
SmolWAM: A Latent World-Action Model for Robotic Manipulation on Edge Devices

Timi Owolabi Hikmah Olawore Frances Adelakun Ayotomiwa Oyewumi
ML Collective

{timiiowolabi, olaworehikmah, francesadelakun127, ayot.oyewumi}@gmail.com

Abstract

World-action models (WAMs) predict a robot’s actions together with the observations those actions will produce. However, the strongest models rely on video diffusion transformers that are far too computationally expensive for resource-constrained edge devices. We present SmolWAM, a latent WAM that we derive from the 2B-parameter Cosmos Policy by keeping its pretrained trunk and rebuilding its inference interface: single-frame visual latents, cached text features that preserve the trunk’s attention sinks, 12 of 28 transformer blocks, and 1,008 tokens from which actions and future visual latents are read out in one rectified-flow step. Trained in 3.6 GPU-hours, SmolWAM succeeds in 95.5% of 400 LIBERO episodes at 57 ms and 2.3 TFLOP per 16-action chunk with 2.4 GiB of GPU memory, $11\times$ faster than its teacher with $18.5\times$ fewer FLOPs and $3.2\times$ less memory, and its world head still forecasts scene motion. Code: github.com/iitimii/SmolWAM; project page: smolwam.vercel.app.

1 Introduction

World-action models (WAMs) predict a robot’s next actions together with the observations they should produce [16, 23, 41]. Cosmos Policy [16] builds one from the Cosmos-Predict2 video model [27] by treating actions, future observations, and values as latent frames of a single denoising sequence, and reaches 98.5% success on LIBERO [19]. Its inference stack, however, was built for video generation: a 4.8B-parameter T5-XXL text encoder [32], a VAE that encodes a 33-frame clip per query, and five denoising steps of a 2B-parameter transformer over 1,764 tokens, which cost 624 ms, 42.6 TFLOP, and 7.55 GiB per 16-action chunk on an A100 even with cached text features.

Efficient policies usually give up the world model or the video prior: compact vision-language-action (VLA) models such as SmolVLA [33] and TurboVLA [39] predict only actions, and FastWAM [42] skips future prediction at test time. We ask whether a strong pretrained WAM can itself be made small while still predicting the future.

We present SmolWAM (Fig. 1). Our premise is that much of the teacher’s cost comes from an interface designed for video generation, not from what control requires. SmolWAM reuses the trunk’s weights, token positions, and rectified-flow readout; it encodes single frames through a small adapter, replaces T5-XXL with cached BERT features [8] padded to preserve the trunk’s attention sinks, keeps 12 of 28 blocks, and reads actions from 28 state-conditioned queries. Most changes break the trunk zero-shot, so a distillation curriculum recovers from pruning before compaction.

We contribute (i) SmolWAM, which retains 97% of its teacher’s success at $11\times$ lower latency; (ii) a method, with a diagnostic behind each step, for deriving compact WAMs from pretrained video WAMs (Sec. 3), including three pitfalls (attention-sink padding, jointly fatal blocks, and chunks that cannot be cut short); and (iii) findings that the compressed world head preserves motion forecasting, whereas latent-action cascades or visual-reward fine-tuning degrade control (Sec. 4).

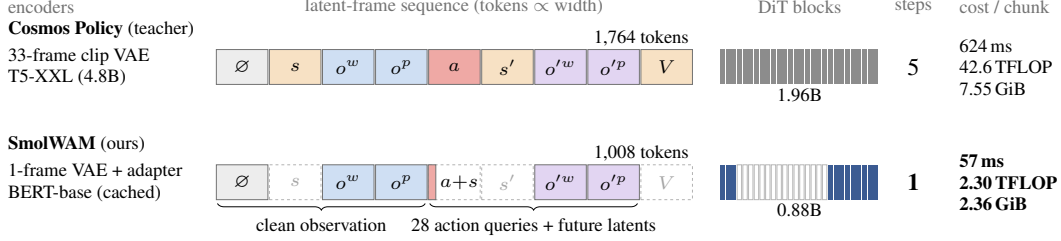


Figure 1: **From Cosmos Policy to SmolWAM.** Latent frames (width $\propto$ tokens), kept DiT blocks, denoising steps, and cost per 16-action chunk on an A100. SmolWAM drops the state and value frames (dashed), replaces the action frame with 28 state-conditioned queries ($a+s$), and reads actions and future camera latents (its world head) from one step.

