DAM-VLA: Decoupled Asynchronous Multimodal Vision Language Action model

Pankhuri Vanjani[1], Zhuoyue Li[1*], Jakub Suliga[1*], Moritz Reuss[2], Gianluca Geraci[1], Xinkai Jiang[1], Rudolf Lioutikov[1,3]
[1]Intuitive Robots Lab, Karlsruhe Institute of Technology (KIT), Germany [2]NVIDIA [3]Robotics Institute of Germany

***Abstract*—Vision-language-action (VLA) models inherit a shared synchronous clock from vision-language pretraining, processing every input at one rate. This is misaligned with physical interaction, where a high-frequency modality changes at hundreds of hertz, vision evolves more slowly, and language stays constant across an episode. A synchronous VLA oversamples slow modalities, undersamples fast ones, and caps action generation at the lowest effective frequency. We hypothesize that decoupling temporal processing per modality, letting each update and retain information at its own sensor rate, yields stronger representations and more robust control. We present DAM-VLA, which maintains per-modality latent buffers refreshed at sensor rates and read continuously by the action head, integrating new high-frequency modalities through gated cross-attention that leaves the pretrained backbone intact. Across seven contact-rich real-world manipulation tasks, DAM-VLA more than doubles the average success rate of the strongest synchronous baseline (95.2% vs. 40.95%) while sustaining smooth, reactive 100 Hz control.**

I. Introduction

Robot manipulation requires models that can act reliably across diverse tasks and environments. Vision-Language-Action (VLA) models have emerged as a promising foundation for this goal, leveraging pretrained vision-language representations to generalize across tasks. However, the architectural assumptions underlying these models were shaped by language and vision pretraining. In those domains a single, uniform processing clock is a natural fit.

Current VLA architectures are all *synchronous* with respect to sensor modalities. At each fixed timestep, every input, vision, force-torque, proprioception, tactile, is encoded together and passed to the policy. This design is inherited from Vision-Language Models (VLMs) optimized for language-grounded perception, not for real-time sensorimotor control. The result is a fundamental mismatch between the model's internal rhythm and the physical structure of the real world and its sensors.

This mismatch arises because heterogeneous sensors produce meaningful information at fundamentally different rates. As one concrete example, force-torque signals capture contact transients at 100-500 Hz, while RGB cameras provide informative signals at 3-10 Hz. Each modality also carries meaning over a different temporal horizon. A force spike is relevant for milliseconds, while a scene layout remains stable across seconds. A synchronous fixed-clock model is doubly misaligned: it ignores both the natural sensor rate at which each modality becomes informative and the natural horizon over which its information remains relevant. This mismatch produces three major challenges. First, **Redundant compute**: the expensive VLM encoder re-processes semantically identical frames every step, wasting the compute budget needed for higher action rates. Second, **Cross-modal rate mismatch**: a uniform clock simultaneously undersamples fast-updating modalities and oversamples slow-updating ones. No modality is processed at the rate its signal structure demands. Third, **Action latency**: policy execution is gated on the arrival of a fully synchronized observation bundle. Waiting for the slowest modality directly increases latency and reduces effective control frequency.

Decoupling is the right architectural principle for multimodal robot learning. Rather than forcing all modalities into a synchronous token stream, each modality updates, retains, and contributes information according to its own temporal dynamics. This principle spans two dimensions. 1) **asynchrony**: each modality is processed at its natural sensor instead of down- or upsampling it to a shared clock. 2) **temporal context**: each modality maintains a buffer sized to its meaningful horizon. Short- and long-lived signals are preserved at their appropriate timescale. These dimensions lead to representations that reflect the information structure of the sensors instead of synchronous projections.

Following these insights we present DAM-VLA, a decoupled asynchronous VLA architecture for dexterous robot manipulation. In this work the heterogeneous modality set comprises visual streams and forces/torques as a representative high-frequency modality updated at the natural sensor rate. The architecture is not specific to this sensor set. Other modalities can be leveraged at their sensor rates via independent memory buffers. **Our contributions are:**

1) **Asynchronous multimodal architecture**: We introduce a decoupled processing design in which each modality stream updates independently at its natural frequency, with per-modality temporal context windows sized to the meaningful horizon of that signal.
2) **Improved performance through asynchronous representations**: By preserving the natural information structure of each sensor, DAM-VLA learns multimodal representations that lead to higher task success rates than synchronous baselines.
3) **Reduced inference latency**: Decoupling action generation from slow modality update cycles, such as periodic VLM re-encoding, enables the policy to act continuously at the control frequency, reducing end-to-end latency and increasing effective control frequency.

