ONE STATE, ONE ROLL: UNIFIED PHYSICAL LATENT FLOW FOR ROBOT MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Embodied foundation models are increasingly moving toward multimodal prediction, jointly covering actions, proprioceptive states, and visual observations. Yet these modalities are usually generated through separate modality-specific pathways. Such separation makes cross-modal physical consistency an additional coordination problem: different pathways can predict incompatible futures, while each modality is evolved without a shared representation of the physical transition. Moreover, most models generate every future chunk from random noise, leaving the local continuity of an already realized physical trajectory largely unused. Robot control is fundamentally the continuous evolution of a physical trajectory. We introduce OneRoll, a unified physical latent flow for robot manipulation. OneRoll linearly fuses the three modalities into one physical token and recovers them through linear projections. The token's velocity on the physical manifold is thus the sum of their linearly mapped derivatives, directly supervising a shared vector field whose integration extends the realized trajectory into the future. This couples the modalities during evolution, improving physical consistency and rollout continuity. Experiments on both simulated and real-robot tasks show encouraging performance, suggesting that unified physical evolution is a promising foundation architecture for embodied models.

1 INTRODUCTION

Robot foundation models learn general-purpose control from visual, language, and proprioceptive context. Vision-language-action models (VLAs) predict executable actions, while world-action models (WAMs) additionally forecast how the robot and its environment may evolve. Robot control is fundamentally the evolution of a physical trajectory, with actions, proprioceptive states, and environmental observations forming intrinsic, coupled components of each physical transition. Existing models do not represent them as one evolving variable: actions are decoded by a policy, whereas states and observations either remain conditioning inputs or are generated through a world-modeling pathway. Even when future actions and physical changes are learned jointly, their representations and generation processes usually remain modality-specific, forcing the model to coordinate several correlated predictions instead of evolving a single physical state.

This separation takes two common forms. Sequential approaches first predict a future world state and then infer an action, introducing an intermediate stage whose errors can propagate into control. Parallel approaches remove this ordering but retain separate modality-specific prediction streams; even when the streams exchange information or co-evolve, they must still be coordinated to avoid inconsistent futures. In both cases, physical consistency is imposed across prediction pathways rather than built into the evolving representation. A unified physical manifold instead makes all modalities describe the same latent future: states and observations constrain physically feasible actions, while actions identify the controllable changes in that future.

Another issue concerns temporal continuity. Most policies form a new future prediction at every planning step by transporting an unstructured prior. However, physical trajectories are locally continuous, and decision making is a task-directed continuation of the joint trajectory already realized. Starting from noise leaves this continuity unused and requires the model to generate the future from an unrelated source. Recent work has begun to exploit this structure through modality-specific paths, including state-to-action (Dai et al., 2026), action-to-action (Jia et al., 2026; Kang

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

et al., 2026), observation-to-action (Gao et al., 2026a), and observation-to-observation prediction (Wang et al., 2026c; Liu et al., 2026). Such methods improve continuity within an individual path, but multiple paths can still produce inconsistent futures without a shared physical constraint. A physical-to-physical, or more generally any-to-any, path evolves all modalities in the same transition and preserves their physical coupling.

Motivated by these observations, we introduce OneRoll, which models robot interaction as a physical manifold trajectory in a unified latent space. At each control step, the action, proprioceptive state, and observation are represented together as a single physical token. Rather than generating each future state from scratch, the model learns a vector field in this space and integrates along it to continuously extend the realized physical trajectory into the future. Actions, states, and observations are then recovered as different linear projections of the evolved physical latent. In this way, the three modalities share the same physical evolution, and their linear projections provide direct, jointly aligned supervision to the underlying latent dynamics, strengthening both physical continuity and cross-modal consistency. Our contributions are summarized as follows:

1. We reformulate robot control as the evolution of an underlying physical trajectory, with action, proprioceptive state, and observation representing different aspects of the same physical trajectory.
2. We instantiate this formulation with a shared vector field on the physical manifold to evolve the realized physical state into the future state. The three modalities are recovered by simple linear projections to strengthen physical continuity and cross-modal consistency.
3. We scale the proposed formulation to 30,000 hours of heterogeneous cross-embodiment data and demonstrate its effectiveness across simulated and real-world evaluations.

2 Related Work

2.1 Embodied Foundation Models for Robot Manipulation

Robot foundation models for manipulation have primarily developed along two directions: vision-language-action (VLA) models, which predict executable actions from multimodal context (Gemini Robotics Team et al., 2025; Shukor et al., 2025), and world-action models (WAMs), which additionally model how the scene evolves alongside the robot action (Ye et al., 2026; Wang et al., 2026b). Both directions are increasingly becoming multimodal, combining visual observations, language instructions, proprioception, and action-related supervision in a shared model. Recent VLA models introduce future-prediction branches and auxiliary objectives to enrich policy features with anticipated visual dynamics and world knowledge (Zhang et al., 2025; Ma et al., 2026). Joint WAMs instead predict future observations and actions: a shared backbone exchanges features between the two modalities, after which separate modality-specific branches decode the predicted observation and action (Zhou et al., 2026; Pan et al., 2026). Despite this multimodal coupling, these approaches generally retain distinct predictive variables and modality-specific generation pathways, rather than evolving action, proprioception, and observation as a single physical latent state. This separation can weaken cross-modal consistency, require additional coordination between predictions, and allow the predicted action and future observation to represent mismatched physical futures.

2.2 Structured Priors and Continuous Dynamics in Robot Control

Robot motion is temporally continuous, motivating generation from informed sources rather than an uninformative prior. Warm-Start Diffusion predicts context-dependent Gaussian moments to initialize conditional generation (Scholz & Turner, 2025). LeaP provides state-to-action initialization through a proprioception-conditioned source prior (Dai et al., 2026). A2A learns action-to-action transport from historical actions to the next chunk (Jia et al., 2026). WarmPrior similarly centers its prior on previous or previewed chunks, while STEP predicts a warm start from action history and current observations before diffusion refinement (Kang et al., 2026; Li et al., 2026). VITA follows an observation-to-action path, flowing directly from visual latents to action latents (Gao et al., 2026a).

In observation-to-observation prediction, OFlow forecasts object-aware semantic latents with temporal flow matching to guide actions, while ODEWorld integrates visual representations through a physical-time vector field (Wang et al., 2026c; Liu et al., 2026). These methods exploit continuity

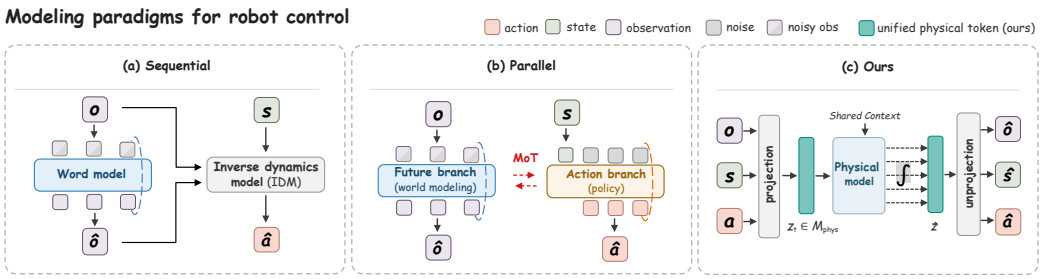

Figure 1: Comparison of robot-control modeling paradigms. Sequential and parallel methods maintain separate world-model and action pathways, whereas OneRoll evolves synchronized modalities through a unified physical token. The unified representation makes cross-modal consistency a property of the evolving physical state.

within individual paths, but combining separate paths does not ensure consistent multimodal futures. OneRoll instead defines a single physical-to-physical path, jointly advancing action, proprioception, and observation through one latent state and shared vector field.