2 Related Work

Generalist VLAs bring pretraining to control [2, 4, 10, 14, 29, 31], often with diffusion [7] or chunked [44] action heads; OpenVLA-OFT [15], TinyVLA [36], SmolVLA [33], and TurboVLA [39] speed them up; SmolWAM borrows TurboVLA’s principles but retains a world model. Latent world models learn dynamics in feature space [1, 9, 24, 45, 46], and GR-1 [37], PAD [11], and WorldVLA [5] generate future images and actions jointly. Recent WAMs build on video models [16, 41] and add action evaluation [47], modalities [28], object state [21], geometry [22], or latent actions [6]; Fast-WAM [42] drops future prediction at test time and Light-WAM [17] decodes actions from a smaller video backbone, whereas SmolWAM reads future latents from its action pass.

3 SmolWAM

3.1 Teacher: Cosmos Policy

On LIBERO, the video diffusion transformer (DiT) [30] of Cosmos Policy (28 blocks, 1.96B parameters) denoises nine latent frames of $14 \times 14 = 196$ tokens (Fig. 1): a blank frame, the robot state s , the wrist and third-person camera latents o^w and o^p , a chunk a of $H=16$ actions, and, at time $t+H$, the state s' , the camera latents o'^w and o'^p , and a value V . Images are encoded by the Wan2.1 VAE [34] from 33-frame clips, and the instruction enters through cross-attention to 512 T5-XXL embeddings. The model is a rectified flow [18, 20]: for $x_\tau = (1 - \tau)x_0 + \tau\epsilon$ it predicts the velocity $v_\theta(x_\tau, \tau) \approx \epsilon - x_0$, which gives the closed-form clean estimate $\hat{x}_0 = x_\tau - \tau v_\theta(x_\tau, \tau)$. Generated frames start at $\tau_{\max} \approx 0.988$ ($\sigma_{\max}=80$); the released pipeline takes five steps.

3.2 Deriving SmolWAM from the teacher

Single-frame latents. SmolWAM encodes each camera image once, as a single frame, with the Wan2.1 encoder (54M parameters). Such latents raise the teacher’s action error (normalized L1 of its one-step chunk to the demonstration) from 0.0089 to 0.038; a 0.5M-parameter adapter maps them to clip-mode latents and restores it to 0.0092.

Cached text that keeps the attention sinks. We replace T5-XXL with cached BERT-base [8] features of the instruction, projected to the context width. The trunk’s cross-attention has bias-free key and value projections, so T5’s ~ 500 zero-padded positions contribute zero logits and values and act as an attention sink [38]: truncating the context to 32 or 128 positions drops the teacher to 0/5 closed-loop successes, and restoring the removed softmax mass analytically returns it to 5/5, matching the padded attention to 6×10^{-8} . SmolWAM thus zero-pads the BERT features to 512 positions; six zero-initialized vision-to-language layers from TurboVLA [39] let the text attend to the images.

Boundary-preserving pruning. Block influence $1 - \cos(h_{\text{in}}, h_{\text{out}})$ [25] is highest for blocks 0, 1, and 25–27, while blocks 3–22 are individually close to the identity. Zero-shot screens (5 episodes each) show that removals do not compose: skipping block 1, 26, or 27 fails every episode, skipping

Table 1: **LIBERO success and inference cost.** Screen: 60 episodes; suites and Full: 400. Cost per 16-action chunk on one A100 (batch 1, bf16); Params excludes the VAE and text encoder; TFLOP counts network and VAE encoder; GiB: peak memory. *Unmodified weights, single-frame latents via the 0.5M-parameter adapter. [†]Suites and Full from an earlier evaluation pipeline, on an L4 (Cosmos Policy reports 98.5% [16]). [‡]Shared GPU. [§]Simulated: INT8 weights, bf16 kernels.

Model	Inference cost				Success rate (%)					
	Params	TFLOP	ms	GiB	Screen	Spatial	Object	Goal	Long	Full
Cosmos Policy (5 steps)	1.96B	42.6	624	7.55	96.7	97.0	100.0	99.0	97.0	98.2 [†]
Cosmos Policy (1 step, 1 frame)*	1.96B	7.52	126	4.39	100.0	–	–	–	–	–
Pruned + LoRA distillation	1.40B	–	–	–	–	83.0	100.0	82.0	85.0	87.5 [†]
SmolWAM	0.88B	2.30	57	2.36	93.3	99.0	100.0	94.0	89.0	95.5
+ 7.5k training steps	0.88B	2.30	60	2.36	93.3	94.0	100.0	94.0	89.0	94.2
INT8 weights [§]	0.88B	2.30	65 [‡]	2.36	91.7	–	–	–	–	–
8 blocks	0.60B	1.76	134 [‡]	1.84	83.3	–	–	–	–	–
SmolWAM-LA (4 steps)	0.92B	4.28	239	2.43	53.3	–	–	–	–	–

block 7, 14, or 20 alone fails none, and skipping four near-identity blocks (12–15) together fails all. SmolWAM keeps blocks 0–2 and 19–27 (0.85B parameters).