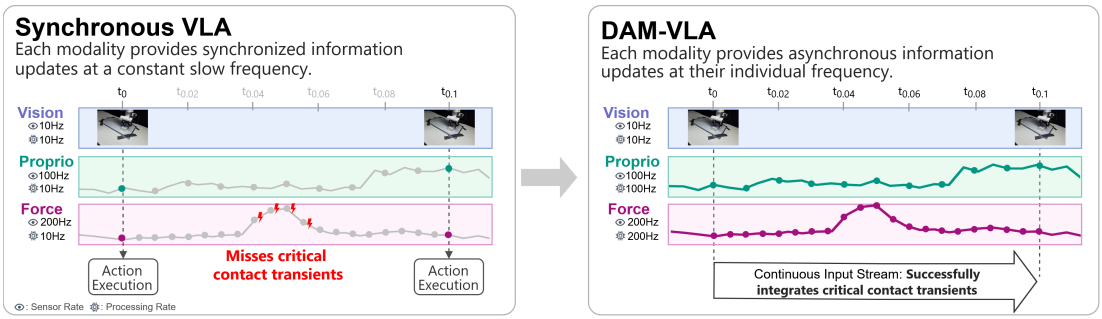

Fig. 1. Standard synchronous VLAs operate on a single slow clock, missing critical high-frequency contact transients. In contrast, DAM-VLA updates each modality asynchronously at its natural sensor rate, successfully capturing fast dynamics and enabling smooth, continuous control.

We evaluate DAM-VLA on seven contact-rich, real world manipulation tasks, using X-VLA [26] as the VLA backbone, with force/torque as the high-frequency modality. The results show that asynchronous decoupled processing improves average success rate by **54.25%** while running smoothly at 100Hz demonstrating, the practical value of multi-rate VLA design.

II. Related Work

Asynchronous and Efficient VLA Inference. Early work on asynchronous robot learning addresses the mismatch between slow VLA inference and real-time control demands. Black et al. [1] and VLA-RAIL [25] overlap chunk generation and execution to maintain continuity, while A2C2 [14] attaches a lightweight correction head that injects time-aware residuals at every control step, recovering reactivity under inference delay. A complementary slow-fast thread splits the model into a semantic VLM pathway and a fast action pathway: FiS-VLA [2] shares parameters between both at a fixed 1:4 frequency ratio, while Realtime-VLA V2 [21] does system-level optimizations. DuoCore [28] bridges them via a latent buffer achieving 30Hz whole-body control. On the efficiency side, VLA-Cache [20] and SD-VLA [13] avoid recomputing static visual tokens across frames,Realtime-VLA V2 [21] focuses on system level optimizations, and VLASH [18], FASTER [11] address temporal misalignment between prediction and execution.

These works either treat asynchrony as a system-level scheduling problem or focus on a single fast-slow split at a fixed ratio. All assume a synchronized observation bundle is available at each inference cycle. We study a complementary problem: heterogeneous sensory streams that update at different rates. None consider the general case of heterogeneous sensors streaming at different rates, nor do they study how the choice of modality integration mechanism affects the quality of the learned multimodal representation. Rather than requiring complete observations at every step, DAM-VLA maintains per-modality latent buffers updated at sensor rates, addressing both issues jointly. Outside the VLA setting, ManipForce [6] shows that training with native async RGB and force/torque streams outperforms downsampled baselines, directly motivating our multi-rate approach.

Multimodal Sensing and Temporal Context in VLAs. Recent work has extended VLA architectures with additional sensing modalities for contact-rich manipulation, with force and tactile sensing as the most commonly integrated examples. TA-VLA [23] and ForceVLA2 [7] inject torque at different architectural points, the latter adding force-based prompts into the VLM for hybrid force-position control. TacVLA [22] fuses tactile tokens with a hard contact gate, while FAVLA [8] uses the slow VLM to predict near-future force variation to schedule the fast action expert. FD-VLA [24] takes the opposite extreme, distilling force entirely from vision without a physical sensor. Across all these works, force signals are fused synchronously into a shared token stream. To handle temporally extended tasks, recent VLAs [19, 17, 27, 10, 5, 15, 9, 12, 3, 4] integrate latent memory through recurrent encoders, memory banks, or attention-based compressors, yet universally enforce a synchronous update regime that ties all sensor streams to a single clock.

DAM-VLA addresses both. Each modality is maintained as an independently buffered asynchronous stream updated at its sensor rate, and integrated via Gated Cross-Attention (GCA) pathways with gating strategies matched to each modality's signal structure. We use force/torque as a new, high-frequency modality.

III. Method

A. Problem Formulation

We consider a policy that receives observations from M heterogeneous sensing modalities $\mathcal{M} = \{m_1, \ldots, m_M\}$, each producing meaningful information at distinct rates. In our experiments, $\mathcal{M}$ contains a third person scene camera at 25Hz, a wrist mounted camera at 25Hz, force/torque at 100Hz sourced directly from the Franka's internal sensor, proprioception at 100Hz and a language instruction that is static for the duration of each episode. A synchronous VLA queries all modalities at a single control frequency, treating every observation as if it contains every relevant information since the last step. This is suboptimal for heterogeneous sensing. It oversamples redundant visual data, undersamples fast contact transients, and blocks action generation until a complete observation bundle is available. We decouple each modality's update rate

