temporal scores describe the earlier submission, whereas the local experiment directly compares trained and unchanged weights. Neither evidence stream can replace the other.

The observed profile motivates, but does not prove, a stability-dynamics trade-off. A video that changes too little may retain appearance and geometric regularity without following the desired manipulation. Alternatively, score differences could reflect input normalization, rollout settings, image resizing, post-processing or evaluator sensitivity. The present artifacts do not distinguish these explanations. A controlled study should fix the upstream checkpoint, trajectory conversion, random draws, resolution and codec, then change one KineJing component at a time.

The limitations are material. Training has one seed; evaluation uses ten previously explored simulated trajectories, square low-resolution images and nine sparse sample positions. The supplied future EEF values are recorded states, not verified intervention commands. Complete-pipeline consistency was checked on one trajectory, and latency on three hot repetitions of one input. The scored competition package predates the local adapter. No new official score, unseen-task generalization, closed-loop benefit, real-robot validation or safety outcome is established here.

Methods

Trajectory and view contracts

The retained TWB input adapter reads permitted initial head, left-wrist and right-wrist images and the trajectory HDF5 data. Each arm contributes Cartesian position, XYZ Euler rotation and gripper opening, giving fourteen columns. Quaternion inputs use the recorded wxyz convention; the adapter checks shape, finiteness, unit quaternion norm and temporal length before conversion. These checks avoid silently treating joint coordinates, quaternions and absolute EEF states as interchangeable. They validate representation, not full coordinate-frame or hardware-command equivalence.

For the local experiment, each trajectory provides nine sampled positions and three 96 x 96 RGB views, stacked along image height before VAE processing. Input normalization uses fixed upstream first and ninety-ninth percentile bounds. Future ground-truth images are targets only; the prediction path uses the initial images and supplied EEF sequence. Later sparse indices have no calibrated mapping to seconds. A separate packaging adapter resizes generated submission videos and permits at most three repeated trailing frames to meet expected counts; this formatting operation is not claimed as a learned improvement or proof that the exact adapter version produced the scored package.

Adapter training and checkpoint selection

For each selected frozen projection W, low-rank adaptation adds a residual delta W = (alpha/r) BA, with rank r = 4 and alpha = 4. We follow the low-rank adaptation construction rather than claiming the decomposition itself as novel [7]. Adapter factors are FP32; their residual is cast for the BF16 core. The base core and action encoder remain frozen. The retained implementation initializes B to zero and A with Kaiming initialization, so the initial residual is zero.

The recorded recipe uses AdamW, learning rate 0.0001, zero weight decay, batch size one and L2 gradient clipping at one. Four passes over 24 training trajectories yield 96 updates. Ninety-five updates record positive finite gradients; one samples the upstream scheduler's zero-loss-weight endpoint. Validation uses two fixed noise draws per trajectory. Checkpoint choice minimizes mean validation weighted future flow loss, breaking ties by the earliest epoch. Epoch four is selected and its hash sealed before the evaluation stage. Evaluation uses the same inputs, noise and eight Euler integration steps for the original and adapted core.

Loading, storage and execution checks

The implementation validates expected checkpoint keys and shapes, loads checkpoint-backed tensors with bounded CPU buffers and rejects missing-weight or random fallback. Unsupported legacy modules are explicitly disabled. The compressed path stores 304 linear matrices in row-scaled FP8 representation while retaining biases, normalization and action parameters in BF16; weights are dequantized for the matrix multiplication. Core attention uses the recorded math backend and the VAE retains its native default backend. A prior incorrect VAE backend scope and a failed output-consistency audit are retained in the development history; the corrected complete entry matches saved staged predictions on the checked trajectory.

Quality metrics and uncertainty

For each trajectory we average squared RGB error over all three views, eight future sample positions and pixels in [0,1], excluding the conditioning frame. The primary estimate is the equally weighted mean over trajectories. PSNR is computed per trajectory as -10 log10(MSE) and then averaged; it is not obtained by transforming the overall mean. SSIM uses an eleven-pixel Gaussian window, sigma 1.5, population covariance and unit data range. The appendix retains all model and persistence means, including the unsuccessful baselines.