State-conditioned action queries. A zero-initialized MLP adds the robot state to 28 action queries that replace the 196-token action frame at its original positions. Beyond these queries we keep only the blank and image frames (1,008 tokens). Zero-shot, dropping the future-state and value frames barely changes the action error (0.009 to ≤ 0.019), but dropping the blank (0.30) or state (0.058) frame or compacting the vector frames to 28 tokens (0.21) does, so the layout is learned.

One-pass readout. SmolWAM evaluates its trunk once at $\tau_{\max}$ with fixed noise and reads the action chunk and future latents from $\hat{x}_0$, so its actions never condition on a denoised future; as FastWAM [42] also finds, acting does not need one, and the teacher solves all 60 screening episodes in one step (Table 1). The future latents form its world head, supervised in training and never decoded.

3.3 Distillation with a layout curriculum

We train on 68,830 observations sampled every fourth step from the 1,700 successful LIBERO demonstrations released with Cosmos Policy, using the teacher’s one-step predictions of the action chunk and future latents, a^T and z^T , cached at fixed noise (66 min on one A100). As in pruning and distillation of language models [12, 26], the loss $\|\hat{a}-a\|_1+\|\hat{a}-a^T\|_1+\|\hat{z}-z\|_1+\|\hat{z}-z^T\|_1$ matches the student’s one-step estimates to both the demonstrations and the teacher, plus 0.1-weighted terms for the state and value frames while they exist. Pruning and compaction are separate distribution shifts, so the student, initialized from the kept teacher blocks, trains in the teacher’s 1,764-token layout for the first 1,000 of 5,000 steps and in the compact layout afterward (AdamW; learning rate 2×10^{-5} for inherited and 10^{-4} for new weights; batch 32; bf16) in 3.6 h on one A100 (App. A.2).

4 Experiments

We ask (Q1) how much of the teacher’s success SmolWAM retains, (Q2) where its savings come from, (Q3) whether it remains a world model, and (Q4) whether richer WAM components help.

Setup. LIBERO [19] has four suites of ten tasks; we use the teacher’s observation pipeline (wrist and third-person images at 224 px, proprioception, 16-action chunks executed open loop), seed 195, and 10 trials per task (400 episodes). A 60-episode screening set (3 tasks per suite, 5 trials each) includes the tasks an earlier pruned student failed most, so it scores below the full protocol. Latency is the wall-clock time from observation to action chunk on one A100-SXM4-40GB (batch 1, bf16), including preprocessing and VAE encoding, with text features cached for every model; FLOPs are counted with PyTorch’s FlopCounterMode, attention included.

Q1: SmolWAM retains most of the teacher’s success. SmolWAM succeeds in 382/400 episodes (95.5%; 95% Wilson interval 93.0–97.1) against the teacher’s 393/400 (98.2%; 96.4–99.1); App. A.6 shows both from the same initial states. We detect no difference on Spatial and Object; the deficit

lies in the multi-step Goal (94 vs. 99) and Long (89 vs. 97) suites, where failed rollouts usually complete the first subgoal and stall on the second. An earlier student that removed only 8 blocks and recovered with LoRA [13] distillation (19.7M trainable parameters, under an hour on one L4, no demonstrations) on 414 teacher-rollout observations reached 87.5%; SmolWAM differs in distillation coverage, full fine-tuning, and the learned layout, effects we have not separated. Longer training does not help, and 8 blocks fall to 83.3% on the screen (Table 1).

Q2: Where the $11\times$ comes from. Running the teacher’s unmodified weights for one step on single-frame inputs cuts latency $5\times$ and compute $5.7\times$, to 126 ms and 7.5 TFLOP, with no loss on the screening set; with no adapter and the text truncated under the exact sink correction, the released model runs $8\times$ faster on an L4 (1.83 to 0.23 s; no failures in an 11-episode screen; App. A.3). SmolWAM’s trunk redesign adds $2.2\times$ in latency, $3.3\times$ in compute, and $1.9\times$ in memory; of its 57 ms, the transformer takes 37, the VAE encode 13, and preprocessing 7 (success against cost in App. A.1). At 57 ms per chunk it issues 17.7 chunks (282 actions) per second; because chunks execute open loop, the chunk rate bounds how often the policy reacts to new observations. Cutting chunks short does not help: re-querying the teacher after 8 of its 16 actions solves no screening episode at one or five denoising steps, so faster reaction needs training-time changes [3].