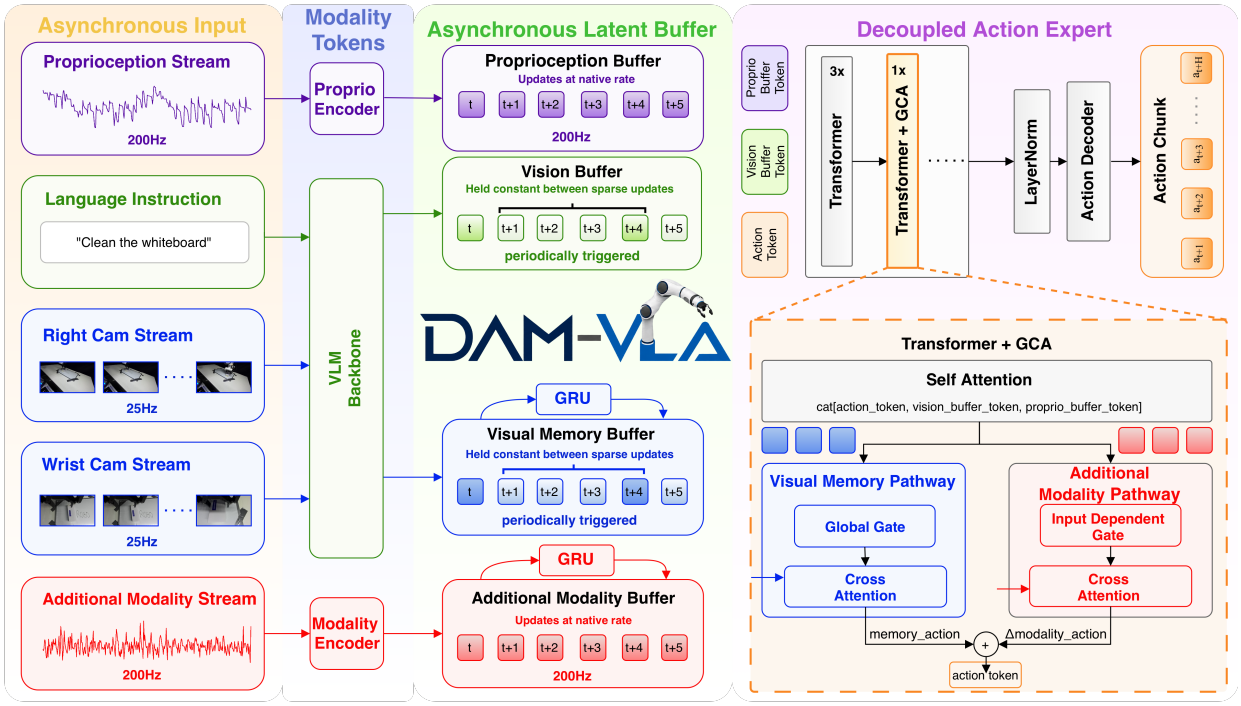

Fig. 2. **DAM-VLA architecture.** Each modality stream encodes tokens into independent latent buffers at their sensor rate: vision periodically, proprioception and force/torque at high frequency. The action expert reads all buffers continuously via parallel GCA pathways, a global-gate pathway for visual memory and an input-dependent gate pathway for force/torque, adding new modalities through dedicated cross-attention modules that preserve the pretrained self-attention structure.

from the control rate, maintaining a per modality latent context z_{m} in a shared buffer. Between update events, the cached context is held as a latent buffer and read by the action head at every control step. The action head is conditioned on this latent buffer, so that action generation is not blocked by individual modality's update rate.

B. DAM-VLA Architecture

a) Asynchronous Data Collection.: Rather than synchronising all sensor streams to a single timestamp before recording, we collect each modality independently at its sensor rate and store observations with per-modality timestamps. During training, we construct the observation context for each action label by fetching a fixed history window from each modality at its natural temporal resolution: sparse visual history of 16 frames at 25Hz, corresponding to approximately 0.64 s of semantic visual context and dense proprioceptive and force/torque histories of 96 samples each at 100Hz, corresponding to approximately 0.96 s of high-resolution state and contact history which captures fine-grained short-horizon dynamics. Vision provides semantic context over a longer horizon while force and proprioception provide high-resolution contact and state information over shorter periods.

b) Multimodal asynchronous Latent buffer: Continuous re-encoding of all modalities at full control frequency is wasteful. Vision changes slowly between manipulation phases, while force at contact change an order of magnitude faster. We maintain a shared latent buffer $\mathcal{B} = \{Z^m\}_{m \in \mathcal{M}}$ that holds one token sequence per modality. Each $Z^m \in \mathbb{R}^{N_m \times d}$ is refreshed at a modality-specific rate. At every inference step the tokens of each Z^m are processed but not consumed, i.e., if the inference frequency is higher than the sensor update the same sensor input is read again. The action head reads the entire buffer at every inference step, fully decoupling action generation from any individual modality's encoding schedule. This work uses these modalities:

1) Language tokens: For each episode a language instruction is encoded once at the beginning.

2) Visual tokens: The primary camera and the wrist camera are encoded by a vision-language encoder to generate a sequence of patch tokens. To avoid redundant re-encoding of semantically identical frames, the visual encoder is invoked every 4 inference steps to get VLM embeddings.

3) Proprioception tokens: Joint states are concatenated with the action tokens as in X-VLA [26], but read at the full 100Hz control rate, then encoded and processed in the asynchronous latent buffer.

4) Force tokens: Joint-torque readings arrive at the full control frequency and are smoothed via exponential moving average before being accumulated in a rolling buffer. A Gated Recurrent Unit (GRU) encodes this buffer and cross-attention over force registers compresses it to Z^{ft}, updated every control step independently of the visual schedule.