3 Methodology

3.1 Preliminaries and Problem Formulation

To clarify the modeling distinction, we first characterize what is evolved by existing robot foundation models. Actions, proprioceptive states, and observations may be generated sequentially or in parallel, but their predictive processes are typically defined in separate modality spaces. Let $\mathbf{x}^m$ denote the variable of modality m, and let $\mathbf{h}_t^m$ denote the information available to its predictor. The corresponding predictive distribution is

$$p_\theta\left(\mathbf{x}^m_{t+1:t+H} \mid \mathbf{h}_t^m\right), \qquad m \in \{a, s, o\}. \tag{1}$$

where m ranges only over the modalities predicted by a given model. An action-only VLA uses $m = a$, while models with world-modeling branches may include $m = o$. For parallel prediction, $\mathbf{h}_t^m$ is derived from a shared context; for sequential prediction, it may contain previously generated modalities. They often use distinct, large decoders. Thus, their evolution remains modality-specific. Their physical consistency must still be coordinated across different prediction pathways.

Figure 1 contrasts this separation with our unified formulation: existing sequential and parallel approaches retain distinct world-model and action pathways, whereas OneRoll maps synchronized modalities to one physical token and evolves them with a shared physical model.

However, these modalities over the same period jointly describe one controlled physical transition. Thus we change the modeling unit from separate modality futures to a joint physical manifold. Let c_t denote the task context at time t. We encode the most recent H-step interaction segment (a mini-chunk) as one physical token, and model its conditional evolution as

$$\mathbf{z}_t = \mathcal{E}(\mathbf{a}_{t-H+1:t}, \mathbf{s}_{t-H+1:t}, \mathbf{o}_{t-H+1:t}) \in \mathcal{M}_{\text{phys}}, \ \mathbf{z}_{t+H} \sim p_\theta(\cdot \mid \mathbf{z}_t, c_t). \tag{2}$$

Here, $\mathbf{z}_t$ is a latent point on manifold $\mathcal{M}_{\text{phys}}$. Because $\mathbf{z}_t$ is constructed from a mini-chunk rather than a single instant, it captures both the current physical state and its recent trend. The conditional distribution $p_\theta(\cdot \mid \mathbf{z}_t, c_t)$ defines an evolution law on $\mathcal{M}_{\text{phys}}$, rather than separate predictions in modality-specific spaces.

Moreover, future prediction should be treated as physical continuation rather than independent generation. Classic flow-based policies construct each future chunk from an unstructured source, and therefore restart the generative process at every replanning step. In contrast, consecutive physical states should lie on a continuous path originating from the last physical state. Let $\mathbf{z}(\tau)$ denote such a path on $\mathcal{M}_{\text{phys}}$, v_θ denote its learned velocity field. The future physical state is obtained by integrating over the path parameter:

$$\widehat{\mathbf{z}}_{t+H} = \mathbf{z}_t + \int_1^0 v_\theta(\mathbf{z}(\tau), \tau; c_t)\, \mathrm{d}\tau. \tag{3}$$

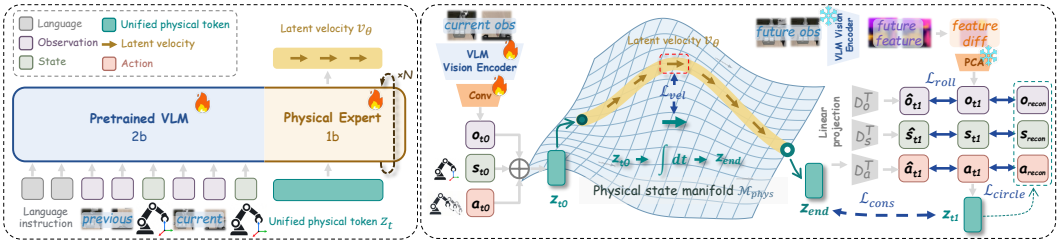

Figure 2: OneRoll architecture. The VLM supplies task context; the physical expert evolves fused action, state, and observation tokens and decodes future modalities. Linear readouts expose the shared latent evolution as action, proprioceptive state, and observation predictions.

t indexes control time, whereas $\tau \in [0, 1]$ parameterizes the integration path. The model therefore evolves the complete joint physical state from realized history to future.

We develop a network architecture and training paradigm that connect the unified physical latent space with actions, proprioceptive states, and observations through modality-specific linear projections. In this way, changes in the physical latent space are reflected in all three modalities, while the information provided by these modalities together helps learn the latent representation and its dynamics. The following subsection introduces the input representation and network architecture, and then describes the training objectives and inference procedure.

3.2 Unified Physical Latent Flow

This subsection describes how OneRoll maps action, proprioception, and observation mini-chunks into the unified physical latent space, evolves the resulting physical token with a learned network along its manifold, and maps it back to the three modalities. We describe these components below.

Figure 2 gives an overview of the complete pipeline, including multimodal encoding, physical-manifold integration, linear modality readouts, and the velocity, rollout, and latent-cycle objectives.

Joint physical representation. A mini-chunk consists of H consecutive actions and proprioceptive states, together with visual observations sampled over the same period. The action and state at each control step are mapped into a shared d-dimensional space through simple linear projections, denoted by $\mathcal{E}_a$ and $\mathcal{E}_s$. For an observation, we use the frozen visual encoder of the pretrained VLM and project its feature onto a PCA basis estimated from the training data. This produces a 512-dim visual coordinate $\mathbf{q}_k$, which is further mapped to the same latent width by a linear projection $\mathcal{E}_o$.

To obtain the physical latent vector, the three embeddings associated with the same control step are added, while the resulting embeddings from different steps are concatenated in temporal order. The k-th physical embedding and the complete mini-chunk physical token are therefore defined as

$$\begin{aligned} \mathbf{z}_{t,k} &= \mathcal{E}_a(\mathbf{a}_{t-H+k}) + \mathcal{E}_s(\mathbf{s}_{t-H+k}) + \mathcal{E}_o(\mathbf{q}_{t-H+k}), \\ \mathbf{z}_t &= \mathrm{cat}(\mathbf{z}_{t,1}, \ldots, \mathbf{z}_{t,H}) \in \mathcal{M}_{\mathrm{phys}} \subset \mathbb{R}^{H \times d}. \end{aligned} \tag{4}$$

Here, H and d denote the chunk length and physical latent width, respectively. When the modalities are sampled at different frequencies, each row incorporates the modalities available at that control step, while any unavailable modality contributes a zero vector. Consequently, addition fuses the modalities that describe the same physical step, while concatenation preserves their temporal order. The complete H-row representation forms the single physical token $\mathbf{z}_t$ introduced in Section 2.1, rather than separate sequences for each modality.

VLM context and physical integrator. Our architecture follows the familiar VLA design consisting of a pretrained vision–language model and a transformer expert. As in a standard VLA, the pretrained VLM processes the language instruction and provides the task context c_t. Relative to the single-frame input of the base VLA, we provide two temporally ordered multi-view observations at $t-\Delta$ and t. Moreover, each multi-view observation is followed by a token encoding the proprioceptive state at the same time step. Together with the language instruction, two temporally consecutive frames form the VLM task context. This short temporal context captures both the current physical configuration and its recent motion, helping the model infer how the physical state evolves.