Q3: SmolWAM remains a world model. On 96 held-out contexts, SmolWAM’s forecast of the view 16 steps ahead lowers latent L1 by 24% relative to copying the current frame (0.101 vs. 0.133), and lowers the end-point error between RAFT [35] optical flow to the predicted and to the real future from 1.63 to 1.43 pixels. Decoded forecasts (Fig. 2) keep the scene and place the arm near its future pose but blur its appearance, and LPIPS [43] stays at the copy-frame level (0.128 vs. 0.115): the world head captures motion rather than texture.

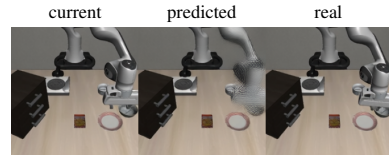


Figure 2: **World-head forecast**, decoded only for display.

Q4: Richer WAM components hurt at this scale. Porting ABot-M0.5 [6] onto SmolWAM (latent-action tokens, mixture-of-transformer towers, four-step flow matching, and dream forcing) yields SmolWAM-LA, which succeeds in 53.3% of screening episodes at 239 ms, and reward fine-tuning of its world head collapses success to 5.0–16.7% while worsening every image metric (App. A.4).

5 Limitations and Conclusion

Limitations. We evaluate SmolWAM only on LIBERO; future work should extend evaluation to additional robotics benchmarks such as RoboCasa and, ultimately, real-robot settings. We also observe that reducing transformer depth disproportionately affects long-horizon tasks, with the largest degradation on LIBERO-Long. Developing pruning strategies that preserve long-horizon control performance therefore remains an important open problem.

Conclusion. A pretrained video WAM tolerates large interface and depth changes when each is distilled rather than applied zero-shot, but not auxiliary objectives that train its shared trunk without an action anchor; SmolWAM keeps its world prediction and 95.5% LIBERO success at a fraction of the teacher’s cost.

References

- [1] Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, et al. V-JEPA 2: Self-Supervised Video Models Enable Understanding, Prediction and Planning. *arXiv preprint arXiv:2506.09985*, 2025.
- [2] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0 : A Vision-Language-Action Flow Model for General Robot Control. In *Proceedings of Robotics: Science and Systems (RSS)*, 2025.

- [3] Kevin Black, Manuel Y. Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. *arXiv preprint arXiv:2506.07339*, 2025.
- [4] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. *arXiv preprint arXiv:2307.15818*, 2023.
- [5] Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, Deli Zhao, and Hao Chen. WorldVLA: Towards Autoregressive Action World Model. *arXiv preprint arXiv:2506.21539*, 2025.
- [6] Ronghan Chen, Yandan Yang, Zuojin Tang, Dongjie Huo, Tong Lin, Haoning Wu, Haoyun Liu, Yuzhi Chen, Lulu Zheng, Botai Yuan, Tianlun Li, Mingxin Wang, Dekang Qi, Bin Hu, Wei Mei, Yuze Xuan, Haolong Yang, Yanqing Zhu, Mu Xu, Zhiheng Ma, and Xinyuan Chang. ABot-M0.5: Unified Mobility-and-Manipulation World Action Model. *arXiv preprint arXiv:2607.00678*, 2026.
- [7] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin CM Burchfiel, and Shuran Song. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.
- [8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.
- [9] Yuntian Gao and Xiangyu Xu. Fast LeWorldModel. *arXiv preprint arXiv:2606.26217*, 2026.
- [10] Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R. Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An Open-Source Generalist Robot Policy. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.
- [11] Yanjiang Guo, Yucheng Hu, Jianke Zhang, Yen-Jen Wang, Xiaoyu Chen, Chaochao Lu, and Jianyu Chen. Prediction with Action: Visual Policy Learning via Joint Denoising Process. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pages 112386–112410, 2024.
- [12] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*, 2015.
- [13] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations (ICLR)*, 2022.
- [14] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Open-VLA: An Open-Source Vision-Language-Action Model. In *Proceedings of the 8th Conference on Robot Learning (CoRL)*, volume 270 of *Proceedings of Machine Learning Research*, pages 2679–2713, 2024.
- [15] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-Tuning Vision-Language-Action Models: Optimizing Speed and Success. In *Proceedings of Robotics: Science and Systems (RSS)*, 2025.
- [16] Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos Policy: Fine-Tuning Video Models for Visuomotor Control and Planning. In *International Conference on Learning Representations (ICLR)*, 2026.

