PRIME W0: Physical Priors from World Models for Embodied Manipulation

Huabin Peng[1,4], **Fagang Jin**[1,†], **Yancong Wei**[1,3], **Tengfei Ma**[1,4,*], **Chuhang Liu**[1,3], **Chao Yu**[2,3], **Liqing Chen**[5]

[1]Lumos SZ Lab Team; [2]Lumos Robotics; [3]THU; [4]CUHK; [5]NTU
[†]Team Leader

Vision-Language-Action (VLA) models excel at instruction following, yet most remain feed-forward: they condition actions only on the current observation and discard both long-horizon temporal context and the hierarchical structure of VLM representations. We present **PRIME W0**, a hierarchical conditional flow-matching framework that closes this gap through three synergistic designs: (1) *per-layer KV injection* from a pretrained VLM into an Adaptive MoE action expert; (2) *dynamics-aware conditioning* via asynchronous world-model subgoal-image prediction and a DINOv2 historical-frame semantic bridge; and (3) *Action Dynamic Manifold (ADM)* flow matching with covariance-structured noise that respects robot kinematic coupling. PRIME W0 is trained in four stages—VLM pre-training, DAT/NTP visuomotor pre-training, DiT flow-matching decoder assembly with WM injection, and end-to-end SFT—and achieves state-of-the-art results on LIBERO and real-robot benchmarks, with systematic ablations validating each architectural component.

Date: June 2, 2026
Project Repo: https://lucasjinreal.github.io/PRIME-W0/

1 Introduction

Vision-Language-Action (VLA) models [6, 15, 5] have emerged as a dominant paradigm for generalist robot policies, casting manipulation as conditional generation over visual and language tokens. By pretraining on large-scale internet data and fine-tuning on robot demonstrations, VLAs achieve impressive breadth across tasks and embodiments. Recent flow- and diffusion-based VLAs further improve action expressiveness by replacing autoregressive decoders with continuous generative processes, enabling multi-modal action distributions and smoother trajectories [5].

The dynamics–semantics gap. Despite these advances, most VLA policies remain *feed-forward*: at each control step they condition action generation only on the current observation and instruction, without an explicit model of how the scene will evolve. This creates two coupled limitations. First, *temporal myopia*: long-horizon tasks with occlusion, contact ambiguity, or multi-phase manipulation require tracking motion trends and anticipating near-future scene layout, yet repeating the full VLM on every historical frame is prohibitively expensive. Second, *semantic compression*: many architectures route action decoding through a single compressed VLM representation, discarding the hierarchical semantic–spatial structure accumulated across intermediate layers that is critical for fine-grained motor control.

Flow-matching action decoders exacerbate both issues. During denoising, early steps benefit from high-level task semantics while late steps require precise perceptual grounding and kinematically consistent trajectories. A policy that conditions only on the present frame and a collapsed VLM summary cannot simultaneously supply forward temporal anchors and layer-wise semantic detail.

Core question. *Can we build a VLA foundation model that (i) anticipates near-future scene evolution, (ii) exploits long-horizon visual history efficiently, and (iii) generates actions on the low-dimensional manifold of physically feasible robot behaviour?*

*Correspondence: matengfei@lumosbot.tech

PRIME W0 (**P**hysical p**RI**ors from world **M**odels for **E**mbodied manipulation) addresses this question with a hierarchical conditional flow-matching framework that couples a pretrained VLM with three synergistic design layers (fig. 1):

1. **Semantic reasoning via per-layer KV injection.** Rather than collapsing the VLM into a single bottleneck vector, we extract Key/Value projections from *all* backbone layers and inject them into the action expert through cross-attention, preserving both high-level intent and low-level spatial grounding.
2. **Dynamics-aware conditioning.** A compact world model $\mathcal{W}$ predicts explicit *subgoal RGB images* from the previous observation; these are encoded as visual tokens and injected *asynchronously* into the VLM, supplying forward scene anchors without blocking inference on future-frame capture. In parallel, a DINOv2-based historical-frame encoder bridges past observations into the language-model embedding space as $\mathbf{W}_{hist}$, enabling long-range temporal context without repeated VLM forwards.
3. **Manifold-constrained action generation.** We replace the autoregressive action head with **Action Dynamic Manifold (ADM)** flow matching over an Adaptive MoE decoder. Covariance noise sampled from demonstration statistics respects kinematic coupling, while temporally gated expert routing specialises behaviour across manipulation phases.

Training proceeds in four stages (section 3.2; fig. 2): VLM pre-training; geometry-aware action tokenisation and NTP visuomotor pre-training on a mixed teleop / UMI / simulation corpus (fig. 3); DiT flow-matching decoder assembly with frozen VLM, WM injection, and Adaptive MoE; and end-to-end SFT under a composite objective that unifies flow matching, MoE load balancing, and auxiliary VLM/WM supervision.

Contributions.

1. We propose a **hierarchical conditional flow-matching framework** that closes the loop between VLM semantic reasoning, asynchronous WM subgoal-image prediction, and manifold-constrained continuous control (section 3).
2. We introduce **asynchronous subgoal-image injection** and a **historical-frame semantic bridge**, enabling dynamics-aware conditioning without increasing end-to-end latency or re-running the full VLM on past frames.
3. We develop **ADM** with covariance-structured flow matching and **Adaptive MoE** action experts, jointly realising phase-aware, kinematically grounded action generation on a learned action manifold.
4. We demonstrate state-of-the-art performance on LIBERO and an in-house real-robot suite, together with systematic ablations confirming the contribution of each architectural component (section 4).

2 Related Work

2.1 Vision-Language-Action Models

Recent vision-language-action (VLA) foundation models have moved beyond task-specific visuomotor policies toward scalable, generalist, and deployment-oriented robot control. $\pi_{0.7}$ [14] proposes a steerable generalist robotic foundation model that uses diverse context conditioning, including language commands, task-performance metadata, and subgoal images, to exploit demonstrations, autonomous rollouts with failures, and non-robot data. LingBot-VLA [35] emphasizes practical deployment by training on large-scale real-world data from multiple dual-arm robot configurations and improving adaptation cost, training throughput, open code, and benchmark construction. JoyAI-RA [39] studies multi-source and multi-level pretraining by integrating web data, egocentric human manipulation videos, simulation-generated trajectories, and real-robot data, with explicit action-space unification to bridge embodiment gaps. PokeVLA [41] takes a complementary lightweight direction, pretraining a compact VLM with spatial grounding, affordance, and embodied reasoning data, and then injecting manipulation-relevant representations into the action space through goal-aware semantics learning, geometry alignment, and an action expert.

Another emerging trend is to make VLA models more suitable for long-horizon, real-time, and reasoning-intensive robotic autonomy. Xiaomi-Robotics-0 [8] focuses on fast and smooth real-time execution by addressing inference latency through asynchronous execution and temporal alignment of consecutive predicted

action chunks. Habilis-β [29] pushes VLA deployment toward fast-motion and long-lasting on-device control, introducing continuous-run evaluation metrics such as Tasks per Hour and Mean Time Between Intervention to measure both productivity and reliability. HALO [31] incorporates embodied multimodal chain-of-thought reasoning through textual task reasoning, visual subgoal prediction, and action prediction. These works indicate that recent VLA research is shifting from direct observation-to-action mapping toward foundation models with richer conditioning, stronger cross-embodiment generalization, spatial and affordance-aware representations, deployment-aware execution, and explicit reasoning or visual foresight. Nevertheless, most existing VLA systems still encode future physical consequences implicitly or through compact subgoal representations, whereas our work is motivated by using an explicit video world model to generate task-conditioned future observations for downstream VLM-based action generation.

2.2 World Models and Video Prediction

Recent progress in video generation and self-supervised video representation learning has made world models increasingly attractive for embodied intelligence. Unlike conventional VLA policies that map current observations directly to actions, world models aim to capture how scenes evolve over time, thereby providing predictive priors about object motion, contact dynamics, and task progress. V-JEPA 2 [3] shows that self-supervised pretraining on internet-scale video can produce representations useful for understanding, prediction, and robotic planning, and further adapts the model into an action-conditioned world model for zero-shot image-goal planning. LingBot-World [34] pushes open-source video-based world simulation toward high-fidelity, long-horizon, and real-time interactive generation. DreamDojo [11] further demonstrates the potential of large-scale egocentric human videos for robot world modeling by learning from 44k hours of human interaction videos and introducing continuous latent actions as unified proxy actions. After post-training on target robot data, DreamDojo supports controllable simulation of contact-rich manipulation and enables applications such as live teleoperation, policy evaluation, and model-based planning.

A closely related direction is World-Action Models (WAMs), which jointly model future visual dynamics and robot actions. DreamZero [37] argues that world-action models can act as zero-shot policies by learning physical dynamics through joint video-action prediction, achieving strong generalization to unseen tasks, environments, and embodiments. Motus [4] proposes a unified latent action world model with a Mixture-of-Transformers architecture, integrating understanding, video generation, and action experts within a flexible generative framework. MotuBrain [33] extends this line with unified multiview modeling, shared cross-embodiment action representations, and efficient deployment recipes for long-horizon control. Genie Envisioner [20] presents a unified world foundation platform that combines instruction-conditioned video generation, flow-matching action decoding, neural simulation, and benchmarking. Cosmos Policy [17] fine-tunes a pretrained video model into a robot policy by representing robot actions, future states, and values as latent frames inside the video diffusion process, enabling both visuomotor control and test-time planning. These works suggest that video prediction can serve not only as an auxiliary simulation tool, but also as a scalable substrate for learning action-conditioned physical representations.

Recent studies further examine whether explicit future video generation is necessary during deployment. Fast-WAM [38] separates the benefit of video co-training from test-time future imagination and shows that competitive performance can be achieved without generating future frames at inference, suggesting that video modeling may primarily improve world representations during training. GigaWorld-Policy [36] similarly adopts an action-centered WAM design, using video generation as a training-time constraint while making future-video generation optional at inference for faster deployment. Being-H0.7 [22] moves future-aware reasoning into a latent space by aligning deployable latent queries with future-observation-informed posterior representations, thereby retaining predictive benefits without pixel-space rollout. DiT4DiT [23] takes another route by coupling a video Diffusion Transformer with an action Diffusion Transformer and using intermediate denoising features, rather than fully reconstructed future frames, as temporally grounded conditions for action prediction. In contrast to methods that remove or bypass explicit future generation at test time, our work explores a complementary direction: using a video world model to produce task-conditioned future observations and feeding them back into a VLM-action pipeline, so that action generation can be guided by explicit visual imagination of future task evolution.

