μ_0: A Scalable 3D Interaction-Trace World Model

Seungjae Lee[1*], Yoonkyo Jung[1*], Jusuk Lee[2], Jonghun Shin[2], Amir Hossein Shahidzadeh[1], Yao-Chih Lee[1], H. Jin Kim[2], Jia-Bin Huang[1†], Furong Huang[1†]

[1]University of Maryland, College Park [2]Seoul National University

World models that capture how actions induce physical change enable scalable robot learning without reliance on embodiment-specific action labels. Pixel-space video models provide broad visual priors but expend model capacity on dense appearance reconstruction, while direct action models require embodiment-specific labels that hinder scalability. We present μ_0, a scalable world model based on 3D traces. Rather than predicting dense pixels or directly modeling actions, μ_0 forecasts smooth 3D trajectories for salient interaction points such as objects, tools, hands, and contact regions, yielding a compact, embodiment-agnostic motion interface. To enable training from diverse video sources, our TraceExtract system automatically extracts 3D supervision by selecting keypoints, constructing globally aligned traces, and associating motion segments with hierarchical language captions. This TraceExtract supervision pretrains μ_0 by combining a pretrained vision-language backbone with a modular trace expert, which represents each query via B-spline control points and predicts future traces. Experiments show that μ_0 outperforms baselines in both 2D and 3D trace prediction, including trace prediction models and tokenized VLM methods. Because μ_0 is frozen and reusable, it can be paired with action experts for downstream robot embodiments. Despite action-free pretraining, the resulting trace-conditioned policies achieve performance competitive with VLA models pretrained with action supervision, such as π_0. These results establish 3D traces as a scalable and transferable representation for cross-embodiment manipulation. Project page: https://mu0-wm.github.io/.

Keywords: world model, 3D interaction trace, robot manipulation

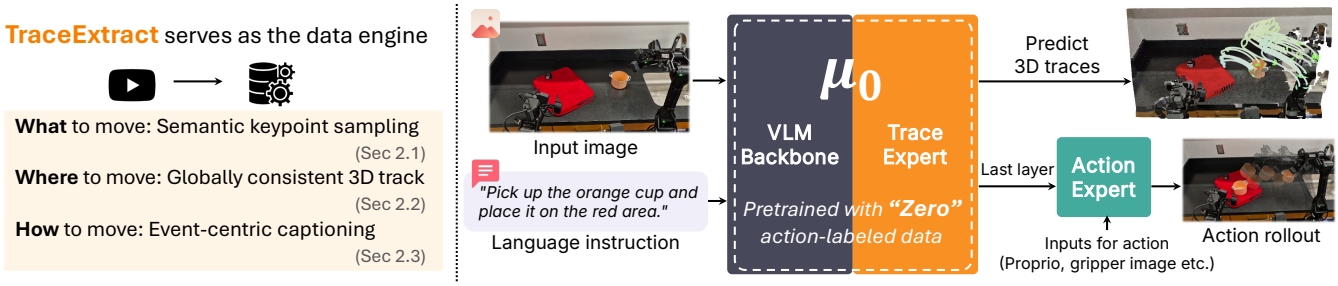

Figure 1: From videos to reusable action priors. TraceExtract extracts event-captioned 3D interaction traces from heterogeneous videos by selecting entity-centric keypoints, lifting them into globally aligned 3D, and pairing motion events with language. This supervision pretrains μ_0 as a world model that predicts compact future trajectories for interaction points, instead of dense pixels or robot-specific actions. Once pretrained, the frozen μ_0 can be reused with any downstream action expert, which consumes trace features to produce executable robot action chunks.

1. Introduction

Robot learning is constrained by a fundamental data paradox. On one hand, videos provide an abundant and scalable source of physical behavior data. On the other hand, the most useful kind of supervision for control, action-labeled robot data, is scarce, expensive, hardware-specific, and incompatible across embodiments. World models offer a path around this bottleneck by learning from observation-rich video data and later grounding their predictions to specific robot embodiments (Lin et al., 2026, Gao et al., 2026, Cho et al.,

*Equal contribution †Equal advising

2026, Wang et al., 2026a, Kim et al., 2026, Wang et al., 2026b). The key question is *what such a model should predict*. Pixel-space video generation is scalable but expends model capacity on dense appearance and background reconstruction, while often failing to capture the metric geometry, contact structure, and occlusion patterns required for manipulation (Du et al., 2023, Hu et al., 2025b, Agarwal et al., 2025). Direct action prediction, as in Vision-Language-Action models, remains limited by the scarcity and embodiment specificity of labeled robot demonstrations. We instead occupy the middle ground: 3D traces of semantic interaction points—object parts, tools, hands, and contact regions—which compactly describe what must move regardless of the robot used.