- [17] Ziang Li, Dongzhou Cheng, Yibin Wang, Shiyue Wang, Xiaoyang Xu, Lingxuan Weng, Juan Wang, and Jiaqi Wang. Light-WAM: Efficient World Action Models with State-Fusion Action Decoding. *arXiv preprint arXiv:2606.08242*, 2026.
- [18] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow Matching for Generative Modeling. In *International Conference on Learning Representations (ICLR)*, 2023.
- [19] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, volume 36, pages 44776–44791, 2023.
- [20] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow. In *International Conference on Learning Representations (ICLR)*, 2023.
- [21] Yushan Liu, Peibo Sun, Shoujie Li, Yifan Xie, Lingfeng Zhang, Xintao Chao, Shiyuan Dong, Fang Chen, Xiao-Ping Zhang, and Wenbo Ding. OA-WAM: Object-Addressable World Action Model for Robust Robot Manipulation. *arXiv preprint arXiv:2605.06481*, 2026.
- [22] Javier Alejandro Lopetegui Gonzalez, Paul Pacaud, and Cordelia Schmid. Spatially Aware World Action Model via Geometric Latent Diffusion. *arXiv preprint arXiv:2609.02531*, 2026.
- [23] Zuxing Lu, Hongjia Zhai, Guanzhi Wang, Huajian Zeng, Jiaqi Yang, Jingyu Liu, Lei Cheng, Yuntai Zhang, Yuheng Qiu, Zezhou Cheng, Ivan Laptev, Danfei Xu, Benjamin Riviere, Giuseppe Loianno, Eric Xing, and Xingxing Zuo. World-Action Models for Robot Learning and Control: A Survey. *arXiv preprint arXiv:2609.16074*, 2026.
- [24] Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. LeWorldModel: Stable End-to-End Joint-Embedding Predictive Architecture from Pixels. *arXiv preprint arXiv:2603.19312*, 2026.
- [25] Xin Men, Mingyu Xu, Qingyu Zhang, Qianhao Yuan, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. ShortGPT: Layers in Large Language Models are More Redundant Than You Expect. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20192–20204, 2025.
- [26] Saurav Muralidharan, Sharath Turuvekere Sreenivas, Raviraj Joshi, Marcin Chochowski, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro, Jan Kautz, and Pavlo Molchanov. Compact Language Models via Pruning and Knowledge Distillation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pages 41076–41102, 2024.
- [27] NVIDIA, Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, et al. Cosmos World Foundation Model Platform for Physical AI. *arXiv preprint arXiv:2501.03575*, 2025.
- [28] NVIDIA, Aditi, Niket Agarwal, Arslan Ali, Jon Allen, Martin Antolini, Adeline Aubame, Alisson Azzolini, Junjie Bai, Maciej Bala, et al. Cosmos 3: Omnimodal World Models for Physical AI. *arXiv preprint arXiv:2606.02800*, 2026.
- [29] Open X-Embodiment Collaboration, Abby O’Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandekar, et al. Open X-Embodiment: Robotic Learning Datasets and RT-X Models. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903, 2024.
- [30] William Peebles and Saining Xie. Scalable Diffusion Models with Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4195–4205, 2023.
- [31] Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient Action Tokenization for Vision-Language-Action Models. In *Proceedings of Robotics: Science and Systems (RSS)*, 2025.