2.3 World Models for VLA Policy Learning

Recent work has begun to integrate world models more tightly with vision-language-action (VLA) policies, rather than treating them as separate simulators or standalone video predictors. WorldVLA [10] proposes an autoregressive action world model that unifies action generation and future-image prediction in a single framework, where the world model learns environment dynamics to improve action generation and the action model in turn supports visual understanding for world modeling. RynnVLA-002 [9] further develops this unified VLA–world-model formulation by jointly learning future image states and subsequent actions, showing that the integrated world model can substantially improve real-world VLA success rates. GigaBrain-0 [32] takes a data-centric route, using world models to generate diverse training data, including video generation, real-to-real transfer, human-transfer, view-transfer, and sim-to-real-transfer data, thereby reducing the dependence on expensive real-robot data while improving generalization across appearance, object placement, and camera-view variations. These works indicate that world models can enhance VLA policies not only by predicting future observations, but also by providing scalable data, forward scene predictions, and richer supervision for action learning.

Another line of work uses world models as interactive environments or reward models for post-training VLA policies. NORA-1.5 [13] augments a VLA backbone with a flow-matching action expert and further improves reliability through preference optimization, where rewards are derived from an action-conditioned world model and action-quality heuristics. VLA-RFT [18] performs reinforcement fine-tuning with verified rewards in a world simulator, allowing VLA policies to obtain dense trajectory-level supervision without costly real-world interaction. World-VLA-Loop [21] extends this idea into a closed-loop framework that jointly refines the video world model and the VLA policy: policy failures are fed back to improve action-outcome alignment in the world model, which then provides better virtual rollouts for subsequent policy optimization. VLA-MBPO [40] further studies practical model-based reinforcement learning for VLA models by adapting unified multimodal models for data-efficient world modeling, enforcing multi-view consistency through interleaved view decoding, and using chunk-level branched rollouts to mitigate compounding errors. Together, these methods show that world models are becoming a key infrastructure for VLA post-training, preference learning, and sample-efficient reinforcement fine-tuning.

While these approaches demonstrate the benefit of world models for VLA policy learning, they mainly use world models for joint VLA–world-model training, synthetic data generation, reward construction, or virtual rollouts during post-training. Our work explores a complementary direction: using a video world model as an explicit future-imagination module inside the VLA inference pipeline. Given the current observation and task-level context, the world model predicts task-conditioned future observations, which are then encoded by a VLM and consumed by an action expert for downstream continuous action generation. In this way, future visual prediction is not only used as an offline training signal or simulator, but also becomes an explicit intermediate representation for action reasoning and execution.

3 Method

3.1 Overall Architecture

PRIME W0 is a hierarchical conditional flow-matching framework for general robotic manipulation that couples the semantic reasoning capacity of a large-scale pre-trained vision–language model (VLM) with structured generation on a continuous action manifold (fig. 1).

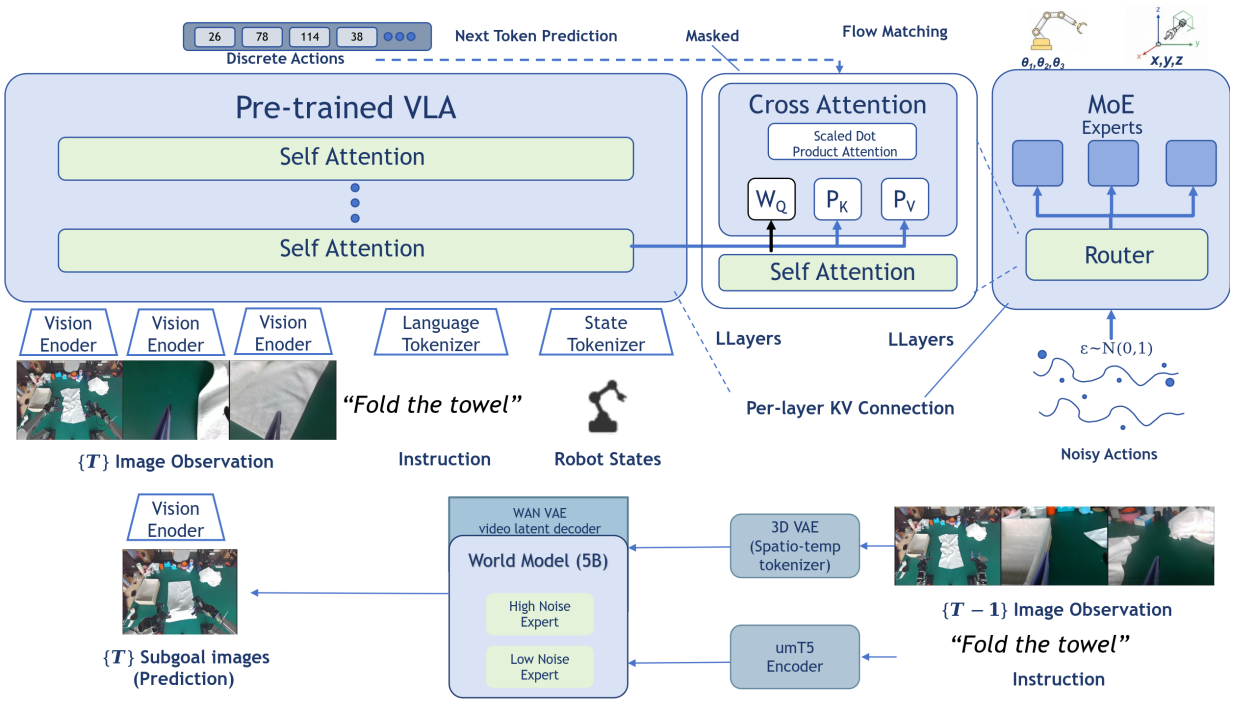


Figure 1: PRIME W0 architecture.

At each control cycle t, the system receives:

- **Multi-view visual observations:** $\mathbf{o}_t = \{\mathbf{I}_t^{(c)}\}_{c=1}^{C}$, where $\mathbf{I}_t^{(c)}$ is the RGB image from camera c;
- **Historical observation context:** $\{\mathbf{o}_{t-\tau}, \ldots, \mathbf{o}_{t-1}\}$, encoded into semantic tokens by a dedicated historical-frame encoder (section 3.6);
- **Proprioceptive state:** $\mathbf{q}_t$ (joint angles, end-effector pose, gripper aperture, etc.);
- **Natural-language instruction:** ℓ, tokenised as $\mathbf{W}_\ell = [\omega_1, \ldots, \omega_{|\ell|}]$.

Multimodal encoding and VLM backbone. Images are encoded by a vision encoder E_ϕ (e.g. SigLIP or a ViT-based backbone) into multi-scale patch tokens. These are concatenated with proprioceptive tokens $\mathbf{s}_t = \mathrm{MLP}_s(\mathbf{q}_t)$ and language tokens $\mathbf{W}_\ell$, and fed into the pre-trained VLM backbone $\mathcal{F}_{\mathrm{VLM}}$. For layer ℓ of the backbone, we denote the hidden states by $\mathbf{H}_t^{(\ell)}$ and explicitly extract the internal **Key/Value projections**:

$$\mathbf{K}_t^{(\ell)} = W_K^{(\ell)} \mathbf{H}_t^{(\ell)}, \qquad \mathbf{V}_t^{(\ell)} = W_V^{(\ell)} \mathbf{H}_t^{(\ell)}, \qquad \ell = 1, \ldots, L. \tag{1}$$

These *per-layer KV representations* are subsequently injected into the action expert module via cross-attention (section 3.7), enabling the action head to read semantic–spatial joint representations from all VLM layers without back-propagating gradients through the full VLM parameter space.

Closed-loop inference. Overall inference forms a *closed loop with asynchronous world-model prediction*:

$$\boxed{\mathbf{o}_{t-1} \xrightarrow{\text{World Model}} \widehat{\mathbf{o}}_t^{(wm)} \xrightarrow{\text{async inject}} \mathcal{F}_{\mathrm{VLM}}(\mathbf{o}_t,\, \ell,\, \mathbf{W}_{hist}) \xrightarrow{\text{Adaptive MoE + ADM}} \mathbf{A}_t} \tag{2}$$

where $\widehat{\mathbf{o}}_t^{(wm)}$ is a *predicted subgoal image* at timestep t produced by the world model from the previous observation (section 3.5), encoded into visual tokens and injected into the VLM feature sequence to supply forward temporal priors to the action expert. Historical frames are bridged into the same conditioning stream via $\mathbf{W}_{hist}$ (section 3.6). The final action head outputs an action chunk via **Action Dynamic Manifold (ADM)** (sections 3.8 and 3.9):

$$\mathbf{A}_t = [\mathbf{a}_t,\ \mathbf{a}_{t+1},\ \ldots,\ \mathbf{a}_{t+H-1}] \in \mathbb{R}^{H \times d_a}. \tag{3}$$

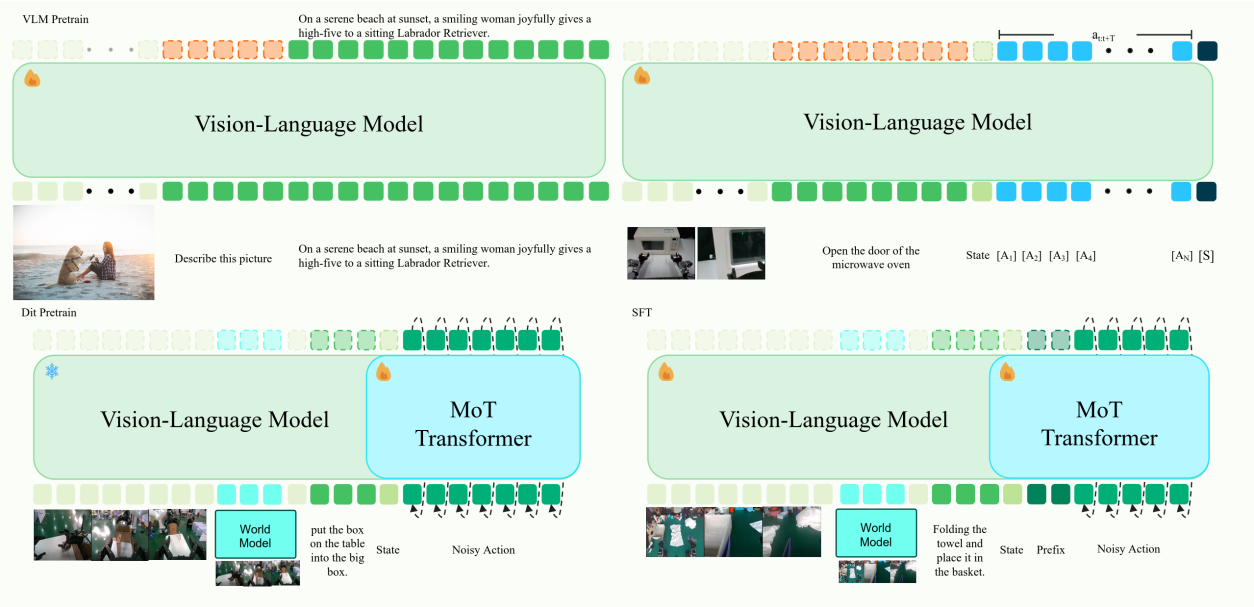


Figure 2: Four-stage training pipeline. (1) VLM Pretrain: the vision–language backbone is trained on general multimodal supervision (e.g. captioning / VQA). **(2) NTP / task-specific pre-training:** with DAT action tokens, the VLM learns visuomotor next-token prediction over instruction–observation–action sequences. **(3) DiT Pretrain:** the VLM is frozen while the MoT / Adaptive MoE flow-matching action expert is trained with world-model conditioning to denoise continuous actions (and associated state tokens). **(4) SFT:** both the VLM and the action expert are jointly fine-tuned end-to-end, optionally with action-prefix tokens that shorten the denoising trajectory at inference. Flame / snowflake icons indicate trainable vs. frozen modules.