Recent motion-centric methods point in this direction through 2D flows (Wen et al., 2024, Xu et al., 2024, Nguyen et al., 2026), 3D flows (Zhi et al., 2025, Huang et al., 2026, Wang et al., 2026c, Hung et al., 2026, Lee et al., 2026), and object trajectories (Bharadhwaj et al., 2024). However, existing systems share three limitations: 1) they under-sample small but task-critical regions such as tool tips and contact patches; 2) they conflate object motion with camera motion by operating in local or 2D image-space coordinates; and 3) they pair long demonstrations with episode-level captions rather than event-level intent. These gaps motivate a trace world model that (i) selects where to measure motion, (ii) preserves global 3D structure, and (iii) binds local motion segments to language. The closest prior work, TraceGen (Lee et al., 2026), predicts 3D traces on a fixed grid with episode-level captions and depth-conditioned input, and is therefore limited along all three axes; our system addresses these limitations.

We present μ_0, a query-conditioned 3D trace-space world model that serves as a reusable motion prior for downstream action experts (Fig. 1). To supply training data at scale, we introduce TraceExtract, a scalable data engine that converts heterogeneous human and robot videos into event-captioned trace supervision by (i) selecting semantic keypoints via DINOv2 entity clusters, (ii) lifting them into globally aligned 3D, and (iii) captioning trace-driven motion events with hierarchical language—scaling trace curation by roughly $8\times$ over prior 3D trace datasets (Lee et al., 2026). μ_0 is built on a pretrained VLM backbone augmented with a permutation-equivariant Trace Expert, which forecasts flexible semantic keypoints as smooth B-spline traces using a semantic flow-matching objective. After video-only pretraining, the frozen μ_0 becomes a reusable motion prior: an Action Expert attending to its trace-denoising features, along with robot observations, proprioception, and language, outputs executable action chunks for any target embodiment. On 2D/3D trace forecasting, μ_0 outperforms prior trace prediction models and tokenized-VLM baselines. In 8 RoboCasa365 simulation (Nasiriany et al., 2026) tasks and 3 real-world UR3 manipulation tasks, μ_0 matches or exceeds action-labeled VLAs (π_0 (Black et al., 2025), $\pi_{0.5}$ (Intelligence et al., 2025)), achieving 120–130% of π_0's and 70–115% of $\pi_{0.5}$'s average success rates, despite using no action supervision during pretraining.

Our **main contributions** are: **(1) TraceExtract**, a scalable data engine that extracts event-captioned 3D trace supervision from heterogeneous manipulation videos via semantic keypoint selection, globally aligned 3D lifting, and hierarchical language captioning. **(2) $\boldsymbol{\mu_0}$**, a query-conditioned 3D trace-space world model with a VLM backbone, permutation-equivariant Trace Expert, B-spline trace targets, and semantic flow-matching training. **(3) Trace-conditioned action adaptation**, which freezes the pretrained μ_0 and trains an action expert on top of its trace-denoising features, enabling action-free video pretraining to transfer to effective robot policies.

2. TraceExtract: A Scalable Cross-Embodiment Data Pipeline

Measurement target. A trace-space world model must decide *where* to measure motion. Dense pixels are redundant and background-heavy, while uniform grids waste queries on static surfaces and can miss small manipulated parts. Thus, we can predict **interaction-centric keypoints** on *objects*, *tools*, *hands*, and