DAM-VLA also maintains a short-term visual memory for context across sparse updates: each visual update appends the new frame embedding to a rolling buffer of the K most recent ones, which a GRU encodes and learned-query cross-attention

compresses to N_{mem} tokens, producing Z^{mem}. Summarizing K frames rather than one snapshot, Z^{mem} stays valid while held constant between multiple inference updates.

c) Dual-pathway modulation via gated cross-attention.: Inputs from all modalities are jointly processed in the action expert of the VLA. Most naive baselines concatenate these heterogeneous information streams into a single flat sequence. This design, inherited from VLM token processing, is ill-suited to asynchronous updates and information decoupling. Also it forces new modality tokens into pretrained self-attention weights that have never seen them, risking corruption of pretrained representations. Instead, DAM-VLA augments the standard token-sequence input with two novel conditioning mechanisms: parallel GCA pathways that modulate the action expert on memory and force, inserted at every four transformer block of the action expert. This preserves pretrained weights untouched while allowing new modalities to inject residual corrections onto action tokens only.

Visual Memory pathway. Compressed visual memory tokens Z^{mem} condition action tokens $Z^{(\ell)}$ via a zero-initialized residual:

$$Z^{(\ell+1)} = Z^{(\ell)} + \tanh(\alpha)\ \mathrm{CA}\big(\mathrm{LN}(Z^{(\ell)}),\ Z^{\text{mem}}\big), \quad (1)$$

where α is a learned scalar initialized to zero so the pathway contributes nothing at the start of training and grows gradually without disrupting pretrained representations. A global gate is appropriate here because temporal visual context is relevant throughout the complete episode.

Additional modality pathway.: Other modalities use the same insertion points but an input-dependent gate, which we demonstrate with force tokens Z^{ft}.

$$Z^{(\ell+1)} = Z^{(\ell)} + \sigma\big(W\,\bar{z}^{\text{ft}}\big)\ \mathrm{CA}\big(\mathrm{LN}(Z^{(\ell)}),\ Z^{\text{ft}}\big), \quad (2)$$

where $\bar{z}^{\text{ft}}$ is the mean-pooled force token and the sigmoid gate is initialized near-closed. A static gate would be driven open by contact-phase gradients and closed by free-space gradients, converging to a compromise that either leaks noise during free motion or under-weights force during contact. The input-dependent gate lets the network learn *when* force is informative without requiring explicit contact detection. Critically, force queries the pre-memory-update action tokens $Z^{(\ell)}$ rather than the memory-updated tokens, computing a pure additive delta

$$\Delta^{\text{ft}} = \mathrm{CA}\big(\mathrm{LN}(Z^{(\ell)}),\ Z^{\text{ft}}\big) - Z^{(\ell)}, \quad (3)$$

that is added on top of the memory update, keeping the two conditioning pathways orthogonal and preventing cross-modal entanglement. The force gate responds to raw contact state, not to a signal already mixed with visual memory context to ensure reactive response at high frequency.

d) Training- and inference-time asynchrony: During training, all modalities are first aligned to a common 100 Hz timeline for consistent action labeling. Visual observations are then sampled with stride S, recovering a sparse history matching the camera's sensor rate, while force is sampled consecutively at the full 100 Hz. This mirrors the inference-time buffer, where vision updates sparsely and force updates every control step. We use asynchronous, delay-aware execution [16] and characterize horizon-dependent replanning, as studied for X-VLA in [11].

Config.	**Isolates**	**Async**	**Force**	**Mem.**	**Integ.**
X-VLA$_{25}$	std. VLA regime (25 Hz)	✗	✗	✗	–
X-VLA$_{100}$	naive high-freq. (100 Hz)	✗	✗	✗	–
X-VLA$_{AFM}$	concat. baseline	✓	✓	✓	cat
DAM-VLA$_{/F/M}$	async. alone	✓	✗	✗	–
DAM-VLA$_{/F}$	memory contribution	✓	✗	✓	GCA
DAM-VLA$_{/M}$	force contribution	✓	✓	✗	GCA
DAM-VLA (Ours)	full model	✓	✓	✓	GCA

TABLE I
EVALUATED CONFIGURATIONS. ALL SHARE THE X-VLA BACKBONE [26], TRAINING DATA, AND TASK SPLIT. *cat*: MODALITY TOKENS CONCATENATED INTO THE INPUT SEQUENCE; *GCA*: OUR GATED CROSS-ATTENTION.

IV. EVALUATION

We evaluate DAM-VLA, our proposed asynchronous decoupled Multimodal VLA with per-modality latent buffering, against synchronous baselines across a suite of manipulation tasks. Our evaluation is structured around four research questions: **(RQ1)** Does synchronous processing limit manipulation performance, and does naive frequency scaling resolve this? **(RQ2)** Does asynchronous decoupling affect task performance? **(RQ3)** Can high frequency modalities, e.g. forces, combined with individual memory improve performance? **(RQ4)** Does the integration mechanism for additional modalities affect task performance in a decoupled asynchronous backbone?