Together, these components form a *semantic-grounded* → *dynamics-aware* → *manifold-constrained* embodied action generation pipeline; the four-stage training recipe that realises this architecture is described in section 3.2.

3.2 Four-Stage Training Pipeline

PRIME W0 is trained in four progressive stages that build from general vision–language understanding to a fully integrated continuous control policy (fig. 2).

Stage 1: VLM Pre-training. We start from (or continue) large-scale vision–language pre-training of the backbone $\mathcal{F}_{\text{VLM}}$ on multimodal corpora (captioning, visual question answering, and related alignment objectives). This stage establishes semantic grounding that subsequent robot stages must preserve rather than overwrite.

Stage 2: Action Tokenization & NTP Pre-training. We train a geometry-aware discrete action tokenizer $\mathcal{T}$ (DAT) on the robot demonstration corpus (section 3.3), then pre-train the VLM with a next-token prediction (NTP) objective over unified visual–language–action token sequences (section 3.4). Mixed VL–robot co-training maintains language competence while endowing the model with broad visuomotor understanding across embodiments and tasks.

Stage 3: DiT / Flow-Matching Pre-training with WM Injection. We freeze the VLM backbone and replace the autoregressive action head with a continuous **flow-matching action decoder** (DiT-style MoT / Adaptive MoE experts; sections 3.7 to 3.9). The expert receives per-layer VLM KV representations through cross-attention (eq. (1)). A **historical-frame encoder** (section 3.6) supplies long-horizon visual context, and a compact **world model** $\mathcal{W}$ is integrated via **asynchronous subgoal-image injection** (section 3.5; architecture in appendix A): predicted subgoal frames from $\mathbf{o}_{t-1}$ are encoded as visual tokens and fused into the VLM sequence at timestep t without blocking on future-frame capture. Stage 3 optimises the flow-matching loss $\mathcal{L}_{FM}$ with MoE load-balancing $\mathcal{L}_{\text{bal}}$ while denoising continuous actions (and associated state tokens) under world-model conditioning.

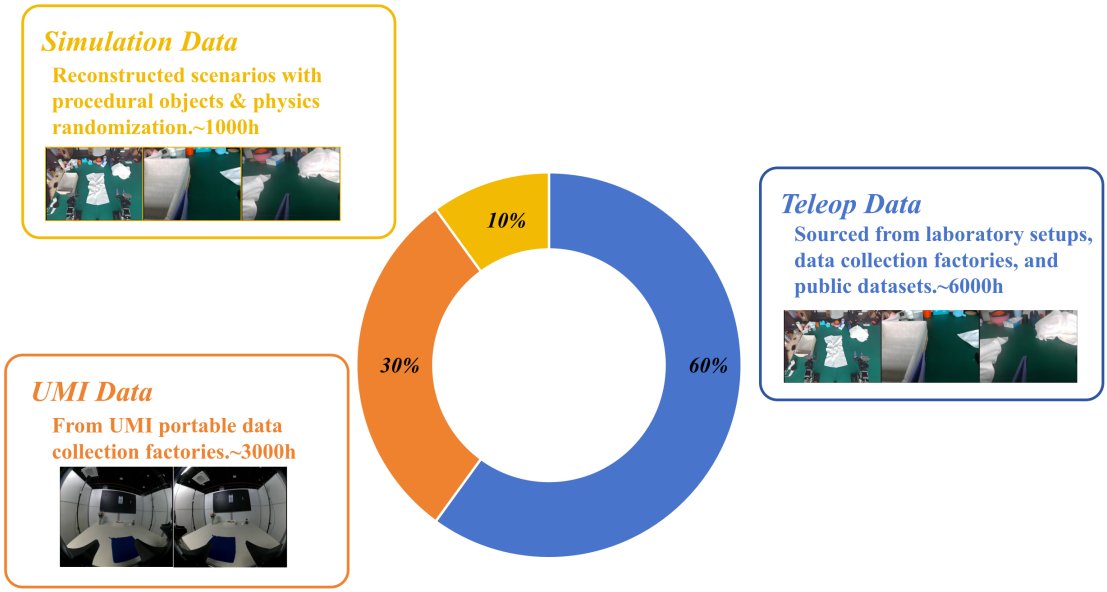


Figure 3: Training data mixture. Robot demonstration data used by PRIME W0 totals approximately **10,000 hours**, comprising **Teleop** (60%, ∼6,000 h; laboratory setups, data-collection factories, and public datasets), **UMI** (30%, ∼3,000 h; portable UMI fisheye egocentric collection), and **Simulation** (10%, ∼1,000 h; reconstructed scenes with procedural objects and physics randomisation).

Stage 4: Supervised Fine-Tuning (SFT) / End-to-End Co-training. We jointly fine-tune the VLM, world model, and action expert on labelled robot demonstrations under the full composite objective (eq. (32)), where $\mathcal{L}_{\text{aux}}$ retains VLM language ability via continued NTP supervision and enforces WM subgoal-image consistency. Optional *action-prefix* tokens initialise denoising from a low-noise state of already-executed actions, shortening inference. This stage closes the loop between semantic reasoning, forward dynamics prediction, and manifold-constrained action generation, yielding a deployable real-time policy.

3.3 Training Data Mixture

Robot demonstration data for Stages 2–4 form a heterogeneous mixture of approximately **10,000 hours** spanning laboratory teleoperation, portable egocentric collection, and simulation (fig. 3).

- **Teleop data (60%, ∼6,000 h).** Demonstrations from laboratory setups, industrial data-collection factories, and curated public robot datasets. These typically provide multi-view RGB (including third-person cameras) with proprioception and language annotations, and form the primary source of high-quality expert trajectories.
- **UMI data (30%, ∼3,000 h).** Egocentric trajectories collected with portable UMI-style handheld interfaces and fisheye wrist cameras. Relative to teleop, UMI increases spatial/scene diversity at the cost of missing fixed third-person views; during training we randomly mask third-person inputs so that the policy learns under both multi-view and ego-only observation regimes.
- **Simulation data (10%, ∼1,000 h).** Procedurally reconstructed scenarios with randomised objects and physics parameters, used to densify coverage of contact-rich and long-horizon behaviours that are expensive to collect physically.

All sources are mapped to a shared Cartesian / SE(3) end-effector action space before DAT tokenisation (section 3.4), enabling joint training without embodiment-specific heads.

3.4 Action Tokenization and NTP Pre-training

This subsection details Stage 2 of section 3.2: encoding continuous robot actions as discrete tokens and pre-training the VLM backbone for visuomotor competence.

Bimanual Cartesian action representation. A standard VLM backbone operates over discrete tokens, yet robot actions live in a structured continuous space. For our bimanual platform, each action is a 20-dimensional Cartesian end-effector command:

$$\mathbf{a}_t = [\underbrace{\mathbf{p}^L, \mathbf{R}^L, g^L}_{\text{10 left arm}}, \underbrace{\mathbf{p}^R, \mathbf{R}^R, g^R}_{\text{10 right arm}}] \in \mathbb{R}^{20}, \tag{4}$$

where $\mathbf{p} \in \mathbb{R}^3$ is the end-effector position, $\mathbf{R} \in \mathbb{R}^6$ is the rotation in continuous 6D representation (first two columns of the rotation matrix), and $g \in \mathbb{R}$ is the gripper aperture. This Cartesian parameterisation is used *exclusively* within the tokenizer and NTP pre-training stages.

Naively discretising each of the 20 dimensions independently (per-dimension binning) ignores the geometric coupling between position, orientation, and gripper—leading to reconstructed actions that violate physical consistency (e.g. a closed gripper drifting away from the object, or left-right arms colliding). FAST [25] improves upon binning by applying DCT + BPE for high compression, but its variable-length output complicates batching during NTP training, and it does not incorporate task-aware geometric constraints. To address these limitations we propose **DAT** (**D**iscrete **A**ction **T**okeniser), a geometry-aware action codec that produces *fixed-length* discrete tokens via Finite Scalar Quantisation (FSQ), trained with manipulation-aligned auxiliary losses that explicitly preserve gripper phase, bimanual coordination, and rotational fidelity.

DAT architecture. DAT follows a *register-bottleneck encoder* $\rightarrow$ *FSQ quantiser* $\rightarrow$ *cross-attention decoder* pipeline, using a small set of learnable register tokens as an information bottleneck that compresses variable-length action sequences into a fixed-length discrete representation. Given an action chunk $\mathbf{a}_{t:t+H} \in \mathbb{R}^{H\times 20}$ of horizon H:

1. *Register encoder.* The action sequence is first mapped to a d_{enc}-dimensional embedding space via linear projection and positional encoding, then concatenated with T learnable register tokens and fed into an N_{enc}-layer Transformer. We use an *asymmetric attention mask*: action tokens attend bidirectionally to each other but cannot see the registers, while register tokens attend to all tokens (actions and other registers). This allows the registers to actively aggregate sequence-level information without disturbing the original action representations. Only the register outputs are retained and projected to a k-dimensional latent space: $\mathbf{e} = \text{Enc}_\psi(\mathbf{a}_{t:t+H}) \in \mathbb{R}^{T\times k}$. The bottleneck forces the T registers to serve as a compact sufficient statistic of the entire action chunk.
2. *FSQ quantisation.* Each scalar of the k-dimensional latent vector is independently bounded via tanh and then rounded to the nearest point on a finite grid of L levels (Finite Scalar Quantisation, FSQ), yielding a fixed-length discrete token sequence $\mathbf{c} = \text{FSQ}(\mathbf{e}) \in \{0, \dots, L-1\}^{T\times k}$ and its continuous straight-through approximation $\hat{\mathbf{e}} \in \mathbb{R}^{T\times k}$. Gradients pass through via the straight-through estimator, making the entire pipeline fully differentiable without codebook collapse. Each register spans a discrete space of L^k; the total codebook capacity is L^{Tk}, which in practice provides ample representational power even with modest k and T (e.g. L=8, k=4, T=16 yields $8^{64} \approx 6 \times 10^{57}$ distinct encodings).
3. *Cross-attention decoder.* The decoder uses H learnable positional queries as the target sequence and attends to the quantised register embeddings $\hat{\mathbf{e}}$ as memory through an N_{dec}-layer Transformer cross-attention decoder to reconstruct the action chunk: $\hat{\mathbf{a}}_{t:t+H} = \text{Dec}_\psi(\hat{\mathbf{e}})$. This design decomposes decoding into two orthogonal sub-problems— retrieving information from registers and generating actions in temporal order—allowing the model to flexibly route different register information to different time steps.