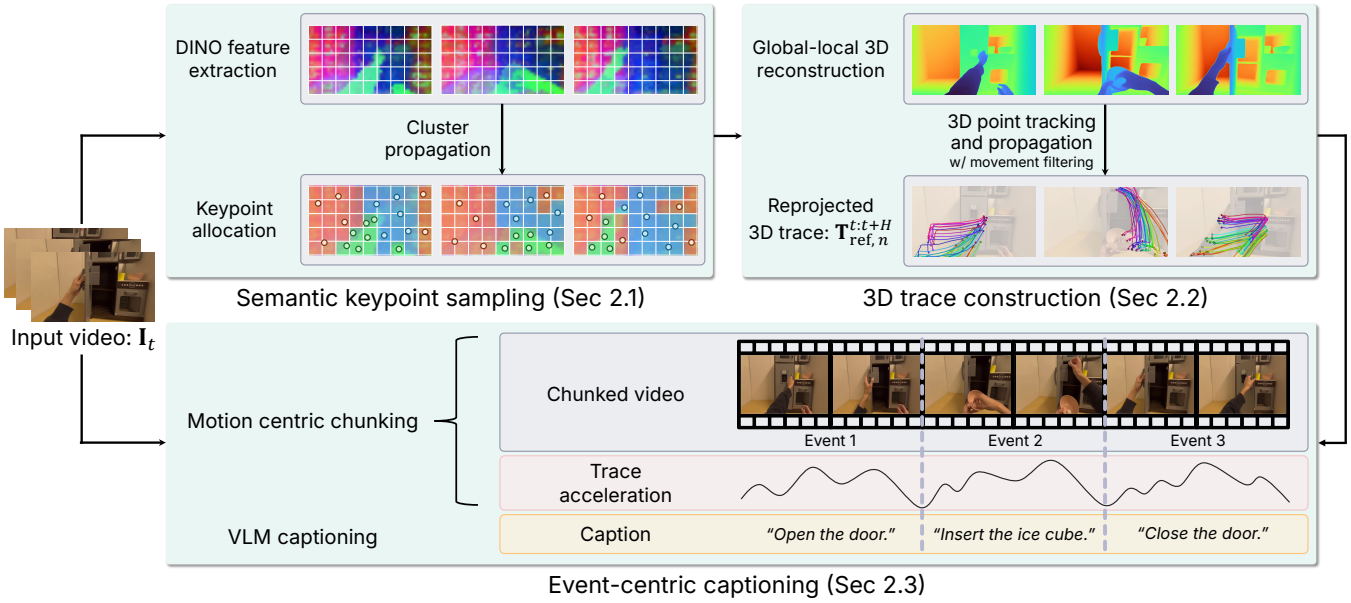

Figure 2: Overview of TraceExtract. From an uncurated human or robot manipulation video, TraceExtract selects DINOv2 entity keypoints (Sec. 2.1), tracks and lifts them into globally aligned 3D traces with chunk-wise reconstruction (Sec. 2.2), and segments traces into motion-centric events for hierarchical VLM captioning (Sec. 2.3), producing event-captioned 3D trace supervision for μ_0.

contact regions; their 3D motion captures what changes and what a robot should reproduce. These traces are embodiment agnostic—the same object motion can guide different robot morphologies—but only when they provide **(1) semantic selection**, so keypoints lie on task-relevant entities; **(2) consistent 3D tracking**, so identities survive camera motion and long horizons; and **(3) event-level language**, so local motion segments are paired with the right skill descriptions.

Pipeline overview. We introduce TraceExtract, the data engine used to train μ_0. It treats trace extraction as interaction-centric supervision and, building on TraceGen (Lee et al., 2026), remedies fixed-grid, short-clip trace curation with **(1) task-relevant keypoints**, **(2) globally consistent 3D identities**, and **(3) language aligned to motion events**. These properties let TraceExtract scale curation by producing {observation, trace, language} triplets for training μ_0 (Sec. 3).

2.1. Semantic Keypoint Sampling

Prior fixed-grid trace extraction (Lee et al., 2026) is simple but area-biased: *(1) backgrounds* can dominate the point budget, *(2) small objects* may receive too few points, and *(3) contact patches or tool tips* can be missed. As shown in Fig. 2, TraceExtract instead *(1) extracts* DINOv2 (Oquab et al., 2024) patch features and clusters them into entity-level groups, *(2) propagates* these entity identities throughout the clip, and *(3) allocates* a fixed keypoint budget per entity and selects spatially diverse points on each entity's high-visibility frames (Appendix A.1). The result is a compact query set focused on action-informative entities; a movement filter further marks static or background-dominated tracks so non-moving points do not overwhelm the interaction signal (Appendix A.2).

2.2. 3D Trace Construction

Global–local reconstruction. After keypoint selection, TraceExtract must preserve each query's identity and 3D position across long videos despite *(1) egocentric camera motion*, *(2) objects entering or leaving the scene*, and *(3) memory limits of full-video reconstruction*. The middle stage of Fig. 2 addresses these issues with global–local reconstruction: it *(1) uses sparse anchor frames* to establish a shared global coordinate frame, *(2) reconstructs dense local chunks* and aligns them back to that frame, *(3) tracks sampled keypoints* in the common 3D space, and *(4) propagates tracks* across chunk boundaries using each point's last valid world-space position.