A. Experimental Setup

We conduct experiments on a Franka Emika Panda arm using a fixed third-person scene camera and a wrist-mounted camera. RGB streams are recorded at 25 Hz and proprioception at 100Hz. Force/torque readings are sourced directly from the Franka's internal sensor at 100 Hz. We evaluate on seven manipulation tasks spanning a range of contact requirements, reactivity demands, and precision constraints: *Scarf* folding, *Whiteboard* cleaning, *Button* pressing, *Handwash* top press, *Socket* insertion, *Sweep* beads into a dustpan, and *Lego* piece arranging.

We measure **(i)** task success rate (%) over 15 trials per task and **(ii)** average episode length (seconds), compared against teleoperation demonstrations as a natural-pacing reference. In the Appendix we further compare and show improvements in trajectory smoothness across two smoothness metrics.

a) Baselines and Ablations: We evaluate six configurations (Table I), each isolating one design axis of DAM-VLA, all built on the same X-VLA backbone [26] and finetuned with

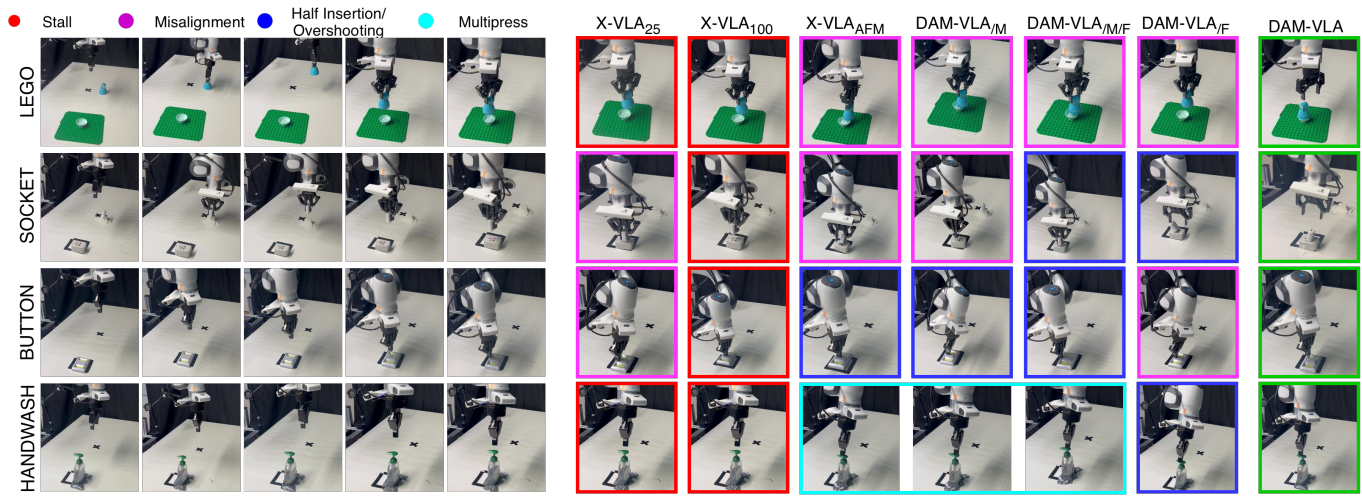

Fig. 3. Qualitative rollout comparison across a subset of manipulation tasks. green boxes indicate successful DAM-VLA executions and other color boxes include the failure modes across different policies.

identical data and splits. The X-VLA baselines test the standard synchronous approach and a naive high-frequency variant that upsamples visual frames to 100Hz without redundancy suppression or temporal memory. X-VLA$_{AFM}$ is the strongest concatenation baseline. It accesses the same force and memory information as DAM-VLA but integrates it via a flat token sequence in contrast to DAM-VLA's gated cross-attention. The remaining ablations isolate the individual contributions of asynchronous decoupling (DAM-VLA$_{/F/M}$), force (DAM-VLA$_{/F}$), and visual memory (DAM-VLA$_{/M}$).

All reported results run on a 100 Hz controller: the X-VLA$_{25}$ and X-VLA$_{100}$ baselines replan at $\approx$1 and $\approx$3.5 Hz, and DAM-VLA (s=22) at $\approx$ 5.5 Hz with smooth, stable execution. To probe the frequency limit, we additionally ran DAM-VLA on a 200 Hz controller, where it stayed reactive from $\approx$ 8 Hz (s=22) up to $\approx$ 17 Hz (s=6). Force and proprioception enter the buffer at 200 Hz and are read every inference step, decoupling the input rate from replanning.

B. Result Analysis

RQ1: Synchronous processing hits a performance ceiling despite frequency scaling.: Table II shows that X-VLA$_{25}$ achieves an average success rate of 40.95%, with strong performance on visually-guided tasks such as whiteboard cleaning (86.7%) and sweep (100%), but near-zero performance on tasks requiring precise contact: handwash (0%), Lego (0%), and socket insertion (6.7%). Increasing control frequency to 100 Hz drops average success further to 21.9%, with degradation across almost all tasks. Notably, sweep falls from 100% to 53.3% and whiteboard from 86.7% to 13.3%, showing that higher frequency actively hurts even on tasks where the synchronous baseline was strong. X-VLA$_{100}$ was trained with visual observations naively upsampled to match the 100 Hz control rate. At this rate, identical frames are paired with different action labels, creating a contradictory training signal that causes the policy to predict small hesitant movements rather than committing to actions. This redundant-frame bias stalls execution and produces jerky motion (Fig.3). These results confirm that naive frequency scaling is not a viable path to high-frequency manipulation, and that the bottleneck is architectural rather than a matter of controller rate.