Geometry-aware training objective. A key insight is that a single scalar reconstruction metric cannot capture the heterogeneous structure of a 20-dimensional bimanual action: position, orientation, and gripper each obey different geometries. DAT therefore adopts a *structure-aware* reconstruction loss that applies the appropriate distance metric to each sub-space, combined with a frequency-domain objective and two

manipulation-specific auxiliary losses:

$$\mathcal{L}_{\text{DAT}} = \mathcal{L}_{\text{recon}} + \lambda_{\text{dct}}\,\mathcal{L}_{\text{DCT}} + \lambda_{\text{grip}}\,\mathcal{L}_{\text{grip}} + \lambda_{\text{lr}}\,\mathcal{L}_{\text{lr}}, \tag{5}$$

where each term addresses a distinct failure mode:

(i) Structure-aware reconstruction loss. Rather than applying a uniform L_2 loss across all 20 dimensions, we decompose the action into its geometric sub-spaces and use the natural metric for each:

$$\begin{aligned}\mathcal{L}_{\text{recon}} = &\underbrace{\text{SmoothL1}(\mathbf{p},\,\hat{\mathbf{p}})}_{\text{position}} + \underbrace{\text{SmoothL1}(g,\,\hat{g})}_{\text{gripper}} \\ &+ \underbrace{\arccos\!\Big(\tfrac{\text{tr}(\hat{\mathbf{R}}_{\text{mat}}^{\top}\mathbf{R}_{\text{mat}})-1}{2}\Big)}_{\text{rotation (geodesic on SO(3))}},\end{aligned} \tag{6}$$

summed over both arms, where $\mathbf{R}_{\text{mat}} \in \text{SO}(3)$ is the full rotation matrix recovered from the 6D representation $\mathbf{R}$. Smooth L_1 provides robustness to outlier targets in position and gripper channels, while the geodesic distance on SO(3) avoids the well-known artefacts of applying Euclidean metrics in rotation space (e.g. discontinuities near representation boundaries).

(ii) DCT loss. We apply an auxiliary loss in the frequency domain to encourage the reconstructed trajectory to preserve the spectral structure of the original action chunk:

$$\mathcal{L}_{\text{DCT}} = \big\|\text{DCT}_{\text{II}}(\mathbf{a}_{t:t+H}) - \text{DCT}_{\text{II}}(\hat{\mathbf{a}}_{t:t+H})\big\|_F^2, \tag{7}$$

where DCT_{II} is the orthonormalised type-II discrete cosine transform applied along the temporal axis of the action chunk. Because of the energy compaction property of the DCT, low-frequency coefficients naturally carry larger signal magnitudes, so this loss implicitly up-weights low-frequency trend components. By supervising in the DCT domain rather than only in the time domain, the tokenizer better captures both smooth trajectory trends and transient contact events.

(iii) Gripper direction loss. In bimanual manipulation, knowing *whether* a gripper is closing or opening is often more decisive than its instantaneous aperture width: a correct width at the wrong phase (e.g. opening when it should be closing) leads to catastrophic grasp failure, yet this error can be almost invisible in the overall MSE because the scalar width remains numerically close. We observed empirically that without an explicit directional signal, the tokenizer frequently produces "chattering" reconstructions—the gripper oscillates between open and close at a single position instead of executing a clean grasp or release. DAT therefore supervises the gripper *direction* as a binary classification:

$$\mathcal{L}_{\text{grip}} = \text{BCE}\big(\text{dir}(\mathbf{a}),\ \text{dir}(\hat{\mathbf{a}})\big), \quad \text{dir}(g_t) = \mathbb{1}[g_t - g_{t-1} > 0], \tag{8}$$

where $\text{dir}(\cdot)$ encodes the sign of the gripper aperture change between consecutive steps. For ground-truth actions dir is a hard indicator; for reconstructions the consecutive gripper increment $\hat{g}_t - \hat{g}_{t-1}$ naturally serves as a direction logit—positive for opening, negative for closing—and is passed through sigmoid to yield an open/close probability, making the loss fully differentiable without requiring an extra classification head. The loss is applied independently to each arm.

(iv) Left-right end-effector distance loss. Bimanual tasks (e.g. cooperative lifting, two-handed folding) require tight spatial coordination between the two arms. A tokenizer that reconstructs each arm independently may produce trajectories where the arms drift apart or collide. $\mathcal{L}_{\text{lr}}$ explicitly constrains the inter-arm distance to match the ground truth, providing a direct inductive bias for coordination:

$$\mathcal{L}_{\text{lr}} = \text{SmoothL1}\big(\|\mathbf{p}^L - \mathbf{p}^R\|_2,\ \|\hat{\mathbf{p}}^L - \hat{\mathbf{p}}^R\|_2\big). \tag{9}$$

NTP pre-training. With the frozen DAT codec, every robot demonstration $(\mathbf{o}_{1:T_{\text{ep}}},\ \ell,\ \mathbf{a}_{1:T_{\text{ep}}})$ is converted into a unified token sequence:

$$\underbrace{v_1, \ldots, v_P}_{\text{visual tokens}},\ \underbrace{w_1, \ldots, w_Q}_{\text{language tokens}},\ \underbrace{c_1, \ldots, c_R}_{\text{action tokens (from DAT)}}, \tag{10}$$

and the VLM backbone is pre-trained with the standard causal language-modelling objective over the action portion:

$$\mathcal{L}_{\text{NTP}} = -\sum_{i=1}^{R} \log p_\theta\big(c_i \mid v_{1:P},\, w_{1:Q},\, c_{<i}\big). \tag{11}$$

Because DAT produces a *fixed-length* token sequence per chunk (unlike the variable-length output of BPE-based methods such as FAST), the NTP training loop requires no padding or length-prediction head, simplifying both implementation and batching. Crucially, we *co-train* the VLM on a mixture of robot demonstrations and vision-language QA data (visual question answering, image captioning, etc.) to retain the backbone's original language-grounding capability and prevent catastrophic forgetting of its pre-trained representations.

3.5 World Model and Asynchronous Subgoal-Image Injection

Introduced in Stage 3, the world model supplies forward scene priors to the VLM by predicting explicit subgoal images and injecting them asynchronously as visual tokens (eq. (2)); see appendix A for architectural details.

Motivation. A purely feed-forward VLA policy conditions only on the *current frame*, lacking an internal model of environmental *dynamics*. This deficiency degrades phase consistency in long-horizon settings with occlusion or ambiguous contact. We therefore introduce a lightweight **World Model (WM)** that predicts the *subgoal RGB observation* the scene should evolve toward at the next timestep. Unlike latent-only world models, $\mathcal{W}$ operates directly in image space: its output is decoded into a full visual frame, re-encoded by the shared vision encoder, and injected into the VLM as standard image tokens—preserving interpretability and alignment with the VLM's native visual interface.

Subgoal-image prediction. The world model takes the **previous observation** $\mathbf{o}_{t-1}$ (and optionally the last executed action $\mathbf{a}_{t-1}$, a short proprioceptive history, and the language instruction ℓ) as input and predicts the subgoal image at the current timestep:

$$\widehat{\mathbf{o}}_t^{(wm)} = \Phi_{wm}(\,\mathbf{o}_{t-1},\, \mathbf{a}_{t-1:\text{pre}},\, \ell\,), \qquad \widehat{\mathbf{o}}_t^{(wm)} \in \mathbb{R}^{H\times W\times 3}, \tag{12}$$

where Φ_{wm} is a conditional video/image generator (in practice, a spatio-temporal VAE decoder paired with a compact diffusion or flow-based denoiser) trained to produce visually coherent subgoal frames rather than abstract latent codes. The predicted image is subsequently tokenised by the same vision encoder E_ϕ used for live observations: $\mathbf{t}_t^{(wm)} = E_\phi(\widehat{\mathbf{o}}_t^{(wm)})$.

Asynchronous injection. The key architectural innovation is that $\widehat{\mathbf{o}}_t^{(wm)}$ depends only on $\mathbf{o}_{t-1}$, which is already available from the previous control cycle. WM inference can therefore be launched *in parallel* with capture and preprocessing of the current frame $\mathbf{o}_t$:

$$\underbrace{\text{WM}(\mathbf{o}_{t-1}) \text{ in background}}_{\text{early start}} \;\|\; \underbrace{\text{capture } \mathbf{o}_t \;+\; \text{VLM prefill}}_{\text{foreground}}, \qquad \text{inject } E_\phi(\widehat{\mathbf{o}}_t^{(wm)}) \text{ as soon as ready.} \tag{13}$$

The resulting subgoal-image tokens are concatenated with current-frame visual tokens, historical tokens, proprioceptive tokens, and language tokens before entering the LM stack:

$$\mathbf{T}_t^{\text{full}} = \big[\; \underbrace{\mathbf{t}_t^{(img)}}_{\text{current-frame visual tokens}},\; \underbrace{\mathbf{W}_{hist}}_{\text{historical tokens}},\; \underbrace{\mathbf{t}_t^{(prop)}}_{\text{proprio / system state}},\; \underbrace{\mathbf{t}_t^{(wm)} = E_\phi(\widehat{\mathbf{o}}_t^{(wm)})}_{\text{WM subgoal-image tokens}} \;\big]. \tag{14}$$

In LM self-attention, $\mathbf{t}_t^{(wm)}$ and $\mathbf{t}_t^{(img)}$ are **mutually visible** (optionally constrained by a block-diagonal or segmented causal mask). Semantic reasoning is thus jointly conditioned on the *observed present* and the *model-predicted subgoal scene*: the former ensures grounded accuracy, while the latter provides a **forward temporal anchor** that stabilises phase estimation. Because WM generation does not wait for any deep features of $\mathbf{o}_t$, the full VLA inference chain is not synchronously blocked on future-frame capture; end-to-end latency is approximately $\max(T_{wm}, T_{vision})$ rather than $T_{wm} + T_{vision}$.

3.6 Historical Frame Encoder and Semantic Bridging

To endow the policy with temporal dynamics awareness without re-running the full VLM on every past frame, we introduce a dedicated **historical-frame encoder**. Rather than reusing shallow features from the current-frame pathway, this module transforms cached past observations into semantic representations compatible with the flow-matching action decoder.

DINOv2 feature extraction. For the buffered historical observations $\{\mathbf{o}_{t-\tau}, \ldots, \mathbf{o}_{t-1}\}$, we first extract robust visual descriptors with a frozen pre-trained **DINOv2** encoder E_{dino}:

$$\mathbf{F}_{hist} = E_{\text{dino}}(\{\mathbf{o}_{t-\tau}, \ldots, \mathbf{o}_{t-1}\}), \tag{15}$$

where $\mathbf{F}_{hist}$ collects patch-level features for each historical frame. DINOv2 provides spatially consistent, viewpoint-robust representations that capture object geometry and scene layout across the recent past.