In parallel with the VLM, we introduce a Mixture-of-Transformers physical expert. The VLM tokens and physical tokens are placed in a shared attention space at each transformer layer, allowing the physical expert to directly access the task context encoded by the VLM. Structurally, the physical expert follows a flow-based VLA action expert, but it operates on a different variable. While an action expert predicts an executable action chunk directly, our physical expert takes the current physical token $\mathbf{z}(\tau)$ and integration time τ as input and predicts an instantaneous velocity with the same shape as the physical token.

Starting from $\mathbf{z}(1) = \mathbf{z}_t$, the physical velocity and the integrated future token are given by

$$v_\theta(\mathbf{z}(\tau), \tau; c_t) = \mathcal{F}_{\text{phys}}(\mathbf{z}(\tau), \phi(\tau); c_t)\,, \quad \widehat{\mathbf{z}}_{t+H} = \mathbf{z}_t + \int_1^0 v_\theta(\mathbf{z}(\tau), \tau; c_t)\ \mathrm{d}\tau. \tag{5}$$

Here, $\phi(\tau)$ denotes the embedding of the integration time. Throughout the integration, the VLM context remains fixed, while the predicted velocity continuously updates the joint physical token along the manifold $\mathcal{M}_{\text{phys}}$. Actions, proprioceptive states, and observations therefore evolve through the same integration process in the unified physical manifold.

Modality readout and physical velocity supervision. After integration, we omit prediction hats for clarity and denote the evolved physical token by $\mathbf{z}_{t+H}$. Actions, proprioceptive states, and visual coordinates are recovered row by row through linear readout matrices: Specifically, $\mathbf{D}_a$, $\mathbf{D}_s$, and $\mathbf{D}_o$ read out actions, proprioceptive states, and visual coordinates, respectively, from the physical latent. Let $\mathbf{x}^a = \mathbf{a}$, $\mathbf{x}^s = \mathbf{s}$, and $\mathbf{x}^q = \mathbf{q}$, with $\mathbf{D}_q = \mathbf{D}_o$. For each modality $m \in \{a, s, q\}$, the readout and its velocity target can be written compactly as

$$\mathbf{x}^m_{t+1:t+H} = \mathbf{z}_{t+H}\mathbf{D}_m^\top, \qquad \mathbf{u}^m_k = \frac{\mathrm{d}\mathbf{x}^m_k(\tau)}{\mathrm{d}\tau} = \mathbf{x}^m_{t-H+k} - \mathbf{x}^m_{t+k}, \qquad m \in \{a, s, q\}. \tag{6}$$

The observation readout predicts the fixed visual coordinate $\mathbf{q}$ rather than raw RGB pixels and is applied only at the sampled visual steps. Although the frozen mapping from an image to $\mathbf{q}$ is nonlinear, the projection between $\mathbf{q}$ and the physical latent is linear. Thus, changes in the physical token are directly reflected in the action, state, and observation-coordinate outputs.

For visual coordinates, $\mathbf{u}^q_k$ is defined only at sampled visual steps and is set to zero otherwise. Writing the three linear encoders as $\mathbf{W}_a$, $\mathbf{W}_s$, and $\mathbf{W}_o$, the velocity of the physical embedding at step k follows directly from linearity:

$$\mathbf{u}^z_k = \frac{\mathrm{d}\mathbf{z}_k(\tau)}{\mathrm{d}\tau} = \frac{\mathrm{d}}{\mathrm{d}\tau}\left(\mathbf{W}_a\mathbf{a}_k(\tau) + \mathbf{W}_s\mathbf{s}_k(\tau) + \mathbf{W}_o\mathbf{q}_k(\tau)\right) = \mathbf{W}_a\mathbf{u}^a_k + \mathbf{W}_s\mathbf{u}^s_k + \mathbf{W}_o\mathbf{u}^q_k. \tag{7}$$

Therefore, the action, state, and observation-coordinate velocities computed from the dataset are mapped directly into the physical latent space through the same linear projections used to construct the physical token. No additional nonlinear transformation or auxiliary dynamics model is required to obtain $\mathbf{u}^z_k$. This establishes a direct correspondence between supervision in the three observable modalities and the learned velocity on the physical manifold. For observations, the velocity is first computed in the fixed visual coordinate $\mathbf{q}$ and then linearly projected into the physical latent space. The following subsection uses these physical velocity targets to define the training objective for v_θ.

3.3 Training Objectives and Inference

The training objectives impose two complementary constraints on the physical latent space. The evolution losses, $\mathcal{L}_{\text{vel}}$ and $\mathcal{L}_{\text{roll}}$, constrain how the physical latent evolves and whether the resulting rollout produces the desired future trajectory. The physical representation losses, $\mathcal{L}_{\text{cons}}$ and $\mathcal{L}_{\text{cycle}}$, constrain what the physical latent represents by aligning the integrated endpoint with the encoded future and preserving its multimodal physical information.

Evolution supervision. For each training window, the preceding and following mini-chunks are encoded as $\mathbf{z}_t$ and $\mathbf{z}_{t+H}$. Following Eq. equation 7, we construct the ground-truth physical velocity $\mathbf{u}^z$ from the instantaneous changes of the observable action, state, and visual coordinates. If a modality is unavailable at a particular step, its contribution is set to zero. We supervise the physical expert at a randomly sampled point $\mathbf{z}(\tau)$ along the physical path:

$$\mathcal{L}_{\text{vel}} = \left\| v_\theta(\mathbf{z}(\tau), \tau; c_t) - \mathbf{u}^z \right\|_2^2. \tag{8}$$

This loss constrains the local velocity field to follow the observable physical evolution of the synchronized trajectory.

To stabilize the result of the complete finite-step integration, we perform the same Euler rollout during training as used at inference and denote its endpoint by $\mathbf{z}_{\text{end}}$. All future modalities are decoded from this integrated endpoint and supervised against the corresponding future mini-chunk. For compactness, let $\mathbf{a}^+ = \mathbf{a}_{t+1:t+H}$, $\mathbf{s}^+ = \mathbf{s}_{t+1:t+H}$, and $\mathbf{q}^+ = \mathbf{q}_{t+1:t+H}$. Let $\mathcal{G}$ denote the rows with valid visual coordinates.

$$\mathcal{L}_{\text{roll}} = \left\|\mathbf{z}_{\text{end}}\mathbf{D}_a^\top - \mathbf{a}^+\right\|_1 + \left\|\mathbf{z}_{\text{end}}\mathbf{D}_s^\top - \mathbf{s}^+\right\|_1 + \left\|(\mathbf{z}_{\text{end}})_{\mathcal{G}}\,\mathbf{D}_o^\top - \text{sg}\Big(\big(\mathbf{q}^+\big)_{\mathcal{G}}\Big)\right\|_1 \tag{9}$$

The action term directly supervises the executable control output, while the state and observation terms provide auxiliary supervision for the integrated trajectory. Gradients are propagated through the complete Euler rollout.

Physical latent consistency and cycle reconstruction. The evolution losses constrain how the physical latent moves, but do not by themselves ensure that the integrated endpoint remains in the same physical latent space as the encoded future. We therefore require the integrated endpoint to represent the same future physical state as the encoded target:

$$\mathcal{L}_{\text{cons}} = \|\mathbf{z}_{\text{end}} - \mathbf{z}_{t+H}\|_2^2 . \tag{10}$$

This objective constrains the finite-step transport map to remain consistent with the physical latent trajectory defined by the encoder.