RQ2: Asynchronous decoupling alone recovers performance lost by naive frequency scaling.: DAM-VLA$_{/F/M}$ isolates the effect of asynchronous decoupling without any additional modality. It runs at 100 Hz with vision updated sparsely and proprioception updated at every control step. It achieves 40.0% average success, nearly matching X-VLA$_{25}$ at 40.95% while running at higher frequency, and outperforming X-VLA$_{100}$ at 21.9%. At inference, vision is cached than re-encoded every step, avoiding the execution collapse seen in X-VLA$_{100}$. Decoupling also begins to unlock tasks that synchronous baselines could not do at all: button press improves from 13.3% to 40.0% and handwash from 0.0% to 20.0%. Some regression remains on visually-guided tasks like sweep (66.7% vs 100.0%), since no memory pathway preserves visual context between sparse updates.

RQ3: Additional modalities at their sensor rates further improve the decoupled foundation.: Building on the decoupled foundation of DAM-VLA$_{/F/M}$ (40.0%), every configuration that adds modality information improves: X-VLA$_{AFM}$ (54.3%), DAM-VLA$_{/F}$ (58.1%), DAM-VLA$_{/M}$ (66.7%), and the full DAM-VLA (95.2%) This confirms the decoupled architecture is a foundation additional modalities can build on, each contributing independently. Memory alone provides meaningful gains over the decoupled baseline. DAM-VLA$_{/F}$ achieves 100% on scarf and sweep and 86.7% on button press, but falls short on contact-critical tasks: partial presses without depth regulation on button, over-pressing that spills liquid on handwash (Fig. 4), and 0% on Lego despite reaching the target (Fig. 3). Force alone also improves over the decoupled baseline but introduces distinct failures without memory. DAM-VLA$_{/M}$ achieves 86.7% on button and 80.0% on handwash,

Model	Scarf	Whiteboard	Button	Handwash	Lego	Socket	Sweep	Avg.
Baselines								
X-VLA$_{25}$	80.0	86.7	13.3	0.0	0.0	6.7	100.0	40.95
X-VLA$_{100}$	80.0	13.3	6.7	0.0	0.0	0.0	53.3	21.9
Ours								
X-VLA$_{AFM}$	100.0	73.3	13.3	86.7	0.0	6.7	100.0	54.3
DAM-VLA$_{/F/M}$	80.0	66.7	40.0	20.0	0.0	6.7	66.7	40.0
DAM-VLA$_{/F}$	100.0	73.3	86.7	40.0	0.0	6.7	100.0	58.1
DAM-VLA$_{/M}$	100.0	86.7	86.7	80.0	13.3	13.3	86.7	66.7
DAM-VLA (Ours)	**100.0**	**100.0**	**93.3**	**100.0**	**93.3**	**80.0**	**100.0**	**95.2**

TABLE II
TASK SUCCESS RATE (%) PER CONFIGURATION OVER 15 TRIALS EACH. BEST RESULT PER COLUMN IN **BOLD**.

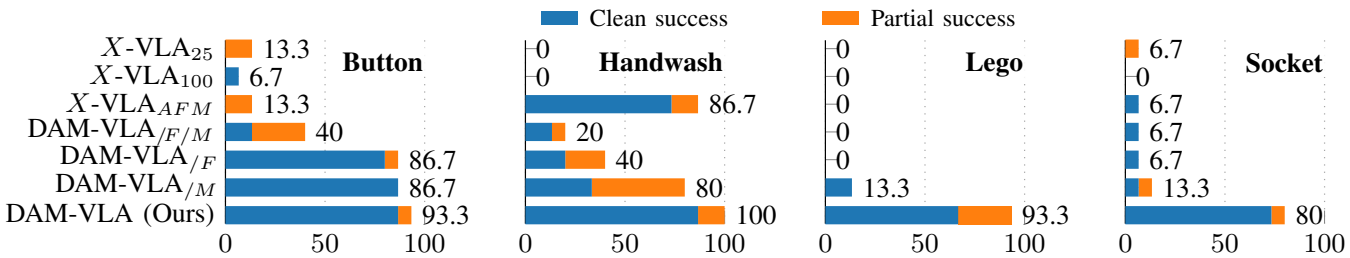

Fig. 4. Success rates across different tasks and model configurations. Blue indicates clean success, while orange indicates partial success.

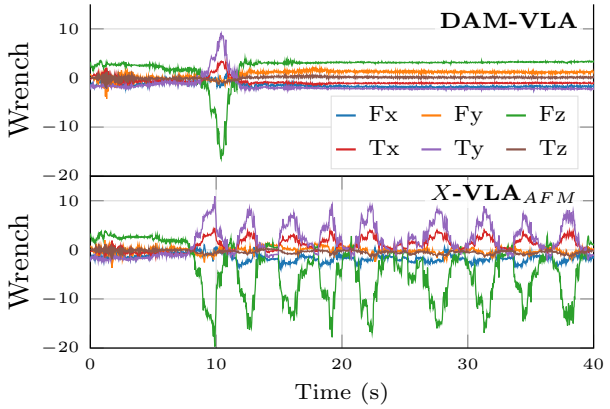

Fig. 5. Handwash execution wrench: DAM-VLA makes a single clean press, whereas X-VLA$_{\text{AFM}}$ repeatedly presses and retracts (multipress) over a longer episode.