For a candidate with per-trajectory errors c and reference errors b, relative improvement is 100 times (1 - mean(c)/mean(b)). The recorded bootstrap resamples ten complete paired trajectories with replacement 10,000 times using seed 20261007. All views and future positions remain together. The reported 95% interval is a trajectory-resampling interval for one fitted model; it is not a confidence statement over random initializations. Multiple secondary comparisons are exploratory, and no confirmatory family-wise significance claim is made.

Benchmark extraction and version separation

We preserve the public leaderboard HTML and extract its rendered table and embedded records without a hidden API. All 60 entry names and order are compared, and 1,560 score/rank cells are checked with tolerance for two-decimal display rounding. We retain published overall ordinal positions, including tie ordering; metric ranks use one plus the count of strictly greater scores. We do not reconstruct TWB-Score by averaging displayed category scores because that does not reproduce the published aggregate. No official package hash is present in the public target record.

Data and code availability

The draft includes aggregate local metrics, the published leaderboard extraction, bibliographic records and checksums. The benchmark dataset and upstream model weights are obtained from their original providers under their respective terms. Raw videos, reference images, upstream weights, personal correspondence and private company records are not redistributed here. The KineJing inference implementation and adapter weights are not publicly released with this draft. Access arrangements sufficient for review and independent reproduction remain to be resolved before a journal submission; no such access is claimed to have been granted.

Acknowledgements and research assistance

We acknowledge the BWM and Wan contributors for the pretrained model components and the TriWorldBench authors for the evaluation framework. AI assistance was used for literature retrieval, artifact inspection, manuscript drafting and consistency checks, including the Scientific Agent Skills procedures [6]. The manuscript does not treat assistant-generated prose or adapters as evidence of model performance. Author verification and responsibility for the final content remain necessary.

Author contributions and competing interests

YiCheng Pan is the named author, affiliated with KineJing Research Team. A detailed contribution statement and disclosure of relevant commercial interests require author confirmation before submission. No institutional endorsement or external independent replication is implied by this draft.

Extended results

Method	Future MSE	PSNR (dB)	SSIM
Original BWM, local BF16	0.08517601	11.050	0.54946
KineJing Pro Local, BF16	0.02464687	16.355	0.63926
Pro Local, FP8 storage	0.02439503	16.377	0.64220
RGB first-frame persistence	0.02005718	17.281	0.69815
Latent persistence	0.01720288	17.899	0.67850
VAE reconstruction (oracle)	0.00059897	32.609	0.94000

All local raw-RGB comparisons. Lower MSE and higher PSNR/SSIM are better. Ten whole trajectories, one training seed, identical local inputs and sampling configuration. The reconstruction reference consumes future reference latents and is not a predictor.

Category	KineJing-Pro	BWM	Pro rank / 60
Tri-view consistency	66.60	81.87	51
Task alignment	50.79	86.05	56
Physical / 3D coherence	62.50	60.40	6
Motion quality	31.53	41.29	47
Temporal consistency	71.22	62.81	6
Visual quality	29.54	31.42	29

Complete six-category leaderboard profile, 25 October 2026 snapshot. These systems were not run under a matched local ablation.

References

1. Liu, Xuanyi et al.. TriWorldBench: A Tri-View Consistency Perspective on Embodied World Models. arXiv:2609.26314 (2026). https://arxiv.org/abs/2609.26314

2. BWM Team. BWM: A Low-Cost High-Fidelity World Simulator for Robot Learning. arXiv:2607.29302 (2026). https://arxiv.org/abs/2607.29302

3. Huang, Ziqi et al.. VBench: Comprehensive Benchmark Suite for Video Generative Models. arXiv:2311.17982 (2023). https://arxiv.org/abs/2311.17982

4. Yue, Hu et al.. EWMBench: Evaluating Scene, Motion, and Semantic Quality in Embodied World Models. arXiv:2505.09694 (2025). https://arxiv.org/abs/2505.09694

5. Shang, Yu et al.. WorldArena: A Unified Benchmark for Evaluating Perception and Functional Utility of Embodied World Models. arXiv:2602.08971 (2026). https://arxiv.org/abs/2602.08971

6. Kassis, Timothy et al.. Scientific Agent Skills: A Library of Procedural Knowledge for Research Agents. arXiv:2609.00065 (2026). https://arxiv.org/abs/2609.00065

7. Hu, Edward J. et al. LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685 (2021). https://arxiv.org/abs/2106.09685