- [32] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.
- [33] Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, Simon Alibert, Matthieu Cord, Thomas Wolf, and Remi Cadene. SmolVLA: A Vision-Language-Action Model for Affordable and Efficient Robotics. *arXiv preprint arXiv:2506.01844*, 2025.
- [34] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Fei Wu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and Advanced Large-Scale Video Generative Models. *arXiv preprint arXiv:2503.20314*, 2025.
- [35] Zachary Teed and Jia Deng. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. In *European Conference on Computer Vision (ECCV)*, pages 402–419. Springer, 2020.
- [36] Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Zhibin Tang, Kun Wu, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, Yaxin Peng, Feifei Feng, and Jian Tang. TinyVLA: Toward Fast, Data-Efficient Vision-Language-Action Models for Robotic Manipulation. *IEEE Robotics and Automation Letters*, 10(4):3988–3995, 2025.
- [37] Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing Large-Scale Video Generative Pre-training for Visual Robot Manipulation. In *International Conference on Learning Representations (ICLR)*, 2024.
- [38] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient Streaming Language Models with Attention Sinks. In *International Conference on Learning Representations (ICLR)*, 2024.
- [39] Hengyi Xie, Chenfei Yao, Xianjin Wu, Yingying Zhu, Dingkan Liang, Xiang Bai, and Han Ding. TurboVLA: Real-Time Vision-Language-Action Model at 32 Hz on an RTX 4090 with <1 GB VRAM. *arXiv preprint arXiv:2607.27205*, 2026.
- [40] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth Anything V2. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pages 21875–21911, 2024.
- [41] Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World Action Models are Zero-shot Policies. *arXiv preprint arXiv:2602.15922*, 2026.
- [42] Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-WAM: Do World Action Models Need Test-time Future Imagination? *arXiv preprint arXiv:2603.16666*, 2026.
- [43] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [44] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.
- [45] Ruijie Zheng, Jing Wang, Scott Reed, Johan Bjorck, Yu Fang, Fengyuan Hu, Joel Jang, Kaushil Kundalia, Zongyu Lin, Loïc Magne, Avnish Narayan, You Liang Tan, Guanzhi Wang, Qi Wang, Jiannan Xiang, Yinzhen Xu, Seonghyeon Ye, Jan Kautz, Furong Huang, Yuke Zhu, and Linxi Fan. FLARE: Robot Learning with Implicit World Modeling. In *Proceedings of the 9th Conference on Robot Learning (CoRL)*, volume 305 of *Proceedings of Machine Learning Research*, pages 3952–3971, 2025.
- [46] Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World Models on Pre-trained Visual Features enable Zero-shot Planning. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pages 79115–79135, 2025.

- [47] Pengfei Zhou, Shengcong Chen, Di Chen, Jiaxu Wang, Rongjun Jin, Bingwen Zhu, Yike Pan, Songen Gu, Kuanning Wang, Shufeng Nan, Xingyu Qiu, Chenhao Qiu, Pu Yang, Yunuo Cai, Jianxiong Gao, Yifan Li, Yanwei Fu, Xiangyu Yue, Zhi Chen, and Jianlan Luo. τ_0 -WM: A Unified Video-Action World Model for Robotic Manipulation. *arXiv preprint arXiv:2606.01027*, 2026.

A Additional results

A.1 Success against cost

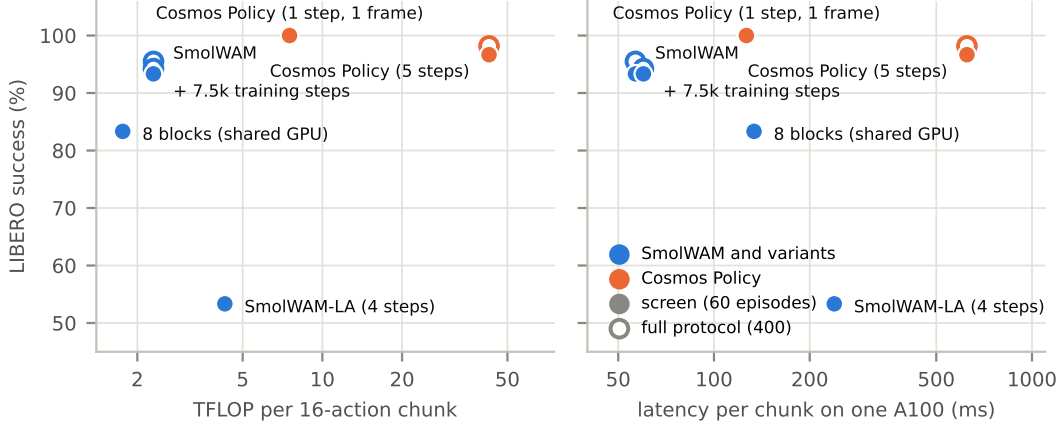


Figure 3: **LIBERO success against cost** for every row of Table 1 with an A100 cost. Filled markers are the 60-episode screen and hollow markers the 400-episode protocol (the teacher’s as in Table 1); the 8-block latency was measured on a shared GPU.

SmolWAM and its 7.5k-step continuation sit in the cheap, successful corner. The 8-block variant saves a further 0.5 TFLOP but loses 10 points on the screen, and SmolWAM-LA is both slower and less successful than SmolWAM. Per-channel INT8 quantization of the 172 largest linear layers shrinks the stored network from 1.64 to 0.85 GiB and costs one screening episode; simulated with bf16 kernels, it does not yet reduce latency.