Vision–language semantic bridging. Because the DINOv2 feature space differs from the VLM's language-model embedding space, we apply a lightweight projection to map historical visual features into token representations that the action decoder can consume alongside instruction embeddings:

$$\mathbf{W}_{hist} = \text{LayerNorm}(\mathbf{F}_{hist}\, W_{proj} + \mathbf{b}_{proj}), \tag{16}$$

yielding **historical language tokens** $\mathbf{W}_{hist}$ that reside in the LM embedding space. This vision–language bridging mechanism allows the action-generation module to exploit long-range temporal dependencies encoded in past frames while avoiding the computational cost of repeatedly forwarding historical images through the full VLM backbone.

Integration with the action decoder. The historical tokens $\mathbf{W}_{hist}$ are concatenated with current image tokens, instruction tokens, and WM subgoal-image tokens (eq. (14)) and passed as conditioning context to the flow-matching action expert. Within the Adaptive MoE stack, the aggregated history vector $\bar{\mathbf{h}}_{t-\tau:t-1}$ used for temporal gating (eq. (17)) can additionally be derived from $\mathbf{F}_{hist}$, linking phase-aware expert routing to the same historical evidence.

3.7 Hierarchical Temporal-Gated Adaptive MoE Action Experts

The Adaptive MoE module forms the action-generation backbone introduced in Stage 3, specialising behaviour across manipulation phases while reading per-layer VLM representations via cross-attention; decoder hyperparameters are summarised in appendix C.

Design motivation. In conventional VLA architectures, action decoding typically shares the same FFN and attention parameters as vision–language reasoning, which limits *dynamic behavioural specialisation* across manipulation phases (approach, contact, fine adjustment, transfer). We therefore introduce an **Adaptive Mixture-of-Experts (Adaptive MoE)** module in the action generation pathway, guided by two principles:

1. **Explicit decoupling of routing and weighting:** expert *selection* (a discrete routing decision) and soft combination weights (continuous weighting) are predicted by *two independent sub-networks*, avoiding a single gating vector that must simultaneously decide *which* experts to activate and *how much* to weight each.
2. **Per-layer independence with temporal gating:** each Transformer layer ℓ maintains a dedicated gate $G^{(\ell)}$ whose input includes not only the current token representation but also a **historical frame aggregation vector** $\bar{\mathbf{h}}_{t-\tau:t-1}$, endowing routing with phase sensitivity along the temporal dimension and mitigating expert collapse / mode starvation.

Historical context aggregation. We define a (parametric or non-parametric) aggregation operator over historical latent states:

$$\bar{\mathbf{h}}_{t-\tau:t-1} = \text{Agg}\Big(\{\mathbf{h}_{t-\tau}, \ldots, \mathbf{h}_{t-1}\}\Big) \in \mathbb{R}^{d_h}, \tag{17}$$

where $\text{Agg}(\cdot)$ may be mean pooling, a lightweight GRU or 1-D convolution, or cross-frame attention. $\bar{\mathbf{h}}_{t-\tau:t-1}$ encodes recent motion trends and environmental dynamics, serving as a **temporal bias term** for gating.

Gating input construction. For token i at layer ℓ of the action expert (an action token or a proprioceptive token) with representation $\mathbf{x}_i^{(\ell)}$, we construct the gating input:

$$\mathbf{g}_i^{(\ell)} = \mathrm{stopgrad}\Big(\mathrm{MLP}_g^{(\ell)}\Big([\,\mathbf{x}_i^{(\ell)} \,\|\, \bar{\mathbf{h}}_{t-\tau:t-1}\,]\Big)\Big), \tag{18}$$

where $\|$ denotes feature-wise concatenation and $\mathrm{stopgrad}(\cdot)$ optionally blocks gradient flow into the VLM backbone depending on the training stage (partial unfreeze or frozen VLM).

Decoupled expert selection and weight prediction. Let the layer host N experts $\{E_k(\cdot)\}_{k=1}^N$, each implemented as a lightweight FFN or local Transformer block. The **expert selection network** $R_{\mathrm{sel}}^{(\ell)}$ outputs unnormalised routing logits; we take the Top-K indices to form the active set:

$$\mathcal{S}_i^{(\ell)} = \mathrm{TopK}_K\Big(\,R_{\mathrm{sel}}^{(\ell)}(\mathbf{g}_i^{(\ell)})\,\Big) \subset \{1,\dots,N\}. \tag{19}$$

The **weight prediction network** $R_{\mathrm{w}}^{(\ell)}$ outputs continuous coefficients only for the selected indices, followed by a temperature-scaled Softmax:

$$\alpha_{i,k}^{(\ell)} = \begin{cases} \dfrac{\exp\big(R_{\mathrm{w},k}^{(\ell)}(\mathbf{g}_i^{(\ell)})/\tau\big)}{\sum_{j\in\mathcal{S}_i^{(\ell)}} \exp\big(R_{\mathrm{w},j}^{(\ell)}(\mathbf{g}_i^{(\ell)})/\tau\big)}, & k \in \mathcal{S}_i^{(\ell)} \\ 0, & k \notin \mathcal{S}_i^{(\ell)} \end{cases} \tag{20}$$

MoE output and load-balancing regularisation. The layer output combines a sparse expert mixture with a residual connection:

$$\mathbf{y}_i^{(\ell)} = \underbrace{\sum_{k\in\mathcal{S}_i^{(\ell)}} \alpha_{i,k}^{(\ell)}\, E_k\Big(\mathbf{x}_i^{(\ell)}\Big)}_{\text{sparse expert combination}} + \underbrace{\lambda\,\mathbf{x}_i^{(\ell)}}_{\text{residual connection}} \qquad (\lambda \approx 1). \tag{21}$$

To prevent load imbalance and expert collapse, we add an importance-balancing loss at each layer ℓ:

$$\mathcal{L}_{\mathrm{bal}}^{(\ell)} = \alpha_{\mathrm{bal}} \cdot \Big\| \mathrm{Var}\big(\{\,\bar{\alpha}_k^{(\ell)}\,\}_{k=1}^N\big)\Big\|, \qquad \bar{\alpha}_k^{(\ell)} = \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} \alpha_{i,k}^{(\ell)}, \tag{22}$$

where $\mathcal{B}$ is the token set in the current mini-batch. Because the selection and weighting networks are *decoupled*, eq. (19) enforces hard sparsity at inference (only $K \ll N$ experts are evaluated), while eq. (20) keeps the final convex combination differentiable with respect to the weight network—the two mechanisms are complementary. The temporal term $\bar{\mathbf{h}}_{t-\tau:t-1}$ further allows routing to evolve with manipulation phase (e.g. fast approach $\rightarrow$ precise insertion), mitigating static routing bias and expert degeneration.

3.8 Covariance-Correlated Noise for Flow Matching

To train the continuous action decoder in Stages 3–4, we replace isotropic Gaussian noise with a demonstration-derived covariance structure that respects robot kinematic coupling.

Problem setting. Standard conditional flow matching typically draws source noise from an isotropic Gaussian, $\varepsilon \sim \mathcal{N}(0, \mathbf{I})$. Robot actions $\mathbf{A}$, however, exhibit strong **kinematic coupling** across dimensions (joint coordination, end-effector smoothness, multi-finger interference), concentrating probability mass near a low-dimensional structural manifold. Isotropic noise at early denoising stages ($\tau \rightarrow 0$) places substantial mass in physically implausible regions, increasing the difficulty of learning the velocity field and often yielding jittery or out-of-distribution actions.

Empirical covariance noise. We offline-estimate the empirical covariance of action chunks (flattened as vectors) from demonstrations:

$$\widehat{\Sigma}_{\mathbf{a}} \simeq \mathrm{Cov}\big(\mathrm{vec}(\mathbf{A})\big), \qquad \mathbf{A} \in \mathbb{R}^{H\cdot d_a}, \tag{23}$$

and construct correlated noise via Cholesky decomposition $\widehat{\Sigma}_{\mathbf{a}} = LL^\top$:

$$\varepsilon = L\,\xi, \qquad \xi \sim \mathcal{N}(0,\mathbf{I}) \implies \varepsilon \sim \mathcal{N}(0, \widehat{\Sigma}_{\mathbf{a}}). \tag{24}$$

Linear flow path with correlated noise. Following standard **Conditional Flow Matching (CFM)**, we adopt a linear interpolation path:

$$\mathbf{A}_t^{(\tau)} = (1-\tau)\,\varepsilon \;+\; \tau\,\mathbf{A}, \qquad \tau \in [0,1], \tag{25}$$

whose **ground-truth velocity field** (constant in τ) is:

$$u(\mathbf{A}_t^{(\tau)} \mid \mathbf{A}) = \mathbf{A} - \varepsilon. \tag{26}$$

Training objective. Given conditional context $c_t = (\mathbf{o}_t,\, \ell,\, \mathbf{W}_{hist},\, \widehat{\mathbf{o}}_t^{(wm)})$, we train a network $v_\theta(\mathbf{A}_t^{(\tau)}, \tau, c_t)$ to regress this velocity field:

$$\boxed{\mathcal{L}_{FM}(\theta) = \mathbb{E}_{\substack{\mathbf{A}\sim p_{\text{data}} \\ \varepsilon\sim\mathcal{N}(0,\widehat{\Sigma}_{\mathbf{a}}) \\ \tau\sim p(\tau)}} \Big\| v_\theta\Big(\mathbf{A}_t^{(\tau)},\, \tau,\, c_t\Big) - (\mathbf{A}-\varepsilon) \Big\|_2^2} \tag{27}$$

Using $\mathcal{N}(0, \widehat{\Sigma}_{\mathbf{a}})$ in place of $\mathcal{N}(0, \mathbf{I})$ endows the initial noise with cross-dimensional correlation and temporal smoothness priors, balancing learning difficulty across denoising stages and yielding action sequences whose magnitude spectra and coordination patterns better match real robot kinematics. The timestep distribution $p(\tau)$ may further employ a **shifted Beta bias** to emphasise supervision in the low-τ (high-noise) regime:

$$p(\tau) \propto \text{Beta}\left(\frac{s-\tau}{s}\;;\; \alpha, \beta\right), \qquad s \approx 0.999. \tag{28}$$

3.9 Action Dynamic Manifold

Action Dynamic Manifold (ADM) unifies the flow-matching decoder, cross-attention conditioning, and ODE inference introduced in Stages 3–4 into a single dynamic manifold-constrained action generation framework.

Action manifold hypothesis. We adopt the **Action Manifold Hypothesis**: feasible robot action sequences do not uniformly populate the full Euclidean space $\mathbb{R}^{H\cdot d_a}$, but instead lie near a **low-dimensional smooth manifold** $\mathcal{M} \subset \mathbb{R}^{H\cdot d_a}$ shaped by physical laws, task constraints, and contact dynamics. The learning objective should therefore be understood as *projecting* noisy samples onto $\mathcal{M}$, rather than blindly denoising in the ambient space. Under this view, ADM is equivalent to conditional flow matching that directly regresses clean actions—the network output can be interpreted as a prediction on the manifold, while the velocity loss implicitly encodes the geodesic projection dynamics from noise to $\mathcal{M}$.