Reference-frame traces. We reproject the tracks into a per-chunk reference camera to obtain screen-aligned 3D traces $\boldsymbol{T}_{\text{ref},n}^{t:t+H} = [x_{n,i}, y_{n,i}, z_{n,i}]_{i=t}^{t+H}$ for each query n. This representation *(1) removes camera motion* and *(2) retains image alignment* for the visual backbone (Sec. 3.1). We further normalize trace speed by arc-length reparameterization, reducing duration differences between human and robot demonstrations. Details are provided in Appendix A.3 and Appendix A.4.

2.3. Event-Centric Captioning

Motion-centric chunking. Long demonstrations need language at multiple resolutions: episode captions miss local subgoals, while frame-level captions are expensive and noisy. TraceExtract therefore uses traces to define captioning units. In the final stage of Fig. 2, we smooth per-frame trace acceleration a_t into $\tilde{a}_t$ with a Savitzky–Golay filter (Savitzky and Golay, 1964), identify action anchors as prominent peaks p_i, and place chunk boundaries at the lowest-acceleration valleys, $b_i = \arg\min_{t\in[p_i,p_{i+1}]} \tilde{a}_t$. This creates short motion-centric events that *(1) limit VLM context length* and *(2) align chunks* with subgoals such as reaching, grasping, moving, and releasing.

Hierarchical VLM captioning. For each chunk, the VLM produces captions from *(1) the start frame*, *(2) the midpoint frame*, and *(3) the end frame*, optionally conditioned on a motion mask and an episode-level task description when available. A text-only LLM then merges adjacent captions over sliding windows, yielding both fine-grained captions and coarser task summaries (Appendix A.5).

2.4. Trace Supervision Interface

Combining semantic queries, reference-frame traces, and event captions, TraceExtract converts each video into tuples $\mathcal{D}_{\text{TE}} = \left\{ \left(I_t, l_c, \boldsymbol{Q}_t, \boldsymbol{T}_{\text{ref}}^{t-h:t+H} \right) \right\}$, where I_t is the observation, l_c is the event or merged task caption, $\boldsymbol{Q}_t = \{\boldsymbol{q}_n^t\}_{n=1}^N$ is the query-keypoint set selected at first visibility or carried from history, and $\boldsymbol{T}_{\text{ref}}^{t-h:t+H}$ contains past and future 3D traces in the reference camera. Then, μ_0 trained on these tuples learns the prediction map $\mu_0 : \left(I_t, l_c, \boldsymbol{Q}_t, \boldsymbol{T}_{\text{ref}}^{t-h:t} \right) \mapsto \hat{\boldsymbol{T}}_{\text{ref}}^{t:t+H}$, which predicts the future motion of the interaction-centric query set.

3. μ_0: Query-Conditioned Trace World Model

Overview. Using the tuples produced by TraceExtract, μ_0 learns a query-conditioned dynamics model rather than a pixel generator. It predicts how interaction-centric keypoints move in 3D from *(1) observation*, *(2) language instruction*, and *(3) optional keypoint history*. This formulation must resolve three coupled challenges: *(1) semantic–metric fusion*, retaining large vision-language priors while adding metric 3D reasoning; *(2) query equivariance*, handling variable and unordered trace-query sets; and *(3) multi-modal dynamics*, representing