We further require the encoded target $\mathbf{z}_{t+H}$ to preserve the multimodal information of the future mini-chunk:

$$\mathcal{L}_{\text{cycle}} = \left\|\mathbf{z}_{t+H}\mathbf{D}_a^\top - \mathbf{a}^+\right\|_1 + \left\|\mathbf{z}_{t+H}\mathbf{D}_s^\top - \mathbf{s}^+\right\|_1 + \left\|(\mathbf{z}_{t+H})_{\mathcal{G}}\,\mathbf{D}_o^\top - \text{sg}\Big(\big(\mathbf{q}^+\big)_{\mathcal{G}}\Big)\right\|_1 \tag{11}$$

The encoded-target loss acts as a multimodal reconstruction loss that closes the cycle across the three observable modalities, while the consistency loss ensures that the integrated endpoint shares the same physical representation. Together, they prevent the physical latent space from discarding modality-specific information or assigning different representations to the same future physical state.

The complete objective can therefore be written as

$$\mathcal{L} = \mathcal{L}_{\text{vel}} + \mathcal{L}_{\text{roll}} + \mathcal{L}_{\text{cons}} + \mathcal{L}_{\text{cycle}}. \tag{12}$$

Scalar coefficients used in the implementation are omitted here for clarity. All modality-dependent terms are evaluated only over valid entries. The action and state terms use their dimensional validity masks, while $\mathcal{G}$ contains the rows with valid visual coordinates.

Closed-loop inference. At each replanning step, the VLM processes the language instruction, two consecutive observation frames, and their corresponding state tokens once to obtain c_t. The latest realized mini-chunk is encoded as $\mathbf{z}_t$, after which the physical expert applies the same Euler integration procedure used during training. The integrated endpoint $\mathbf{z}_{\text{end}}$ is linearly projected to an action chunk, and the controller executes its first H actions. The issued actions and the newly received state and visual feedback are then appended to the history cache before the next replanning step.

The state and observation readouts are used only for training. During inference, the three modalities are decoded only after the shared physical evolution and do not depend on one another, so no modality-specific noise or sequential prediction is required.

4 Experiments

We evaluate whether the unified physical latent improves generalist robot control across simulated and real-world settings. We first describe the training and evaluation setup, then report results on each benchmark, and finally isolate the contribution of individual modalities and training losses through ablation studies.

4.1 Experimental Setup

Implementation Details. We assemble and curate a pretraining corpus from 56 publicly available data sources, totaling approximately 30,000 hours of robot interaction across 11 embodiments. Each

Table 1: Success rates (%) across generalization levels on LIBERO-X.

Model	Level 1	Level 2	Level 3	Level 4	Level 5	Avg.
π_0 (Black et al., 2025)	29.4	21.9	11.0	7.6	5.1	15.0
X-VLA (Zheng et al., 2026)	30.1	22.6	10.3	6.0	4.1	14.6
GR00T N1.5 (NVIDIA Research, 2025)	43.3	32.9	18.7	13.3	9.7	23.6
$\pi_{0.5}$ (Intelligence et al., 2025)	65.2	53.2	36.0	24.1	18.0	39.3
τ_0-VLA (Cai et al., 2026b)	46.5	34.2	20.5	13.0	10.5	24.9
XM-R0 (Cai et al., 2026a)	71.4	49.6	33.7	22.1	17.2	38.8
OpenWAM (Wang et al., 2026d)	35.9	28.2	21.6	13.0	10.0	21.7
OneRoll	**67.1**	**56.5**	**48.2**	**30.9**	**23.5**	**45.2 (+5.9)**

source retains its native control frequency and is converted to our unified data interface. We compute normalization statistics independently for each source and dimension, using a quantile transform that maps the 1st and 99th percentiles to $[-1, 1]$. Unavailable modalities and dimensions are zeroed and excluded from supervision through validity masks. To balance datasets of different sizes, we apply weighted sampling across data sources and sample contiguous windows within each episode.

Benchmarks and evaluation. We evaluate OneRoll on LIBERO-X (Wang et al., 2026a), EBench (Gao et al., 2026b), and a real-robot manipulation suite, which test multi-level generalization, TableTop manipulation, and physical deployment, respectively. Additional real-to-simulation transfer results are reported in Appendix A.2.

4.2 Simulated Evaluation on LIBERO-X

Setup. LIBERO-X (Wang et al., 2026a) is a public MuJoCo benchmark with a Franka Panda robot and five generalization levels. We post-train on 2,520 demonstrations for 40,000 steps, following the benchmark's protocol. Each level contains its corresponding set of benchmark tasks; success is averaged over evaluation trials and then over tasks, with results reported separately for all five levels. The exact task and trial counts for each level will be added once the evaluation protocol is finalized. The five levels progressively increase the required generalization, allowing us to measure how performance changes as task variation becomes more challenging.

Evaluation Results on LIBERO-X. OneRoll achieves the best average success rate at 45.2%, improving over the strongest baseline by 5.9 points. It reaches 48.2% on Level 3 versus 36.0% for $\pi_{0.5}$ and degrades more gracefully on harder levels, supporting unified physical evolution for compositional generalization. Although XM-R0 is slightly stronger on Level 1, OneRoll leads on Levels 2–5, with its largest margin appearing on Level 3.

4.3 Simulated Evaluation on EBench

Setup. EBench (Gao et al., 2026b) is an Isaac Sim benchmark for generalist mobile manipulation. It contains 26 tasks organized into three families. We evaluate only the seven TableTop tasks listed in Table 2, which cover dexterous and precise single- or dual-arm manipulation.

We restrict the evaluation to TableTop because both our pretraining corpus and our method use only single- and dual-arm manipulation data; they contain no mobile-base tasks. This setting therefore tests whether OneRoll is effective in the manipulation regime represented by its training data, without introducing a separate source of capability from mobile-base control.

Evaluation Results on EBench TableTop. Table 2 reports success rate (SR) and the benchmark partial-credit score for each TableTop task. Each cell is formatted as "SR / Score"; the final pair is the unweighted mean across the seven tasks. The baseline methods and the latest OneRoll run use 20 trials per task.

OneRoll reaches 51.1% mean SR and a mean score of 67.4, achieving the best mean SR and the second-highest mean score behind Qwen-RobotManip. It achieves 99% SR and a perfect score on both frame placement and glass-box placement, and leads on installing gears, peg insertion, and tightening the nut. Its weakest results are on collecting coffee beans and flipping the cup.

Table 2: EBench TableTop results across seven dexterous manipulation tasks.