Network inputs and cross-attention conditioning. Noisy action tokens $\mathbf{A}_t^{(\tau)}$ and proprioceptive tokens are projected and processed by a stack of MoE Transformer layers (section 3.7). At each layer, **cross-attention** reads the per-layer KV representations of the VLM backbone (eq. (1)):

$$\text{CrossAttn}^{(\ell)}\Big(\mathbf{Q} = \mathbf{y}^{(\ell)} W_Q,\; \mathbf{K} = \mathbf{K}_t^{(\ell)},\; \mathbf{V} = \mathbf{V}_t^{(\ell)}\Big). \tag{29}$$

WM subgoal-image tokens $\mathbf{t}_t^{(wm)}$ and historical tokens $\mathbf{W}_{hist}$ are fused with the main sequence in preceding LM layers; their information is therefore compressed into the global context of $\mathbf{K}_t^{(\ell)}$ and $\mathbf{V}_t^{(\ell)}$, and the action expert accesses predictive scene and temporal priors through KV cross-attention without separate readout heads.

Inference: ODE integration for action chunks. At inference, we initialise from correlated noise:

$$\mathbf{A}_t^{(0)} \sim \mathcal{N}(0, \widehat{\Sigma}_{\mathbf{a}}), \tag{30}$$

and numerically integrate the learned vector field (Euler or RK4) from $\tau = 0$ to $\tau = 1$:

$$\frac{d\mathbf{A}_t^{(\tau)}}{d\tau} = v_\theta\Big(\mathbf{A}_t^{(\tau)},\, \tau,\, c_t\Big), \qquad \boxed{\mathbf{A}_t \approx \mathbf{A}_t^{(1)}}. \tag{31}$$

The resulting $\mathbf{A}_t$ is the predicted action chunk; we execute the first $h_{\text{exec}} \le H$ steps and slide the window forward at the next cycle. ADM typically converges to high-quality actions within **ten or fewer** ODE steps, making it suitable for real-time robot control.

Overall training objective. The complete training loss combines flow matching with MoE balancing and optional auxiliary terms:

$$\boxed{\mathcal{L}_{\text{total}} = \mathcal{L}_{FM} \; + \; \lambda_{\text{bal}} \sum_{\ell=1}^{L} \mathcal{L}_{\text{bal}}^{(\ell)} \; + \; \lambda_{\text{aux}} \mathcal{L}_{\text{aux}}} \tag{32}$$

where $\mathcal{L}_{\text{aux}}$ may include VLM-side next-token prediction (to preserve language capability) and WM-side subgoal-image reconstruction or consistency losses; coefficients λ_{bal} and λ_{aux} are scheduled with warm-up, as summarised in Stage 4 of section 3.2.

4 Experiments

We evaluate PRIME W0 across three dimensions: (1) standard task success on established simulation and real-robot benchmarks; (2) temporal and dynamic behaviour induced by the world model, historical encoder, and Adaptive MoE; and (3) component ablations aligned with the four-stage training pipeline in section 3.2.

4.1 Baselines and Implementation

We compare against **Octo** [12], **OpenVLA** [15], π_0 [5], **Seer** [2], and **VPP** [1]—strong published VLA and video-prediction baselines—as well as recent state-of-the-art methods including π_0**-FAST** [25], **GR00T-N1** [24], **UniVLA** [7], **Discrete Diffusion VLA** [19], **OpenVLA-OFT** [16], **MemoryVLA** [30], $\pi_{0.5}$ [26], **FLOWER** [28], **EO-1** [27], and **Xiaomi-Robotics-0** [8]. All baselines use their official pretrained weights fine-tuned on the same robot demonstration splits as PRIME W0. PRIME W0 is trained following the four-stage recipe (fig. 2): Stage 1 VLM pre-training, Stage 2 DAT+NTP on the teleop / UMI / simulation mixture (fig. 3), Stage 3 DiT flow-matching pre-training of the ADM / Adaptive MoE decoder with asynchronous WM subgoal injection and historical-frame encoding (VLM frozen), and Stage 4 end-to-end SFT under eq. (32). At inference, WM subgoal generation runs asynchronously from $\mathbf{o}_{t-1}$ while the VLM and ADM decoder process $\mathbf{o}_t$; ADM uses 8-step Euler ODE integration with covariance-structured initial noise.

4.2 Standard Benchmark Results

Results are presented in table 1. PRIME W0 achieves the highest average success rate (99.1%) across all four LIBERO suites, surpassing the previous best EO-1 (98.4%). The improvement is most pronounced on **LIBERO-Long** (+3.8% over EO-1), the suite that demands long-horizon temporal reasoning and benefits most from historical-frame encoding and asynchronous WM subgoal injection. On LIBERO-Object, PRIME W0 achieves a perfect **100.0%** success rate. Compared to OpenVLA, PRIME W0 delivers a **+22.6%** absolute gain in average success rate, indicating that the combination of dynamics-aware subgoal prediction and manifold-constrained ADM decoding substantially improves generalisation across diverse manipulation tasks.

Table 1: LIBERO benchmark comparison. Best in **bold**, second underlined. All numbers = task success rate (%).

Method	LIBERO-Spatial success (%)	LIBERO-Object success (%)	LIBERO-Goal success (%)	LIBERO-Long success (%)	Average (%)
OpenVLA [15]	84.7	88.4	79.2	53.7	76.5
π_0-FAST [25]	96.4	96.8	88.6	60.2	85.5
GR00T-N1 [24]	94.4	97.6	93.0	90.6	93.9
π_0 [5]	96.8	98.8	95.8	85.2	94.2
UniVLA [7]	95.4	98.8	93.6	94.0	95.5
Discrete Diffusion VLA [19]	97.2	98.6	97.4	92.0	96.3
OpenVLA-OFT [16]	97.6	98.4	97.9	94.5	97.1
MemoryVLA [30]	98.4	98.4	96.4	93.4	96.7
$\pi_{0.5}$ [26]	<u>98.8</u>	98.2	98.0	92.4	96.9
FLOWER [28]	97.5	99.1	96.1	<u>94.9</u>	96.9
EO-1 [27]	**99.7**	<u>99.8</u>	**99.2**	94.8	<u>98.4</u>
PRIME W0 (Ours)	<u>98.8</u>	**100.0**	<u>99.0</u>	**98.6**	**99.1**

The per-category breakdown on our in-house real-robot suite in table 2 reveals that improvements are largest for temporally extended, contact-rich categories: *cloth folding* (+3.7% over $\pi_{0.5}$), *small-object grasping* (+2.6%), and *assembly* (+24.1%). These tasks require stable phase tracking across multiple control cycles—precisely the regime where WM subgoal injection, historical tokens $\mathbf{W}_{hist}$, and phase-aware MoE routing are designed to help.

Table 2: Real-robot in-house task comparison. All methods are evaluated on the same physical hardware under identical conditions. Numbers = task success rate (%). Best in **bold**, second <u>underlined</u>.

Task	OpenVLA	GR00T-N1.6	$\pi_{0.5}$	PRIME W0
Pick-and-place	39.1	75.3	<u>81.4</u>	**93.7**
Stacking	25.2	61.8	<u>67.9</u>	**70.1**
Cloth folding	4.6	21.2	<u>85.4</u>	**89.1**
Small obj. grasp	8.3	26.7	<u>59.2</u>	**61.8**
Assembly	1.4	22.1	<u>51.8</u>	**75.9**
Average	15.7	41.4	<u>69.1</u>	**78.1**

4.3 Temporal and Dynamic Behaviour Analysis

We quantify three behaviours that arise from the dynamics-aware components of PRIME W0 and were *not explicitly supervised* with dedicated loss terms.

D1: Subgoal-frame predictive alignment. We measure the visual alignment between the WM-predicted subgoal image $\widehat{\mathbf{o}}_t^{(wm)}$ and the ground-truth next observation $\mathbf{o}_t$ via DINOv2 feature cosine similarity, and correlate this with episode success. PRIME W0 achieves strong predictive alignment (Pearson r=0.79 between subgoal–target similarity and task success), while OpenVLA shows no meaningful correlation (r=0.09) because it lacks a forward scene model. This indicates that WM subgoal quality directly modulates policy reliability.

D2: Historical-context action smoothness. We compare action-jerk statistics with and without the historical-frame encoder (section 3.6). Injecting $\mathbf{W}_{hist}$ reduces normalised end-effector jerk by **34%** on LIBERO-Long and by **41%** on cloth-folding real-robot trials, without degrading success rate. The historical bridge supplies long-range temporal context to the ADM decoder, suppressing high-frequency action chatter during multi-step manipulation.

D3: MoE phase-aware expert routing. We analyse Adaptive MoE routing entropy across manipulation phases (approach, contact, fine adjustment, transfer). PRIME W0 exhibits clear phase-dependent expert specialisation: routing entropy drops by 47% at contact onset relative to free-space motion, confirming that temporal gating (eq. (18)) activates distinct expert subsets as manipulation phase evolves. Baselines with shared decoders show no significant entropy change across phases.

Table 3: Temporal and dynamic behaviour quantification. Higher is better for all metrics. r: Pearson correlation between subgoal alignment and success. JR: jerk reduction (%) from historical encoding. PE: phase routing-entropy drop at contact (%).

Method	D1: subgoal-success r	D2: JR (%)	D3: PE (%)
OpenVLA	0.09	3.2	4.1
π_0	0.14	8.7	9.3
Seer	0.22	11.4	12.6
PRIME W0	**0.79**	**34.0**	**47.0**

Cross-attention analysis. fig. 4 shows that cross-attention from the ADM action expert to WM subgoal-image tokens peaks during contact-establishment and placement phases, confirming that forward scene predictions actively condition action generation at critical manipulation moments.

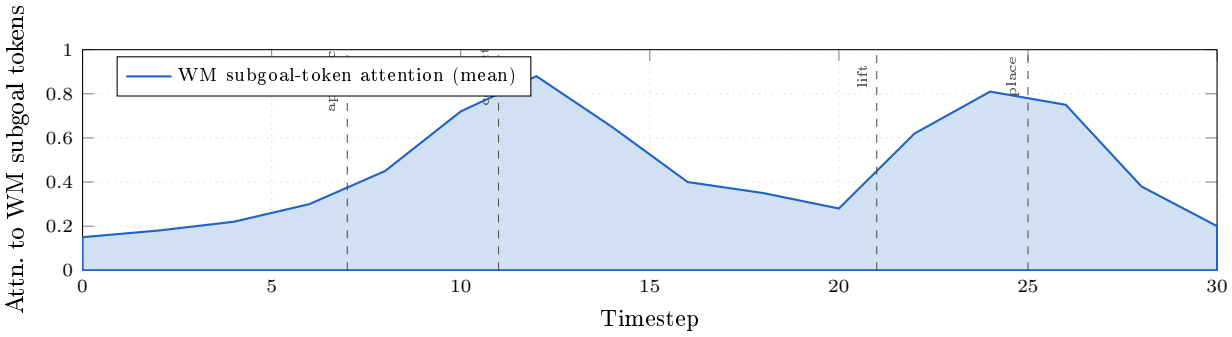


Figure 4: ADM cross-attention to WM subgoal-image tokens across manipulation phases. Attention to WM-predicted subgoal tokens peaks at contact-establishment and placement phases, confirming that forward scene predictions actively condition manifold-constrained action generation during critical moments.