A.2 Training curves

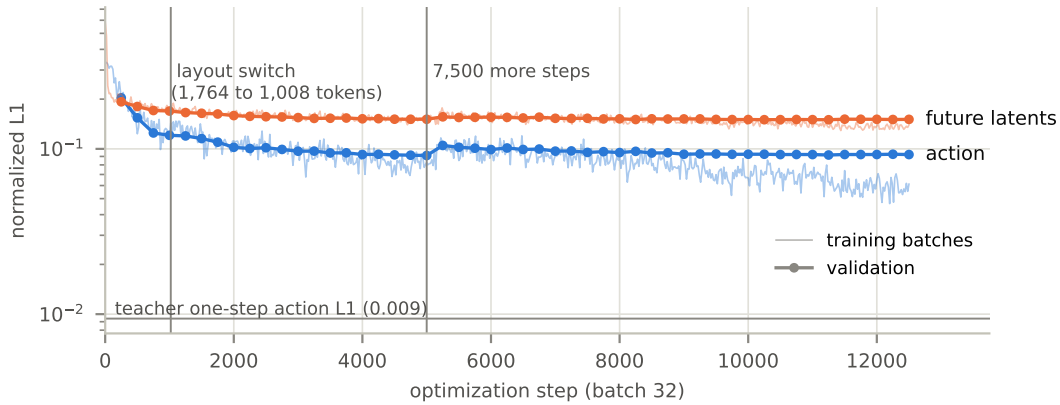


Figure 4: **Distillation of SmolWAM (batch 32)**: normalized L1 of the one-step action chunk and future latents against the demonstrations, on training batches and held-out validation rows. Vertical lines mark the switch from the teacher’s 1,764-token layout to the compact one, and the start of 7,500 further steps.

Validation action L1 falls fastest in the teacher’s layout, keeps falling after the switch to the compact layout, and settles near 0.09, ten times the teacher’s own one-step error, while SmolWAM still succeeds in 95.5% of episodes: offline L1 is a poor proxy for closed-loop success. The 7,500 further steps lower the training loss but neither the validation loss nor success (Table 1).

A.3 The zero-training ladder on an L4

Table 2: **Zero-training speedups on the released weights** (L4, uncontended single-query benchmark). Deviation: mean absolute difference from the five-step model’s actions (raw units) on the benchmark’s single observation; over a 161-observation offline set, one step deviates by 0.003. Closed-loop scores are single-trial screens, so they screen rather than resolve small differences.

Configuration (cumulative)	s/chunk	deviation	closed loop
Cosmos Policy 2B, 5 denoising steps (as released)	1.83	–	5/5, 3/3 Spatial
1 denoising step	1.22	0.005	5/5
+ no future-image decode	0.54	0.005	5/5
+ single-frame VAE encode + exact text sink	0.23	0.044	5/5, 3/3 Spatial, 3/3 Goal

Each change is applied to the released model with no training. At one step, a query on the L4 spends 0.2 s in the transformer, 0.35 s encoding 33-frame clips, and 0.68 s decoding future images that the controller never reads. The released evaluation pipeline decodes them; the teacher pipeline of Table 1 does not.

A.4 Richer WAM components

SmolWAM-LA adds 39M parameters to SmolWAM: tokens from a frozen 32-dimensional latent-action model behind an asymmetric attention mask, mixture-of-transformer towers for the latent-action, arm, and gripper streams, four-step conditional flow matching, and dream forcing, which trains the action head on predicted latents. Its one-step action estimate matches SmolWAM’s offline (validation L1 0.089 vs. 0.091), but four-step sampling raises the error to 0.155, and sampling with one or two steps recovers only 63.3% and 70.0% of screening episodes; its latent-action predictions are worse than predicting the mean code (MSE 1.15, variance 0.64), and its world head is no better (latent L1 0.102). Reward-weighted fine-tuning of the world head under five rewards built from optical flow, depth [40], and LPIPS worsened every image metric (LPIPS from 0.130 to 0.167–0.177) and collapsed success to 5.0–16.7% within 300 steps, likely because the objective has no action term while the shared trunk trains.