Model	Coffee beans		Flip cup		Frame		Install gear		Peg in hole		Glass in box		Tighten nut		Average	
	SR	Score	SR	Score	SR	Score	SR	Score	SR	Score	SR	Score	SR	Score	SR	Score
FastWAM (Yuan et al., 2026)	0.0	0.0	0.0	8.0	0.0	10.0	15.0	15.0	0.0	0.0	5.0	10.0	0.0	48.0	2.9	13.0
X-VLA (Zheng et al., 2026)	0.0	2.0	5.0	14.0	30.0	47.0	5.0	13.0	5.0	13.0	15.0	35.0	0.0	45.0	8.6	24.1
InternVLA-A1.5 (Ma et al., 2026)	0.0	16.0	30.0	45.0	0.0	0.0	15.0	20.0	0.0	0.0	0.0	0.0	40.0	70.0	12.1	21.6
Cosmos3-Edge (NVIDIA, 2026)	0.0	4.0	30.0	53.0	20.0	28.0	15.0	18.0	30.0	40.0	25.0	30.0	10.0	48.0	18.6	31.6
GPT-6-Astra-High	0.0	24.0	10.0	44.0	65.0	80.0	15.0	20.0	20.0	**60.0**	20.0	30.0	10.0	55.0	20.0	44.7
$\pi_{0.5}$ (Intelligence et al., 2025)	30.0	55.0	25.0	43.0	45.0	48.0	10.0	13.0	10.0	23.0	65.0	77.0	20.0	58.0	29.3	45.3
π_0 (Black et al., 2025)	20.0	35.0	30.0	35.0	25.0	40.0	30.0	38.0	15.0	43.0	85.0	92.0	25.0	63.0	32.9	49.4
GigaBrain-0.7 (GigaBrain Team et al., 2026)	**50.0**	**71.0**	40.0	61.0	70.0	72.0	20.0	28.0	15.0	55.0	55.0	68.0	15.0	58.0	37.9	59.0
OpenWAM-Alpha (Wang et al., 2026d)	20.0	40.0	40.0	59.0	50.0	60.0	40.0	50.0	40.0	**60.0**	75.0	85.0	30.0	63.0	42.1	59.6
Amapbot	25.0	59.0	55.0	64.0	80.0	82.0	35.0	40.0	15.0	35.0	95.0	98.0	30.0	63.0	47.9	63.0
Qwen-RobotManip (Qwen Team, 2026)	5.0	51.0	**90.0**	**97.0**	95.0	97.0	**50.0**	58.0	10.0	35.0	90.0	95.0	10.0	55.0	50.0	**69.7**
OneRoll	15.0	27.0	0.0	30.1	**99.0**	**100.0**	**50.0**	**73.1**	**45.0**	51.6	**99.0**	**100.0**	**50.0**	**90.3**	**51.1**	67.4

Table 3: Real-robot long-horizon success counts over 10 trials per subtask; each entry reports the number of successful trials out of 10.

Model	Prepare	Contact & shred	Return	Remove shreds	Pour oil	Add potato shreds	Add seasonings	Stir-fry	Serve & reset
$\pi_{0.5}$	8	4	8	5	10	8	6	5	5
OpenWAM	9	6	7	4	9	6	4	2	1
OneRoll	**10**	**8**	**10**	**8**	**10**	**9**	**9**	**8**	**7**

4.4 Real-Robot Evaluation

Setup. We further evaluate whether the pretrained physical representation transfers to real-world manipulation after task-specific post-training. The robot has two 7-DoF arms, each equipped with a wrist camera, together with a head-mounted camera. We evaluate a long-horizon potato-preparation task with nine ordered subtasks. The first four subtasks prepare shredded potatoes: pick up the potato and grater, contact and shred, return the potato and grater, and remove the shreds. The next five subtasks cook and serve them: pour oil, add potato shreds, add seasonings, stir-fry, and serve and reset. Each subtask receives its corresponding language instruction, while the two task segments can also be executed as continuous sequences.



Figure 3: OneRoll executing potato shredding, the first task of our real-robot suite, which spans four subtasks (top to bottom): Prepare, Contact & shred, Return, and Remove shreds. Red arrows indicate arm motion. The whole automatic execution can be found in our Video Material.

Evaluation Results on Real-World. We evaluate each of the nine subtasks for 10 trials and count a trial as successful when the corresponding subtask is completed. Table 3 reports these success counts, allowing the two task segments and their individual failure modes to be compared directly. OneRoll matches or exceeds both baselines on every subtask and produces smoother action sequences during sustained contact and cooking. Figures 3 and 5 show the corresponding sequential rollouts.

Table 4: Ablation studies of input modalities and training losses on LIBERO-X; architecture and continuation variants are summarized in Figure 4.

Variant	Level 1	Level 2	Level 3	Level 4	Level 5	Avg.
w/o obs.	62.4	54.3	43.7	25.6	21.7	41.5
w/o state	61.3	52.6	41.1	24.4	19.1	39.7
action-only	59.2	50.4	38.5	22.1	17.6	37.6
1-frame VLM	64.4	55.3	46.7	29.5	20.5	43.3
w/o $\mathcal{L}_{\text{cons}}$	62.1	52.9	43.2	25.6	19.1	40.6
w/o $\mathcal{L}_{\text{roll}}$	60.1	51.2	39.6	23.0	18.2	38.4
OneRoll	**67.1**	**56.5**	**48.2**	**30.9**	**23.5**	**45.2**

4.5 Ablation Studies

Setup and results. We ablate the architecture, input modalities, and physical training losses on LIBERO-X. The six architecture and continuation variants are summarized by average success rate in Figure 4, while the remaining ablations are reported in Table 4.

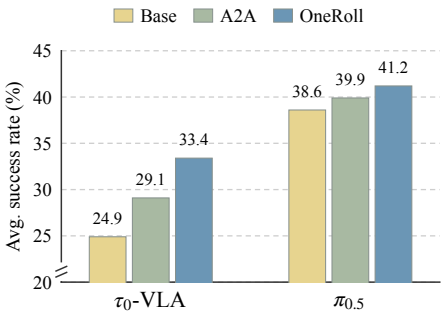

Figure 4: Architecture ablation.

Architecture ablation. Figure 4 compares the native action expert, an action-to-action (A2A) continuation, and our unified physical expert under the same post-training protocol. The results support two conclusions: **1.** A2A continuation improves over the native action expert for both base policies, increasing average success from 24.9% to 29.1% for τ_0-VLA and from 38.6% to 39.9% for $\pi_{0.5}$. **2.** OneRoll further improves over A2A, reaching 33.4% and 41.2%, respectively, which supports the benefit of evolving a unified physical representation beyond action-only continuation.

Input and objective ablations. Table 4 summarizes the effect of removing each input modality, reducing visual context, and omitting individual physical training objectives. The results support three conclusions: **1.** The modalities provide complementary physical information. Removing observation or proprioceptive state reduces average success from 45.2% to 41.5% and 39.7%, respectively, while retaining only action reduces it to 37.6%. **2.** Short temporal context is useful: restricting the VLM to a single frame reduces average success to 43.3%. **3.** Both endpoint alignment and trajectory-level supervision are necessary for stable unified evolution. Removing $\mathcal{L}_{\text{cons}}$ reduces average success to 40.6%, while removing $\mathcal{L}_{\text{roll}}$ reduces it further to 38.4%.

5 Discussion

Conclusion. We presented OneRoll, which evolves actions, states, and observations through a unified physical latent trajectory. A shared vector field extends the realized trajectory, while linear readouts recover each modality. Strong results in simulation and real-world manipulation support this formulation. Ablations suggest that trajectory continuity and multimodal unification provide complementary gains: the unified physical expert improves upon action-to-action continuation across both evaluated base policies, and state and observation supervision further support action generation. These findings highlight the value of learning shared physical dynamics and identify the choice of what a model evolves as a key architectural consideration. Unified physical evolution thus offers a promising design principle for larger-scale embodied foundation models, motivating further study with broader experience, more diverse embodiments, and longer temporal horizons.

Limitation. Our experiments use a single approximately 3B-parameter instantiation and do not yet establish how the unified physical flow scales with backbone or expert capacity. In addition, future visual latents are supervised in the feature space of a frozen pretrained vision encoder and fixed PCA basis, which may limit task-specific physical prediction and adaptation to new embodiments; future work could jointly adapt the visual encoder with a slowly updated EMA target encoder to let the representation evolve while keeping the prediction target stable.

AI USE STATEMENT

Generative AI tools assisted with limited writing and software engineering, including grammar, clarity, LaTeX formatting, and routine code debugging. They were not used to generate the hypothesis, method, comparisons, data, reported numbers, or conclusions. The authors checked all claims and AI-assisted text, tested suggested code, and remain responsible for the complete manuscript.