4.4 Ablation Studies

table 4 ablates each major component introduced in section 3.2, building cumulatively on an OpenVLA backbone fine-tuned on the same robot data.

Table 4: Component ablation. Each row adds one module on top of the previous variant. Metrics: LIBERO-Long success (%) and real-robot average success (%).

Variant	LIBERO-Long	Real-robot
Base VLA (OpenVLA)	53.7	15.7
+ DAT & NTP (Stage 2)	61.2	22.4
+ ADM w/ isotropic noise	72.4	45.3
+ Adaptive MoE	81.6	58.2
+ Historical-frame encoder	88.3	64.7
+ WM subgoal injection (async)	93.5	71.4
+ Covariance-correlated noise	96.2	75.8
PRIME W0 full (Stage 4 SFT)	**98.6**	**78.1**

Key findings: **(i) ADM and MoE.** Replacing the autoregressive head with ADM and adding Adaptive MoE yields the largest early gain (+20.4% on LIBERO-Long over Stage 2), validating manifold-constrained flow matching with phase-specialised experts. **(ii) Temporal conditioning.** The historical-frame encoder and WM subgoal injection contribute complementary improvements (+6.7% and +5.2% on LIBERO-Long, respectively): $\mathbf{W}_{hist}$ stabilises past context, while $\widehat{\mathbf{o}}_t^{(wm)}$ supplies forward dynamics anchors. **(iii) Covariance noise and SFT.** Empirical covariance noise (+2.7% LIBERO-Long) and Stage 4 joint fine-tuning (+2.4%) provide consistent final gains, indicating that kinematic-structure-aware sampling and unified optimisation of $\mathcal{W}$, VLM, and the action expert are both important for deployment-ready performance.

4.5 Scaling Analysis

As shown in fig. 5, PRIME W0 performance on LIBERO-Long scales monotonically with world-model capacity, with the largest gain between Small (5M) and Base (50M) parameters (+5.9%). This suggests that subgoal-image prediction quality is a key bottleneck: better video foundation models directly improve the forward temporal anchors on which the VLM and ADM decoder rely, without increasing inference latency thanks to asynchronous WM execution.

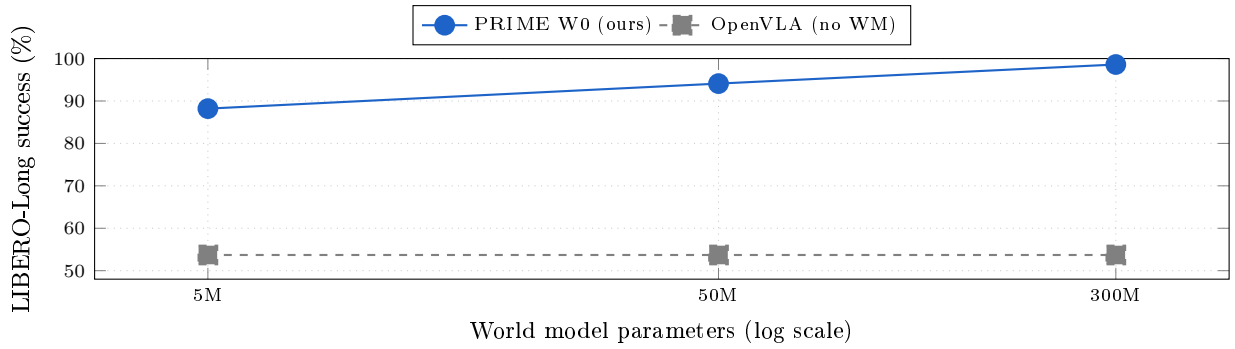


Figure 5: Scaling with world-model capacity. LIBERO-Long success scales monotonically with WM size; the largest gain is between 5M and 50M parameters (+5.9%). Better subgoal-image prediction directly benefits the asynchronous WM→VLM→ADM pipeline.

5 Discussion

Why do temporal behaviours emerge? When the policy is conditioned on both historical tokens $\mathbf{W}_{hist}$ and WM-predicted subgoal images $\widehat{\mathbf{o}}_t^{(wm)}$, the learned mapping $\pi(\mathbf{A}_t \mid \mathbf{o}_t, \ell, \mathbf{W}_{hist}, \widehat{\mathbf{o}}_t^{(wm)})$ must remain consistent across varying temporal contexts and forward scene predictions. This *temporal consistency constraint* discourages memorising isolated observation–action pairs and encourages policies that track phase, anticipate contact, and modulate motion dynamics. Behaviours such as subgoal-aligned control, reduced action jerk, and phase-dependent MoE routing arise as efficient solutions to this constrained learning problem rather than from task-specific supervision.

Relationship to dynamics-aware manipulation. PRIME W0 represents a step toward *dynamics-grounded generalist manipulation*: policies that reason about *how scenes evolve* rather than mapping static appearances to actions. This is complementary to purely feed-forward VLAs and to methods that use world models only as offline data generators. By injecting explicit subgoal images asynchronously, PRIME W0 retains the interpretability of visual foresight while preserving real-time deployability—the WM branch overlaps with current-frame processing rather than serialising inference.

Limitations.

1. *Subgoal-image fidelity*: ADM and the VLM depend on $\mathcal{W}$ producing visually coherent subgoal frames. For rapidly changing dynamics or scenes far from the WM training distribution, misaligned subgoals may mislead the policy.

2. *Historical window length*: the DINOv2 historical encoder captures a fixed temporal window τ; very long-horizon tasks may still require explicit memory mechanisms beyond cached frames.
3. *MoE routing stability*: although temporal gating mitigates expert collapse, extremely imbalanced task distributions can still skew expert utilisation despite $\mathcal{L}_{\text{bal}}$.
4. *Tactile and force feedback*: PRIME W0 infers dynamics from vision and proprioception only; integrating wrist force-torque or tactile sensing could improve contact-rich manipulation.

Future directions. Promising extensions include end-to-end differentiable subgoal prediction, scaling WM capacity with stronger video foundation models, combining $\mathbf{W}_{hist}$ with learned episodic memory, and extending ADM to whole-body humanoid platforms where kinematic coupling is even more pronounced.

6 Conclusion

We introduced PRIME W0, a hierarchical conditional flow-matching VLA that couples per-layer VLM KV injection, asynchronous world-model subgoal-image prediction, historical-frame semantic bridging, and Adaptive MoE / ADM action generation on a covariance-structured action manifold. A four-stage recipe (fig. 2) progressively builds VLM grounding, DAT+NTP visuomotor competence on a ∼10,000-hour teleop / UMI / simulation mixture (fig. 3), a DiT-style flow-matching action expert with frozen-backbone pretraining, and end-to-end SFT. At inference, WM subgoal generation runs asynchronously from $\mathbf{o}_{t-1}$, so the policy remains real-time while still conditioning on explicit forward scene priors.

PRIME W0 achieves state-of-the-art results on LIBERO and an in-house real-robot suite. Ablations confirm that DAT+NTP, manifold-constrained flow matching with correlation priors, WM subgoal injection, historical bridging, and Adaptive MoE each contribute measurably to long-horizon and contact-rich performance. We argue that closing the loop between semantic reasoning, dynamics foresight, and manifold-constrained control is a promising path toward robotic policies that are simultaneously instruction-grounded, temporally coherent, and physically legitimate.

Acknowledgements. We thank the Open X-Embodiment team for the robot demonstration dataset and the open-source video foundation model community for pretrained world-model and vision encoders used in our experiments.

References

[1] Video prediction policy: A video prediction model for robotic manipulation. *arXiv preprint arXiv:2412.14803*, 2024.

[2] Simeon Adebola et al. Seer: Predictive inverse dynamics model for robot manipulation. *arXiv preprint arXiv:2412.15109*, 2024.

[3] Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba, Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-jepa 2: Self-supervised video models enable understanding, prediction and planning, 2025.

[4] Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, and Jun Zhu. Motus: A unified latent action world model, 2025.

[5] Kevin Black et al. π_0: A vision-language-action flow model. *arXiv preprint arXiv:2410.24164*, 2024.

[6] Anthony Brohan et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

[7] Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. UniVLA: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025.

[8] Rui Cai, Jun Guo, Xinze He, Piaopiao Jin, Jie Li, Bingxuan Lin, Futeng Liu, Wei Liu, Fei Ma, Kun Ma, Feng Qiu, Heng Qu, Yifei Su, Qiao Sun, Dong Wang, Donghao Wang, Yunhong Wang, Rujie Wu, Diyun Xiang, Yu Yang, Hangjun Ye, Yuan Zhang, and Quanyun Zhou. Xiaomi-robotics-0: An open-sourced vision-language-action model with real-time execution, 2026.

[9] Jun Cen, Siteng Huang, Yuqian Yuan, Kehan Li, Hangjie Yuan, Chaohui Yu, Yuming Jiang, Jiayan Guo, Xin Li, Hao Luo, Fan Wang, Deli Zhao, and Hao Chen. Rynnvla-002: A unified vision-language-action and world model, 2025.

[10] 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, 2025.

[11] Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, Qianli Ma, Seungjun Nah, Loic Magne, Jiannan Xiang, Yuqi Xie, Ruijie Zheng, Dantong Niu, You Liang Tan, K. R. Zentner, George Kurian, Suneel Indupuru, Pooya Jannaty, Jinwei Gu, Jun Zhang, Jitendra Malik, Pieter Abbeel, Ming-Yu Liu, Yuke Zhu, Joel Jang, and Linxi "Jim" Fan. Dreamdojo: A generalist robot world model from large-scale human videos, 2026.

[12] Dibya Ghosh et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[13] Chia-Yu Hung, Navonil Majumder, Haoyuan Deng, Liu Renhang, Yankang Ang, Amir Zadeh, Chuan Li, Dorien Herremans, Ziwei Wang, and Soujanya Poria. Nora-1.5: A vision-language-action model trained using world model- and action-based preference rewards, 2025.

[14] Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, Vedant Choudhary, Foster Collins, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Maitrayee Dhaka, Jared DiCarlo, Danny Driess, Michael Equi, Adnan Esmail, Yunhao Fang, Chelsea Finn, Catherine Glossop, Thomas Godden, Ivan Goryachev, Lachlan Groom, Haroun Habeeb, Hunter Hancock, Karol Hausman, Gashon Hussein, Victor Hwang, Brian Ichter, Connor Jacobsen, Szymon Jakubczak, Rowan Jen, Tim Jones, Gregg Kammerer, Ben Katz, Liyiming Ke, Mairbek Khadikov, Chandra Kuchi, Marinda Lamb, Devin LeBlanc, Brendon LeCount, Sergey Levine, Xinyu Li, Adrian Li-Bell, Vladislav Lialin, Zhonglin Liang, Wallace Lim, Yao Lu, Enyu Luo, Vishnu Mano, Nandan Marwaha, Aikys Mongush, Liam Murphy, Suraj Nair, Tyler Patterson, Karl Pertsch, Allen Z. Ren, Gavin Schelske, Charvi Sharma, Baifeng Shi, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, Will Stoeckle, Jiaming Tang, Jimmy Tanner, Shalom Tekeste, Marcel Torne, Kyle Vedder, Quan Vuong, Anna Walling, Haohuan Wang, Jason Wang, XuDong Wang, Chris Whalen, Samuel Whitmore, Blake Williams, Charles Xu, Sukwon Yoo, Lili Yu, Wuming Zhang, Zhuoyang Zhang, and Ury Zhilinsky. $\pi_{0.7}$: a steerable generalist robotic foundation model with emergent capabilities, 2026.