A.5 Single-flag ablations

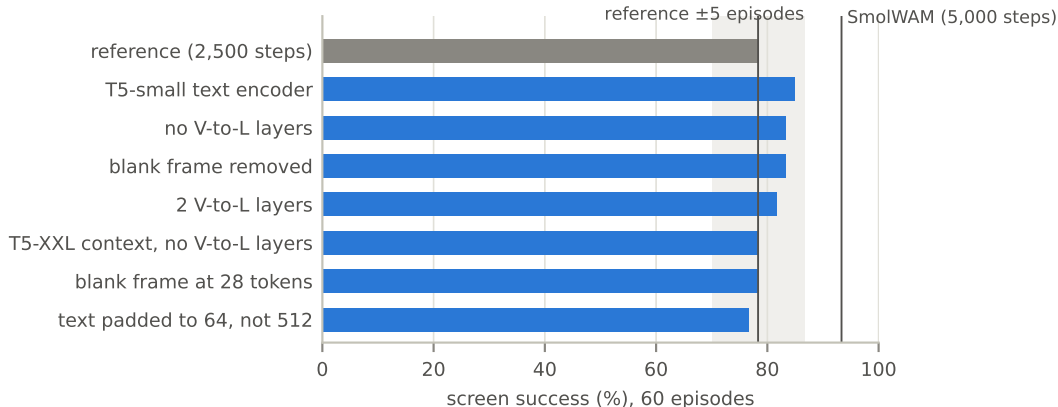


Figure 5: **Single-flag ablations** of the SmolWAM recipe, each trained for 2,500 steps (half the main run) and scored on the 60-episode screen. The shaded band spans five episodes either side of the reference.

Every change lands within five episodes of the reference (78.3%), so at this budget the screen cannot separate these choices; the full-budget model scores 93.3% on the same screen. Six planned ablations, among them removing the world head, the distillation targets, and the layout curriculum, were not run.

A.6 Rollouts

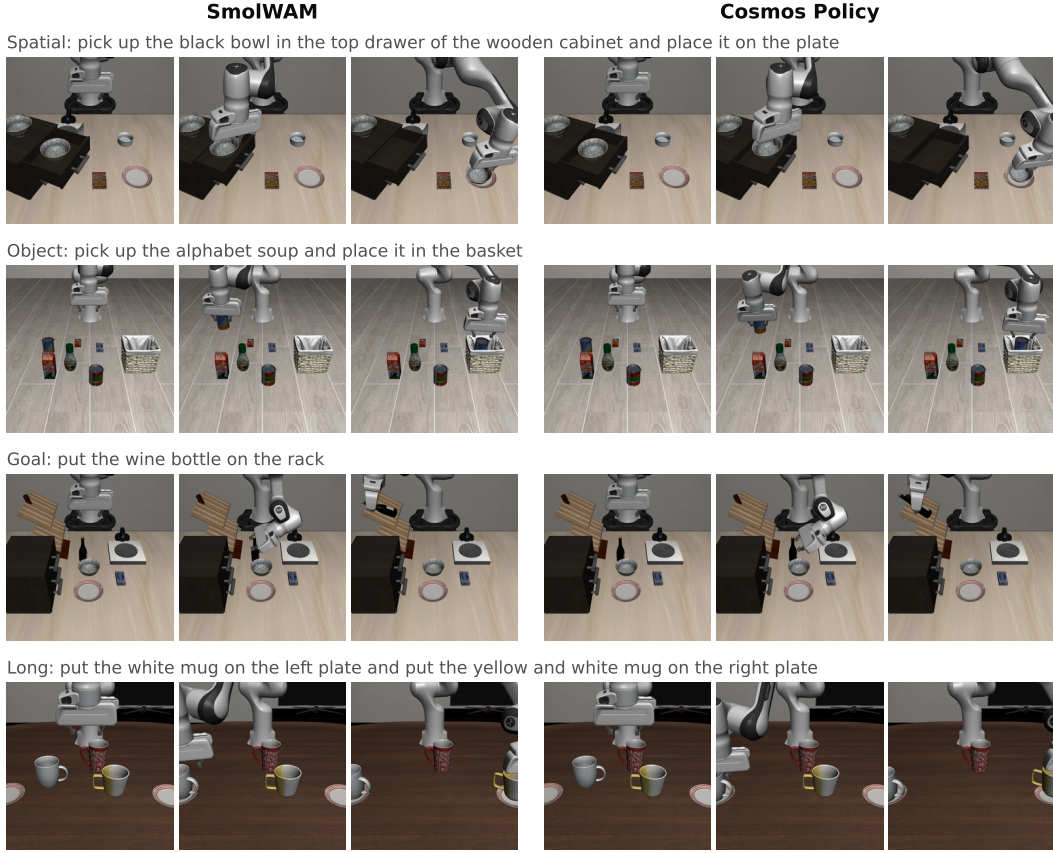


Figure 6: **Rollouts** of SmolWAM (left) and Cosmos Policy (right) from the same initial state, one task per suite: first, middle, and last frames. Both succeed on all four, in 131, 147, 154, and 234 steps (SmolWAM) and 143, 147, 138, and 247 steps (Cosmos Policy). Recorded on an Apple M2 Pro.