ETHICS STATEMENT

Our pretraining corpus uses publicly available robot-interaction sources, and the real-robot experiments use episodes collected on our own platform. The study involves no human participants, personal information, or sensitive data. Tasks use ordinary household objects in a controlled laboratory and are not presented as unsupervised deployment. Hardware limits, workspace supervision, and emergency stops are required because failures can cause damage or unsafe motion. We release no private data and follow the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

Section 3 specifies the unified physical-token construction, modality readouts, vector-field integration, and training objectives. Appendix A.4 documents the data interface, preprocessing, model dimensions, history handling, optimizer, schedule, and integration steps; Section 4 and Appendix A.1–A.3 define the evaluation protocols and task instructions. The benchmarks and pretraining sources are public, with published baselines identified by reference. Upon acceptance, we will release code, configurations, checkpoints, data interfaces, table and figure scripts, evaluation settings, and random seeds.

REFERENCES

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, et al. π_0: A Vision-Language-Action Flow Model for General Robot Control. In *Robotics: Science and Systems (RSS)*, 2025. URL https://arxiv.org/abs/2410.24164.

Rui Cai, Jun Guo, Xinze He, Piaopiao Jin, et al. Xiaomi-robotics-0: An open-sourced vision-language-action model with real-time execution, 2026a. URL https://arxiv.org/abs/2602.12684.

Xiaowei Cai, Yunuo Cai, Bingao Chen, Jingxiao Chen, et al. τ_0-VLA: a Hierarchical Robot Foundation Model with World-Model-Guided Test-Time Computation, 2026b. URL https://arxiv.org/abs/2608.16885.

Meipo Dai, Qiyuan Zhuang, He-Yang Xu, Ying-Jie Shuai, Yijun Wang, Qi Dou, and Xiu-Shen Wei. Where should action generation begin? a learnable source prior for generative robot policies, 2026. URL https://arxiv.org/abs/2606.17408.

Dechen Gao, Boqi Zhao, Andrew Lee, Ian Chuang, Hanchu Zhou, Hang Wang, Zhe Zhao, Junshan Zhang, and Iman Soltani. Vita: Vision-to-action flow matching policy. In *International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=BTe5VLBjPg.

Ning Gao, Jinliang Zheng, Xing Gao, Haoxiang Ma, Hanqing Wang, et al. EBench: Elemental diagnosis of generalist mobile manipulation policies, 2026b. URL https://arxiv.org/abs/2606.18239.

Gemini Robotics Team, Abbas Abdolmaleki, Saminda Abeyruwan, Joshua Ainslie, et al. Gemini Robotics 1.5: Pushing the frontier of generalist robots with advanced embodied reasoning, thinking, and motion transfer, 2025. URL https://arxiv.org/abs/2510.03342.

GigaBrain Team, Angen Ye, Axiang Sun, Can Jin, et al. GigaBrain-0.7: Scaling embodied foundation models to emergent capabilities with a three-system architecture, 2026. URL https://arxiv.org/abs/2608.15875.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, et al. $\pi_{0.5}$: a Vision-Language-Action Model with Open-World Generalization, 2025. URL https://arxiv.org/abs/2504.16054.

Jindou Jia, Gen Li, Xiangyu Chen, Tuo An, Yuxuan Hu, Jingliang Li, Xinying Guo, and Jianfei Yang. Action-to-action flow matching. In *Robotics: Science and Systems (RSS)*, 2026. URL https://arxiv.org/abs/2602.07322.

Sinjae Kang, Chanyoung Kim, Kaixin Wang, Li Zhao, and Kimin Lee. Warmprior: Straightening flow-matching policies with temporal priors, 2026. URL https://arxiv.org/abs/2605.13959.

Jinhao Li, Yuxuan Cong, Yingqiao Wang, Hao Xia, Shan Huang, Yijia Zhang, Ningyi Xu, and Guohao Dai. Step: Warm-started visuomotor policies with spatiotemporal consistency prediction, 2026. URL https://arxiv.org/abs/2602.08245.

Dongxiu Liu, Haoyi Niu, Peng Cheng, Yuan Gao, Xirui Kang, Sangli Teng, Koushil Sreenath, and Xianyuan Zhan. Odeworld: A continuous predictive architecture via physical-time flow, 2026. URL https://arxiv.org/abs/2607.27924.

Haoxiang Ma, Junhao Cai, Xiaoxu Xu, Hao Li, Yuyin Yang, Yang Tian, Jiafei Cao, Hongrui Zhu, Zherui Qiu, Zhaxizhuoma, Yuqiang Yang, Jiaqi Peng, Xueyuan Wei, Yangkun Zhu, Jiahao Jiang, Xing Gao, Hanqing Wang, Feng Yuan, Kailin Li, Xueyue Zhu, Tai Wang, Yan Ding, Jiangmiao Pang, Jia Zeng, Jingjing Zhang, Bowen Zhou, Yao Mu, Chunhua Shen, and Weinan Zhang. InternVLA-A1.5: Unifying understanding, latent foresight, and action for compositional generalization, 2026. URL https://arxiv.org/abs/2607.04988.

NVIDIA. Cosmos 3: Omnimodal world models for physical AI, 2026. URL https://arxiv.org/abs/2606.02800.

NVIDIA GEAR Team. GR00T N1.6: An improved open foundation model for generalist humanoid robots, 2025. URL https://research.nvidia.com/labs/gear/gr00t-n1_6/.

NVIDIA Research. Gr00t n1.5: An improved open foundation model for generalist humanoid robots, 2025. URL https://research.nvidia.com/labs/gear/gr00t-n1_5/.

Bikang Pan, Fan Liu, Haotao Lu, Jingya Wang, and Ye Shi. Selfwam: A self-grounded unified world action model for fast robot control, 2026. URL https://arxiv.org/abs/2608.00725.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient action tokenization for vision-language-action models. In *Robotics: Science and Systems (RSS)*, 2025. URL https://arxiv.org/abs/2501.09747.

Qwen Team. Qwen-RobotManip technical report: Alignment unlocks scale for robotic manipulation foundation models, 2026. URL https://arxiv.org/abs/2606.17846.

Jonas Scholz and Richard E. Turner. Warm starts accelerate conditional diffusion, 2025. URL https://arxiv.org/abs/2507.09212.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, Simon Alibert, Matthieu Cord, Thomas Wolf, and Remi Cadene. SmolVLA: A vision-language-action model for affordable and efficient robotics, 2025. URL https://arxiv.org/abs/2506.01844.

Guodong Wang, Chenkai Zhang, Qingjie Liu, Jinjin Zhang, Jiancheng Cai, Junjie Liu, and Xinmin Liu. LIBERO-X: Robustness litmus for vision-language-action models. In *Robotics: Science and Systems (RSS)*, 2026a. URL https://arxiv.org/abs/2602.06556.

Jiaxu Wang, Yicheng Jiang, Tianlun He, Jingkai Sun, Qiang Zhang, Junhao He, Jiahang Cao, Zesen Gan, Mingyuan Sun, Qiming Shao, and Xiangyu Yue. Mvista-4d: View-consistent 4d world model with test-time action inference for robotic manipulation. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026b. URL https://arxiv.org/abs/2602.09878.

Kuanning Wang, Ke Fan, Chenhao Qiu, Zeyu Shangguan, Yuqian Fu, Yanwei Fu, Daniel Seita, and Xiangyang Xue. Oflow: Injecting object-aware temporal flow matching for robust robotic manipulation, 2026c. URL https://arxiv.org/abs/2604.17876.

Yuran Wang, Siqiao Huang, Mingleyang Li, Chenhao Zhang, et al. Openwam: An open, modular exploration towards systematic world-action model pretraining, 2026d. URL https://arxiv.org/abs/2609.07398.