but in 46.67% of rollouts presses repeatedly, unable to retain that contact was made, and on Lego overshoots, failing to align. The full model resolves both failure modes by combining force-guided contact with memory-stabilized sequencing. On sweep, DAM-VLA$_{/F}$ achieves 100% but takes 40 s per episode versus 22.5 s for DAM-VLA, slowing at every contact boundary without force. Memory stabilizes sequential context and prevents repetitions. Force provides the contact signal needed to terminate interactions precisely. Together they reach 93.3% on Lego and 80.0% on socket where all synchronous baselines scored near zero.

RQ4: Gated cross-attention preserves pretrained representations under new modalities.: X-VLA$_{AFM}$ has the same force and memory inputs as DAM-VLA but concatenates them into one flat token sequence. Despite identical information, it degrades across all tasks (54.3% vs 95.2%). Pushing unseen new tokens through pretrained self-attention disrupts the backbone's visual-language features. The policy reaches targets but cannot finish interactions, under-pressing on button, repeatedly pressing and retracting on handwash (Fig.5), and stalling on whiteboard. GCA avoids this by adding new tokens as zero-initialized residuals, leaving pretrained weights untouched. The gap shows multimodal gains depend the information itself, but how it enters a pretrained backbone.

V. CONCLUSION

We introduced DAM-VLA, a VLA built on a simple principle: each modality should update and be remembered at its own natural sensor rate, not forced onto one shared clock. DAM-VLA keeps a separate latent buffer per modality, refreshes each at its sensor rate, and lets the action head read all of them continuously. New high-frequency modalities are added through gated cross-attention, so the pretrained backbone stays intact. Across seven contact-rich real-world tasks, this more than doubles the success rate of the strongest synchronous baseline (95.2% vs. 41%) while running smoothly and reactively at 100 Hz. These results show that letting each sensor update the model at its own rate, instead of forcing everything onto one shared rate, is a practical way to build better manipulation policies. Next, we plan to use force to correct actions while they happen, and to add other fast sensors in the same way.

Acknowledgments

The research presented in this paper was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 448648559.

References

[1] Kevin Black, Manuel Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. *Advances in Neural Information Processing Systems*, 38: 33383–33407, 2026.

[2] Hao Chen, Jiaming Liu, Chenyang Gu, Zhuoyang Liu, Renrui Zhang, Xiaoqi Li, Xiao He, Yandong Guo, Chi-Wing Fu, Shanghang Zhang, et al. Fast-in-slow: A dual-system vla model unifying fast manipulation within slow reasoning. *Advances in Neural Information Processing Systems*, 38:98049–98083, 2026.

[3] Yinpei Dai, Hongze Fu, Jayjun Lee, Yuejiang Liu, Haoran Zhang, Jianing Yang, Chelsea Finn, Nima Fazeli, and Joyce Chai. Robomme: Benchmarking and understanding memory for robotic generalist policies. *arXiv preprint arXiv:2603.04639*, 2026.

[4] Yihuai Gao, Jinyun Liu, Shuang Li, and Shuran Song. Gated memory policy. *arXiv preprint arXiv:2604.18933*, 2026.

[5] Myungkyu Koo, Daewon Choi, Taeyoung Kim, Kyungmin Lee, Changyeon Kim, Younggyo Seo, and Jinwoo Shin. Hamlet: Switch your vision-language-action model into a history-aware policy. *arXiv preprint arXiv:2510.00695*, 2025.

[6] Geonhyup Lee, Yeongjin Lee, Kangmin Kim, Seongju Lee, Sangjun Noh, Seunghyeok Back, and Kyoobin Lee. Manipforce: Force-guided policy learning with frequency-aware representation for contact-rich manipulation. *arXiv preprint arXiv:2509.19047*, 2025.

[7] Yang Li, Hongru Jiang, Junjie Xia, Hongquan Zhang, Jinda Du, Yunsong Zhou, Jia Zeng, Ce Hao, Jieji Ren, Qiaojun Yu, et al. Forcevla2: Unleashing hybrid force-position control with force awareness for contact-rich manipulation. *arXiv preprint arXiv:2603.15169*, 2026.

[8] Yao Li, Peiyuan Tang, Wuyang Zhang, Chengyang Zhu, Yifan Duan, Weikai Shi, Xiaodong Zhang, Zijiang Yang, Jianmin Ji, and Yanyong Zhang. Favla: A force-adaptive fast-slow vla model for contact-rich robotic manipulation. *arXiv preprint arXiv:2602.23648*, 2026.

[9] Min Lin, Xiwen Liang, Bingqian Lin, Liu Jingzhi, Zijian Jiao, Kehan Li, Yuhan Ma, Yuecheng Liu, Shen Zhao, Yuzheng Zhuang, et al. Echovla: Robotic vision-language-action model with synergistic declarative memory for mobile manipulation. *arXiv preprint arXiv:2511.18112*, 2025.