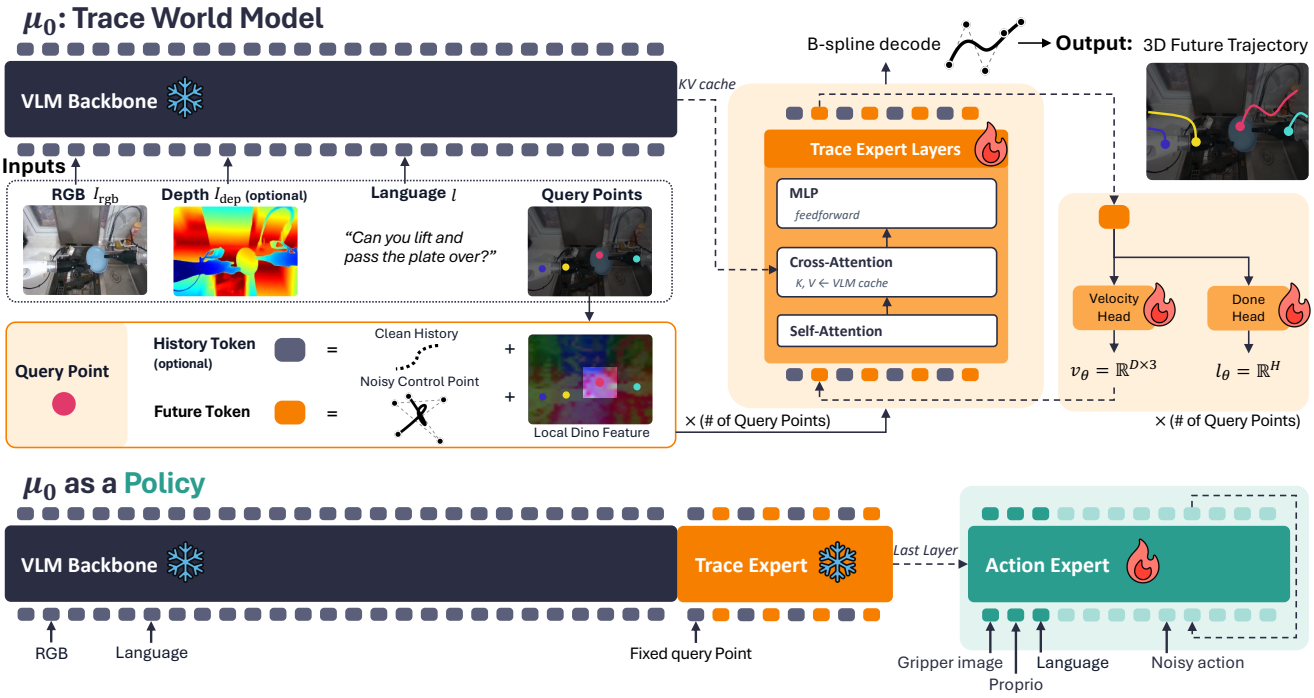

Figure 3: Overview of μ_0 and its action-expert interface. TraceExtract provides event-captioned 3D traces for semantic query keypoints. The VLM-conditioned trace context (Sec. 3.1) encodes RGB, language, and optional depth; spline query tokens (Sec. 3.2) represent each keypoint as an exchangeable B-spline query grounded by local DINO features; semantic flow matching (Sec. 3.3) denoises control points into smooth future 3D traces; and the action expert (Sec. 3.4) maps frozen trace features to executable robot actions.

plausible futures without averaging away contact-rich motion. We address these challenges with the three components in Fig. 3: (3.1) a VLM-conditioned backbone for scene and language context, (3.2) a permutation-equivariant trace expert for query-wise spline prediction, and (3.3) a semantic flow objective for structured future generation. Together, these choices turn flexible semantic keypoints into compact metric motion tokens rather than fixed grids or dense scene fields. Downstream action experts can then consume μ_0's trace representation, allowing video-only world-model pretraining to support robot control.

3.1. Multi-Modal Conditioning Backbone

Trace prediction requires both global intent and metric scene context: *(1) language* specifies the desired outcome, *(2) RGB* identifies objects and affordances, and *(3) depth* (optional) disambiguates 3D geometry when available.

Semantic reuse. As illustrated on the left of Fig. 3, we use a pretrained SmolVLM2-2.2B prefix to encode the RGB observation and instruction, then attach a trace expert that cross-attends to the VLM key-value cache while maintaining a separate motion-specific stream (Shukor et al., 2025). This separates *semantic memory*, preserved by the VLM, from *motion computation*, learned by the trace expert.

Depth pathway. Because metric depth is outside the native VLM input space, we route it through *(1) a separate trainable patch stem* before *(2) sharing deeper SigLIP layers* with RGB tokens. This lets the model exploit geometric cues without disrupting pretrained RGB statistics. Architectural and optimization details are in Appendix B.1.

3.2. Permutation-Equivariant Trace Expert

Exchangeable queries. A trace world model should accept arbitrary query keypoints, and its predictions should not depend on the order in which those keypoints are listed. μ_0 therefore treats each keypoint as an exchangeable query, matching the query-token block in Fig. 3. All queries share the same processing stack, preserving permutation equivariance across the keypoint dimension.