Jenai Xuning Yang, Rishit Dagli, Alex Zook, Hugo Hadfield, Ankit Goyal, Stan Birchfield, Fabio Ramos, and Jonathan Tremblay. RoboLab: A high-fidelity simulation benchmark for analysis of task generalist policies, 2026. URL https://arxiv.org/abs/2604.09860.

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. URL https://arxiv.org/abs/2602.15922.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-WAM: Do world action models need test-time future imagination?, 2026. URL https://arxiv.org/abs/2603.16666.

Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, XinQiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, He Wang, Zhizheng Zhang, Li Yi, Wenjun Zeng, and Xin Jin. DreamVLA: A vision-language-action model dreamed with comprehensive world knowledge. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-0818. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/22d4f952efa13970f0b1ffb22170d416-Abstract-Conference.html.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, et al. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2510.10274.

Pengfei Zhou, Shengcong Chen, Di Chen, Jiaxu Wang, Rongjun Jin, Bingwen Zhu, Yike Pan, Songen Gu, Kuanning Wang, Shufeng Nan, Xingyu Qiu, Chenhao Qiu, Pu Yang, Yunuo Cai, Jianxiong Gao, Yifan Li, Yanwei Fu, Xiangyu Yue, Zhi Chen, and Jianlan Luo. τ_0-WM: A unified video-action world model for robotic manipulation, 2026. URL https://arxiv.org/abs/2606.01027.

A Appendix

This appendix provides additional details for reproducibility and analysis. We describe the implementation of the unified physical latent and its integration procedure, the composition and preprocessing of the training data, the benchmark-specific evaluation protocols, and additional ablations and qualitative results.

A.1 Real-Robot Task Decomposition and Language Instructions

The potato-shredding segment of the real-robot suite is decomposed into four ordered subtasks. Each English sentence below is used verbatim as the language instruction for the corresponding subtask.

1. **Pick up a potato and the grater**. *Pick up a potato with the left gripper and the grater with the right gripper.*
2. **Contact and shred**. *Adjust the potato against the grater, then repeatedly grate it so the shreds fall into the water.*
3. **Return the potato and grater**. *Return the potato and grater to their designated locations.*
4. **Remove the shreds**. *Use the strainer to lift the potato shreds out of the water and transfer them into the bowl.*

The cooking segment then consists of the following five ordered subtasks, with the same convention for language instructions.

1. **Pour oil**. *Pick up the oil bottle, pour oil into the pan, and put the oil bottle back.*
2. **Add potato shreds**. *Pick up the bowl of potato shreds, pour the shreds into the pan, and put the empty bowl back.*
3. **Add seasonings**. *Pick up the seasoning container, add the seasonings to the pan, and put the seasoning container back.*
4. **Stir-fry**. *Pick up the spatula and stir-fry the potato shreds in the pan.*
5. **Serve and reset**. *Transfer the potato shreds from the pan to the plate, then return the pan and spatula to their designated locations.*



Figure 5: OneRoll executing potato cooking, the second task of our real-robot suite, which spans five subtasks (top to bottom): Pour oil, Add potato shreds, Add seasonings, Stir-fry, and Serve & reset. Red arrows indicate arm motion.

A.2 RoboLab-120 Evaluation

Setup. RoboLab-120 (Yang et al., 2026) is a public IsaacSim benchmark containing 120 manipulation tasks. We post-train on real DROID trajectories for 120,000 steps and evaluate in the unseen simulator using a Franka arm with a Robotiq 2F-85 gripper. We report the mean success rate across all 120 tasks and 1,200 evaluation episodes. The benchmark groups its episodes into 640 simple, 390 moderate, and 170 complex tasks.

Evaluation Results on RoboLab-120. Table 5 reports success rate (SR) and the benchmark score at each difficulty level. Model parameter counts follow the RoboLab leaderboard when available.

Table 5: RoboLab-120 results across task difficulty levels. SR denotes success rate (%).

Model	Params.	Simple		Moderate		Complex		Overall	
		SR (%)	Score	SR (%)	Score	SR (%)	Score	SR (%)	Score
GR00T N1.6 (NVIDIA GEAR Team, 2025)	3B	8.8	13.6	7.9	25.1	0.0	11.9	7.2	17.1
π_0-FAST (Pertsch et al., 2025)	3B	20.2	26.7	13.3	29.2	2.9	21.9	15.5	26.9
Cosmos3-Edge (NVIDIA, 2026)	4B	25.6	–	23.3	–	11.8	–	22.9	–
DreamZero (Ye et al., 2026)	14B	26.1	35.5	30.0	48.4	14.1	35.9	25.7	39.8
$\pi_{0.5}$ (Intelligence et al., 2025)	3.3B	29.7	38.9	31.5	50.5	13.5	44.1	28.0	43.4
BiMind v0.1 (Yang et al., 2026)	7B	38.8	46.2	32.8	50.9	14.1	42.5	33.3	47.2
Phoenix (Yang et al., 2026)	–	**42.8**	**51.9**	25.6	40.0	22.9	37.0	34.4	45.9
Cosmos3-Nano (NVIDIA, 2026)	16B	40.6	51.0	35.4	52.3	**25.3**	**54.4**	**36.8**	**51.9**
OneRoll	**3B**	39.6	49.9	**37.6**	**55.2**	19.0	50.4	36.0	51.7

OneRoll obtains 36.0% overall success and a score of 51.7, with its strongest relative performance on moderate tasks (37.6% SR and 55.2 score). It is competitive with the larger policies despite using 3B parameters.

Competency-axis results. Table 6 reports the RoboLab leaderboard (Yang et al., 2026) results broken down by relational, visual, and procedural competency. Each cell reports SR and score. Because tasks may belong to multiple competency axes, these axis-specific values are not additive and do not determine the all-task overall result, which is reported in Table 5.

Table 6: RoboLab competency-axis results. SR denotes success rate (%).

Model	Params.	Relational		Visual		Procedural		Overall	
		SR (%)	Score	SR (%)	Score	SR (%)	Score	SR (%)	Score
GR00T N1.6	3B	10.0	23.2	5.0	15.1	2.7	11.1	7.2	17.1
π_0-FAST	3B	23.3	36.7	8.8	19.5	6.4	16.4	15.5	26.9
Cosmos3-Edge	4B	26.7	–	16.8	–	15.0	–	22.9	–
DreamZero	14B	33.3	45.5	17.5	31.9	26.4	42.3	25.7	39.8
$\pi_{0.5}$	3.3B	33.8	46.8	23.8	38.8	21.8	41.6	28.0	43.4
BiMind v0.1	7B	35.7	51.7	29.4	43.0	21.4	33.3	33.3	47.2
Phoenix	–	25.0	37.3	**33.6**	**45.4**	21.8	30.3	34.4	45.9
Cosmos3-Nano	16B	**49.3**	**60.9**	26.1	42.9	**27.3**	**46.8**	**36.8**	**51.9**
OneRoll	**3B**	47.1	59.2	26.6	43.7	25.2	44.3	36.0	51.7

A.3 Evaluation Protocols

This section will provide the task lists, number of trials, post-training settings, and success criteria for LIBERO-X, RoboLab-120, and the real-robot suite.

A.4 Training Data and Optimization Details

This section provides the data-source composition, preprocessing choices, model configuration, and optimization hyperparameters. The detailed model configuration and optimization settings are as follows. Our main model uses a Qwen3.5-VL-2B backbone and a layer-aligned physical expert with approximately 1B parameters. The VLM is initialized from a pretrained checkpoint, while the physical expert is trained from scratch. We set the physical latent width to $d = 1{,}536$ and the mini-chunk length to $H = 10$. The learned physical velocity field is integrated using $N = 10$ Euler steps.