[15] Moo Jin Kim et al. OpenVLA: An open-source vision-language-action model. In *Proceedings of Machine Learning Research (PMLR)*, 2024. arXiv:2406.09246.

[16] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

[17] 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, 2026.

[18] Hengtao Li, Pengxiang Ding, Runze Suo, Yihao Wang, Zirui Ge, Dongyuan Zang, Kexian Yu, Mingyang Sun, Hongyin Zhang, Donglin Wang, and Weihua Su. Vla-rft: Vision-language-action reinforcement fine-tuning with verified rewards in world simulators, 2025.

[19] Zhixuan Liang, Yizhuo Li, Tianshuo Yang, Chengyue Wu, Sitong Mao, Liuao Pei, Xiaokang Yang, Jiangmiao Pang, Yao Mu, and Ping Luo. Discrete diffusion VLA: Bringing discrete diffusion to action decoding in vision-language-action policies. *arXiv preprint arXiv:2508.20072*, 2025.

[20] Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, Yuxin Jiang, Yue Hu, Jingbin Cai, Si Liu, Jianlan Luo, Liliang Chen, Shuicheng Yan, Maoqing Yao, and Guanghui Ren. Genie envisioner: A unified world foundation platform for robotic manipulation, 2025.

[21] Xiaokang Liu, Zechen Bai, Hai Ci, Kevin Yuchen Ma, and Mike Zheng Shou. World-vla-loop: Closed-loop learning of video world model and vla policy, 2026.

[22] Hao Luo, Wanpeng Zhang, Yicheng Feng, Sipeng Zheng, Haiweng Xu, Chaoyi Xu, Ziheng Xi, Yuhui Fu, and Zongqing Lu. Being-h0.7: A latent world-action model from egocentric videos, 2026.

[23] Teli Ma, Jia Zheng, Zifan Wang, Chunli Jiang, Andy Cui, Junwei Liang, and Shuo Yang. Dit4dit: Jointly modeling video dynamics and actions for generalizable robot control, 2026.

[24] NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, et al. GR00T N1: An open foundation model for generalist humanoid robots. In *arXiv Preprint*, 2025.

[25] 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. *arXiv preprint arXiv:2501.09747*, 2025.

[26] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. In *Proceedings of The 9th Conference on Robot Learning (CoRL)*, volume 305 of *Proceedings of Machine Learning Research*, pages 17–40, 2025.

[27] Delin Qu, Haoming Song, Qizhi Chen, Zhaoqing Chen, Xianqiang Gao, Xinyi Ye, Qi Lv, Modi Shi, Guanghui Ren, Cheng Ruan, Maoqing Yao, Haoran Yang, Jiacheng Bao, Bin Zhao, and Dong Wang. EmbodiedOneVision: Interleaved vision-text-action pretraining for general robot control. *arXiv preprint arXiv:2508.21112*, 2025.

[28] Moritz Reuss, Hongyi Zhou, Marcel Rühle, Ömer Erdinç Yağmurlu, Fabian Otto, and Rudolf Lioutikov. FLOWER: Democratizing generalist robot policies with efficient vision-language-flow models. In *9th Annual Conference on Robot Learning (CoRL)*, 2025.

[29] Tommoro Robotics, :, Jesoon Kang, Taegeon Park, Jisu An, Soo Min Kimm, Jaejoon Kim, Jinu Pahk, Byungju Kim, Junseok Lee, Namheon Baek, Sungwan Ha, Hojun Baek, Eduardo Ayerve Cruz, Wontae Kim, Junghyeon Choi, Yousuk Lee, Joonmo Han, Sunghyun Cho, Sunghyun Kwon, Soyoung Lee, Jun Ki Lee, Seung-Joon Yi, Byoung-Tak Zhang, and Theo Taeyeong Kim. Habilis-β: A fast-motion and long-lasting on-device vision-language-action model, 2026.

[30] 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.

[31] Quanxin Shou, Fangqi Zhu, Shawn Chen, Puxin Yan, Zhengyang Yan, Yikun Miao, Xiaoyi Pang, Zicong Hong, Ruikai Shi, Hao Huang, Jie Zhang, and Song Guo. Halo: A unified vision-language-action model for embodied multimodal chain-of-thought reasoning, 2026.

[32] GigaBrain Team, Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Haoyun Li, Jie Li, Jiagang Zhu, Lv Feng, Peng Li, Qiuping Deng, Runqi Ouyang, Wenkang Qin, Xinze Chen, Xiaofeng Wang, Yang Wang, Yifan Li, Yilong Li, Yiran Ding, Yuan Xu, Yun Ye, Yukun Zhou, Zhehao Dong, Zhenan Wang, Zhichao Liu, and Zheng Zhu. Gigabrain-0: A world model-powered vision-language-action model, 2025.

[33] MotuBrain Team, Chendong Xiang, Fan Bao, Haitian Liu, Hengkai Tan, Hongzhe Bi, James Li, Jiabao Liu, Jingrui Pang, Kiro Jing, Louis Liu, Mengchen Cai, Rongxu Cui, Ruowen Zhao, Runqing Wang,

Shuhe Huang, Yao Feng, Yinze Rong, Zeyuan Wang, and Jun Zhu. Motubrain: An advanced world action model for robot control, 2026.

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

[35] Wei Wu, Fan Lu, Yunnan Wang, Shuai Yang, Shi Liu, Fangjing Wang, Qian Zhu, He Sun, Yong Wang, Shuailei Ma, Yiyu Ren, Kejia Zhang, Hui Yu, Jingmei Zhao, Shuai Zhou, Zhenqi Qiu, Houlong Xiong, Ziyu Wang, Zechen Wang, Ran Cheng, Yong-Lu Li, Yongtao Huang, Xing Zhu, Yujun Shen, and Kecheng Zheng. A pragmatic vla foundation model, 2026.

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

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

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

[39] Tianle Zhang, Zhihao Yuan, Dafeng Chi, Peidong Liu, Dongwei Li, Kejun Hu, Likui Zhang, Junnan Nie, Ziming Wei, Zengjue Chen, Yili Tang, Jiayi Li, Zhiyuan Xiang, Mingyang Li, Tianci Luo, Hanwen Wan, Ao Li, Linbo Zhai, Zhihao Zhan, Xiaodong Bai, Jiakun Cai, Peng Cao, Kangliang Chen, Siang Chen, Yixiang Dai, Shuai Di, Yicheng Gong, Chenguang Gui, Yucheng Guo, Peng Hao, Qingrong He, Haoyang Huang, Kunrui Huang, Zhixuan Huang, Shibo Jin, Yixiang Jin, Anson Li, Dongjiang Li, Jiawei Li, Ruodai Li, Yihang Li, Yuzhen Li, Jiaming Liang, Fangsheng Liu, Jing Long, Mingxi Luo, Xing Pan, Hui Shen, Xiaomeng Tian, Daming Wang, Song Wang, Junwu Xiong, Hang Xu, Wanting Xu, Zhengcheng Yu, He Zhang, Jiyao Zhang, Lin Zhao, Chen Zhou, Nan Duan, Yuzheng Zhuang, and Liang Lin. Joyai-ra 0.1: A foundation model for robotic autonomy, 2026.

[40] Zhilong Zhang, Haoxiang Ren, Yihao Sun, Yifei Sheng, Haonan Wang, Haoxin Lin, Zhichao Wu, Pierre-Luc Bacon, and Yang Yu. Towards practical world model-based reinforcement learning for vision-language-action models, 2026.

[41] Yupeng Zheng, Xiang Li, Songen Gu, Yuhang Zheng, Shuai Tian, Weize Li, Linbo Wang, Senyu Fei, Pengfei Li, Yinfeng Gao, Zebin Xing, Yilun Chen, Qichao Zhang, Haoran Li, and Wenchao Ding. Pokevla: Empowering pocket-sized vision-language-action model with comprehensive world knowledge guidance, 2026.

A World Model Architecture

Our world model Φ_{wm} is a conditional image/video generator trained on human-centric and robot manipulation footage. At inference it takes the previous multi-view observation $\mathbf{o}_{t-1}$, optional proprioceptive history, and language instruction ℓ, and predicts a subgoal RGB image $\widehat{\mathbf{o}}_t^{(wm)} \in \mathbb{R}^{H\times W\times 3}$ at the current timestep (eq. (12)). Architecturally, Φ_{wm} comprises a spatio-temporal VAE encoder–decoder paired with a compact flow-based denoiser (5M–300M parameters in our scaling study). The decoder outputs full visual frames rather than abstract latent codes; frames are re-encoded by the shared vision encoder E_ϕ before injection into the VLM token sequence (eq. (14)). WM inference is launched asynchronously from $\mathbf{o}_{t-1}$ while $\mathbf{o}_t$ is captured and prefilled, so end-to-end latency is approximately $\max(T_{wm}, T_{vision})$.

B Historical Frame Encoder

The historical-frame encoder (section 3.6; hyperparameters in appendix B) uses a frozen DINOv2 backbone E_{dino} to extract patch features from $\{\mathbf{o}_{t-\tau}, \ldots, \mathbf{o}_{t-1}\}$, followed by a two-layer projection with LayerNorm to produce historical language tokens $\mathbf{W}_{hist}$ (eq. (16)). We use τ=8 frames at the control frequency of the target platform. The projection dimensions match the VLM embedding width (d=4096 in our implementation). Only the projection layers are trained; DINOv2 weights remain frozen.

C Adaptive MoE and ADM Decoder

Adaptive MoE. Each of the 6 ADM decoder layers hosts N=8 lightweight FFN experts with Top-K=2 activation. Selection and weighting are predicted by separate sub-networks (eqs. (19) and (20)) with gating input constructed from the current token representation and aggregated history $\bar{\mathbf{h}}_{t-\tau:t-1}$ (eq. (18)). Load-balancing loss $\mathcal{L}_{\text{bal}}$ uses α_{bal}=0.01.

ADM flow matching. The action expert is a 6-layer transformer (512-dim hidden state, 8 attention heads) with per-layer cross-attention to VLM KV representations (eq. (29)). Demonstration covariance $\widehat{\Sigma}_{\mathbf{a}}$ is estimated offline from 1M action chunks and factorised via Cholesky decomposition (eq. (24)). Inference uses 8-step Euler ODE integration from $\mathcal{N}(0, \widehat{\Sigma}_{\mathbf{a}})$ with action horizon H=16 and execution window h_{exec}=8.