Spline targets. For each query, we subtract the current 3D anchor and represent the future as cubic B-spline control points following Liu et al. (2026). This target provides *(1) compactness*, replacing dense waypoints with a small control set; *(2) smoothness*, suppressing tracker jitter and high-frequency artifacts; and *(3) easier denoising*, reducing the output dimension.

Query tokenization. We tokenize each keypoint's history and noisy future controls as per-query tokens. Each token combines *(1) segment embeddings* for history versus future, *(2) Fourier embeddings* for current pixel location, and *(3) DINO features* for local semantics. Together, these choices ground each token in its visual entity while keeping the query set exchangeable. The VLM backbone and trace expert together form the pretrained μ_0, which downstream action experts reuse as a frozen motion prior. Full target-fitting, tokenization, and DINO-fusion details are in Appendix B.2.

3.3. Flow Matching with Semantic Structure

Even with smooth spline targets, future object motion is uncertain: *(1) multiple paths* can satisfy the same instruction, and *(2) traces* may be truncated or partially occluded. A deterministic regressor would tend to average these futures, yielding traces that are not necessarily actionable.

Conditional denoising. We instead train the trace expert as a conditional flow model over B-spline control points, shown in the denoising block of Fig. 3. Starting from noisy control points, the model predicts the velocity field toward clean controls under *(1) VLM context*, *(2) per-query token conditions*, and *(3) flow-time modulation* injected with adaLN-Zero (Peebles and Xie, 2023).

Structural constraints. The objective adds two terms for controllable traces: *(1) validity prediction*, which identifies when a keypoint trajectory should terminate under occlusion or track loss, and *(2) semantic rigidity*, which encourages keypoints within the same DINO cluster to preserve local geometry. The training loss is $\mathcal{L} = \mathcal{L}_{\text{flow}} + \lambda_{\text{done}}\mathcal{L}_{\text{done}} + \lambda_{\text{rig}}\mathcal{L}_{\text{rig}}$, where $\mathcal{L}_{\text{flow}}$ matches the control-point velocity field, $\mathcal{L}_{\text{done}}$ supervises per-step trajectory validity, and $\mathcal{L}_{\text{rig}}$ preserves local geometry within DINO clusters. At inference, μ_0 runs a denoising loop to decode the control points and reconstruct smooth 3D traces from them. The full objective and inference equations are in Appendix B.3.

3.4. Trace-Conditioned Action Expert

Embodiment transfer. μ_0 is pretrained from TraceExtract video supervision, but robot execution requires actions in a target embodiment. As shown on the right of Fig. 3, we freeze the pretrained μ_0—comprising *(1) the VLM backbone* and *(2) the trace expert*—then train only an action expert. This makes the pretrained μ_0 reusable across action experts while keeping the learned 3D motion prior embodiment agnostic and limiting action supervision to the target robot interface.

Policy interface. The policy uses frozen trace-denoising features as intermediate motion tokens rather than requiring a complete rollout or inverse-kinematics replay at every control step. Specifically, it *(1) reads* features from a single partial-denoising step of μ_0, *(2) injects* them into VLM features via gated cross-attention, and *(3) predicts* continuous action chunks with an action denoiser conditioned on gripper-camera,

Table 1: 2D and 3D trace prediction evaluation. Comparison of trajectory prediction quality over time horizons $T \in \{8, 16, 32\}$. The shaded column reports inference time for trace prediction on one image, with † denoting API latency. All baselines receive the same image and text pairs, except ‡ which requires depth input.