Visual targets are 1024-D latents obtained by applying the frozen visual encoder and a fixed PCA basis to 2×2-pooled features of the main third-person view; the targets are latent differences with respect to the current frame rather than absolute latents. Rather than resizing every view to a fixed resolution, each source is rendered onto its own canvas under a fixed budget of 90,112 pixels that preserves the native aspect ratio (nominal 256×320), so no view is stretched. We represent actions in a unified 40-D space covering dual-arm end-effector poses, grippers, waist and base motion, and reserved joint-control slots, expressed as increments relative to the measured pose at the start of each mini-chunk. Proprioception is represented by a 23-dimensional vector containing joint and gripper states, waist and head states, and odometry. To keep the policy from locking up at deployment time, where no motion history exists, we apply history dropout during training: with probability 0.05, the entire conditioning history is replaced by its just-reset form, i.e., all history frames and states are set to the current observation and the previous action block is set to identity rows. We train with AdamW ($\beta = (0.9, 0.95)$, weight decay 0.05) and gradient clipping at 1.0. Pretraining runs for 600,000 steps on 208 GPUs with a per-GPU batch size of 8, giving a global batch size of 1,664. We use cosine learning-rate decay from 5×10^{-5} to 6×10^{-6} without warmup, a $0.5\times$ learning-rate multiplier on the visual encoder, bfloat16 mixed precision, and fully sharded data parallelism.

A.5 Additional Method Details

Token layout and multimodal projection. We use a mini-chunk length of $H = 10$. Thus, one action mini-chunk is represented by 10 consecutive action tokens, rather than by a single token for the whole chunk. For a batch of size B, the action input has shape $B \times H \times d_a$, where $d_a = 40$ in the unified action interface. Each time row is projected independently into the expert latent width, while the expert's positional encoding preserves the ordering of the 10 rows. We do not flatten the action sequence over time before projection.

Let a_i denote the action row at time index i. The action encoder produces an action token

$$e_i^a = E_a(a_i), \qquad i = 1, \ldots, H, \tag{13}$$

where E_a is a learned row-wise projection. State values and their validity mask are concatenated before projection and added to the corresponding action token:

$$e_i^{a+s} = E_a(a_i) + E_s([s_i, m_i]). \tag{14}$$

The state adapter therefore covers all H time rows and allows missing or invalid state channels to be excluded by the mask m_i.

The auxiliary visual path is temporally sparse. The visual tower first produces spatial tokens for each frame. The spatial grid is pooled and flattened to one feature vector per selected frame, which is then mapped by a fixed visual basis and a learned adapter into the expert latent width. With the current stride of six and $H = 10$, the auxiliary visual feature is written at the sixth action position (zero-based code index 5). The language tokens and the main visual context remain on the native Qwen text/vision path; they are not concatenated with the 10 action rows into one flattened vector. Consequently, only the per-frame auxiliary visual feature is spatially pooled and flattened, whereas action and state retain their temporal row structure.

Encoding, velocity, and readout matrices. The OneRoll action pathway contains three distinct learned maps: an action input encoder, a latent velocity output, and a latent-to-action readout. These maps are not tied as transposes and are not constrained to be orthogonal. During initialization, the implementation performs an analytic parameter construction. In particular, the initial action readout is constructed from the Moore–Penrose pseudoinverse of the action input matrix, with a bias correction so that encoding followed immediately by readout is as close as possible to the identity. The initial velocity map is formed by combining the action input map with the action-output map. These relations define the starting point only: after initialization, all matrices remain ordinary trainable parameters.

The state encoder, state readout, visual adapter, and visual readout are independent matrices. They do not share transpose relations with the action maps. State and auxiliary visual channels, together with their corresponding readouts, are zero-initialized so that introducing these paths does not cause a discontinuous change in the initial action behavior. The fixed visual basis is computed before training (using the prescribed PCA basis or fixed-seed normalized random basis); it is not the transpose

of an action readout. No orthogonal constraint, spectral normalization, or weight normalization is applied to these matrices. Action normalization and validity masks are handled at the data and loss levels, while parameter stability is obtained through the initialization, optimization objective, and gradient clipping.

History–future interpolation. Each pretraining example contains two temporally adjacent action blocks. The history block consists of the already executed actions before the decision time and provides a warm start; the future block consists of the target actions after that time. Each block is relativized to its own block-initial state and then normalized. Let z_h and z_f be their latent encodings. The training point at time τ is the linear interpolation

$$z_\tau = (1-\tau)z_h + \tau z_f, \qquad u^\star = z_f - z_h, \tag{15}$$

and the OneRoll flow objective trains the model velocity toward $u^\star$.

The interpolation time is sampled once per example and shared by all 10 tokens. Specifically, τ is drawn from $\mathrm{Beta}(1.5, 1.0)$ and clipped to a range that stays close to, but does not include, the two endpoints. The pretraining path does not add additional Gaussian noise to the action blocks. Besides the random time, the main warm-start variation is `fresh_start_p=0.05`: for a small fraction of examples, the history is replaced by a reset-like stationary history, matching the absence of motion history at deployment. Training and inference use the same latent encoding, Euler integration, and modality readouts.

Losses and differentiable rollout. The configured objective is logged as six scalar terms:

$$\mathcal{L} = \mathcal{L}_{\text{flow}} + 0.5\mathcal{L}_{\text{enc}} + 0.5\mathcal{L}_{\text{roll}} + \mathcal{L}_{\text{cons}} + 0.5\mathcal{L}_{\text{state}} + 0.5\mathcal{L}_{\text{obs}}. \tag{16}$$

Here, $\mathcal{L}_{\text{flow}}$ supervises the latent velocity, $\mathcal{L}_{\text{enc}}$ checks reconstruction of the future action from the future latent, $\mathcal{L}_{\text{roll}}$ checks action reconstruction after integrating from the history latent, and $\mathcal{L}_{\text{cons}}$ aligns the integrated endpoint with the future latent target. The state and observation terms supervise the corresponding readouts of the future target and the rollout endpoint. Code comments refer to eight losses because state and observation supervision are each evaluated at both endpoints before being aggregated into the two scalar terms shown above.

The rollout uses $N = 10$ Euler steps. Here $H = 10$ denotes the temporal rows in one action mini-chunk, whereas N denotes the numerical integration steps; the two settings describe different parts of the computation. At every step, the latent velocity is recomputed. Rollout losses are fully differentiable: the computation graph is retained through all Euler steps, the latent readouts, and the prefix key/value conditioning used by the model. Only deployment-time inference uses a no-gradient path. Action dimensions, time rows, state channels, auxiliary visual targets, and padded samples are weighted by their respective validity masks, so padding is never treated as physical supervision.

Deployment interface. The pretrained model outputs one 10-row action chunk. The deployment adapter executes a configured prefix of this chunk before replanning. A modified replanning interval may send a different number of rows, but it cannot exceed the model's 10-row output and does not change the pretraining chunk definition.

At the next decision, image, pose, gripper, and state history comes from the environment's actual observations. The action history contains only setpoints that were actually sent to the environment; predicted but unexecuted rows are discarded. The relative-action anchor is recomputed from the current measured state. At first launch, the history cache is cleared. Missing visual and state history is filled by repeating the current observation, while missing action history is represented by identity, zero-increment rows derived from the measured pose. This gives the first inference the same reset-like form as the training examples sampled with `fresh_start_p`.