[10] Minghui Lin, Pengxiang Ding, Shu Wang, Zifeng Zhuang, Yang Liu, Xinyang Tong, Wenxuan Song, Shangke Lyu, Siteng Huang, and Donglin Wang. Hif-vla: Hindsight, insight and foresight through motion representation for vision-language-action models. *arXiv preprint arXiv:2512.09928*, 2025.

[11] Yuxiang Lu, Zhe Liu, Xianzhe Fan, Zhenya Yang, Jinghua Hou, Junyi Li, Kaixin Ding, and Hengshuang Zhao. Faster: Rethinking real-time flow vlas. *arXiv preprint arXiv:2603.19199*, 2026.

[12] Chaojun Ni, Cheng Chen, Xiaofeng Wang, Zheng Zhu, Wenzhao Zheng, Boyuan Wang, Tianrun Chen, Guosheng Zhao, Haoyun Li, Zhehao Dong, et al. Swiftvla: Unlocking spatiotemporal dynamics for lightweight vla models at minimal overhead. *arXiv preprint arXiv:2512.00903*, 2025.

[13] Weikang Qiu, Tinglin Huang, and Rex Ying. Efficient long-horizon vision-language-action models via static-dynamic disentanglement. *arXiv preprint arXiv:2602.03983*, 2026.

[14] Kohei Sendai, Maxime Alvarez, Tatsuya Matsushima, Yutaka Matsuo, and Yusuke Iwasawa. Leave no observation behind: Real-time correction for vla action chunks. *arXiv preprint arXiv:2509.23224*, 2025.

[15] Hao Shi, Bin Xie, Yingfei Liu, Lin Sun, Fengrong Liu, Tiancai Wang, Erjin Zhou, Haoqiang Fan, Xiangyu Zhang, and Gao Huang. Memoryvla: Perceptual-cognitive memory in vision-language-action models for robotic manipulation. *arXiv preprint arXiv:2508.19236*, 2025.

[16] Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, et al. Smolvla: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

[17] Ajay Sridhar, Jennifer Pan, Satvik Sharma, and Chelsea Finn. Memer: Scaling up memory for robot control via experience retrieval. *arXiv preprint arXiv:2510.20328*, 2025.

[18] Jiaming Tang, Yufei Sun, Yilong Zhao, Shang Yang, Yujun Lin, Zhuoyang Zhang, James Hou, Yao Lu, Zhijian Liu, and Song Han. Vlash: Real-time vlas via future-state-aware asynchronous inference. *arXiv preprint arXiv:2512.01031*, 2025.

[19] Marcel Torne, Karl Pertsch, Homer Walke, Kyle Vedder, Suraj Nair, Brian Ichter, Allen Z Ren, Haohuan Wang, Jiaming Tang, Kyle Stachowicz, et al. Mem: Multi-scale embodied memory for vision language action models. *arXiv preprint arXiv:2603.03596*, 2026.

[20] Siyu Xu, Yunke Wang, Chenghao Xia, Dihao Zhu, Tao Huang, and Chang Xu. Vla-cache: Towards efficient vision-language-action model via adaptive token caching in robotic manipulation. *arXiv e-prints*, pages arXiv–2502, 2025.

[21] Chen Yang, Yucheng Hu, Yunchao Ma, Yunhuan Yang, Jing Tan, and Haoqiang Fan. Realtime-vla v2: Learning to run vlas fast, smooth, and accurate. *arXiv preprint arXiv:2603.26360*, 2026.

[22] Kaidi Zhang, Heng Zhang, Zhengtong Xu, Zhiyuan Zhang, Md Rakibul Islam Prince, Xiang Li, Xiaojing Han, Yuhao Zhou, Arash Ajoudani, and Yu She. Tacvla:

Contact-aware tactile fusion for robust vision-language-action manipulation. *arXiv preprint arXiv:2603.12665*, 2026.

[23] Zongzheng Zhang, Haobo Xu, Zhuo Yang, Chenghao Yue, Zehao Lin, Huan-ang Gao, Ziwei Wang, and Hao Zhao. Ta-vla: Elucidating the design space of torque-aware vision-language-action models. *arXiv preprint arXiv:2509.07962*, 2025.

[24] Ruiteng Zhao, Wenshuo Wang, Yicheng Ma, Xiaocong Li, Francis EH Tay, Marcelo H Ang Jr, and Haiyue Zhu. Fd-vla: Force-distilled vision-language-action model for contact-rich manipulation. *arXiv preprint arXiv:2602.02142*, 2026.

[25] Yongsheng Zhao, Lei Zhao, Baoping Cheng, Gongxin Yao, Xuanzhang Wen, and Han Gao. Vla-rail: A real-time asynchronous inference linker for vla models and robots. *arXiv preprint arXiv:2512.24673*, 2025.

[26] Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. *arXiv preprint arXiv:2510.10274*, 2025.

[27] Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé III, Andrey Kolobov, Furong Huang, and Jianwei Yang. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. In *International Conference on Learning Representations*, volume 2025, pages 54277–54296, 2025.

[28] Teqiang Zou, Hongliang Zeng, Yuxuan Nong, Yifan Li, Kehui Liu, Haotian Yang, Xinyang Ling, Xin Li, and Lianyang Ma. Asynchronous fast-slow vision-language-action policies for whole-body robotic manipulation. *arXiv preprint arXiv:2512.20188*, 2025.