	Method	top1-ADE ↓			top5-ADE ↓			top1-FDE ↓			top5-FDE ↓			top1-DTW ↓			top5-DTW ↓			Inf. Time ↓
		$T=8$	16	32	8	16	32	8	16	32	8	16	32	8	16	32	8	16	32	
2D	Gemini-3.1-pro	0.190	0.274	0.305	0.161	0.232	0.253	**0.311**	0.425	0.424	0.254	0.321	0.311	**0.183**	0.258	0.284	0.152	0.208	0.224	78s†
	Gemini-3-flash	**0.187**	**0.271**	0.299	0.158	0.231	0.254	0.312	0.414	0.405	0.252	0.329	0.316	**0.183**	0.260	**0.281**	0.150	0.211	0.227	62s†
	GPT-5.5	0.199	0.281	0.307	0.178	0.249	0.272	0.329	0.411	**0.404**	0.284	0.344	0.329	0.196	0.274	0.299	0.173	0.238	0.259	38s†
	Track2Act (Bharadhwaj et al., 2024)	0.209	0.311	0.369	0.190	0.262	0.293	0.350	0.493	0.555	0.287	0.351	0.346	0.206	0.303	0.358	0.181	0.245	0.270	0.85s
	Hamster (Li et al., 2025)	0.202	0.276	**0.297**	0.178	0.239	0.256	0.326	**0.400**	0.411	0.274	0.320	0.330	0.197	0.261	0.277	0.170	0.220	0.233	14.4s
	μ_0 **(Ours)**	0.202	0.279	0.315	**0.124**	**0.188**	**0.227**	0.322	0.410	0.447	**0.186**	**0.261**	**0.284**	0.184	**0.254**	0.296	**0.114**	**0.171**	**0.211**	**0.29s**
3D	3DFlowAction (Zhi et al., 2025)	0.615	0.692	0.716	0.531	0.605	0.630	0.753	0.819	0.818	0.648	0.714	0.712	0.614	0.688	0.711	0.529	0.600	0.623	3.38s
	Dream2Flow‡ (Dharmarajan et al., 2026)	0.354	0.451	0.505	0.201	0.286	0.336	0.497	0.616	0.660	0.287	0.378	0.403	0.352	0.449	0.500	0.198	0.281	0.329	106.8s
	TraceGen‡ (Lee et al., 2026)	0.327	0.416	0.464	0.208	0.276	0.325	0.478	0.548	0.642	0.267	0.329	0.370	0.298	0.375	0.413	0.204	0.262	0.299	1.20s
	μ_0 **(Ours)**	**0.209**	**0.288**	**0.325**	**0.132**	**0.199**	**0.239**	**0.331**	**0.425**	**0.464**	**0.200**	**0.278**	**0.305**	**0.191**	**0.263**	**0.308**	**0.127**	**0.187**	**0.223**	**0.29s**

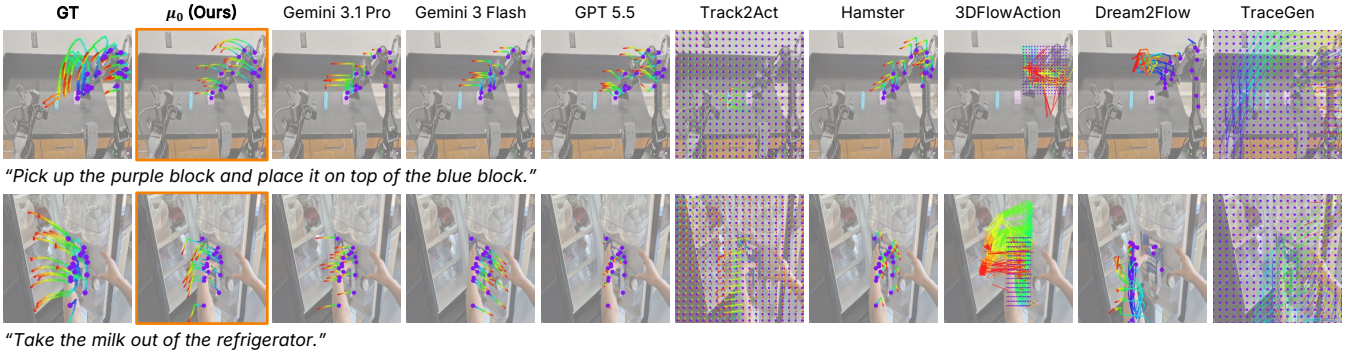

Figure 4: Qualitative comparison of predicted traces. We compare predicted traces from μ_0 and baselines on two manipulation tasks. μ_0 produces coherent and goal-directed traces, while avoiding the noisy or misaligned predictions observed in prior methods (more results in Appendix D.2).

proprioception, and language tokens. Details are in Appendix B.4.

4. Experiment

4.1. μ_0 Trace Prediction Quality

In this section, we evaluate the trace prediction quality of μ_0 both qualitatively and quantitatively, comparing our method against several 2D and 3D baselines. **Evaluation metrics.** Following Huang et al. (2026), we compute metrics exclusively on moving points. We evaluate trajectory prediction using Average Displacement Error (ADE) and Final Displacement Error (FDE) (Thakkar et al., 2026). To further evaluate trajectory shape independent of temporal misalignments, we utilize Dynamic Time Warping (DTW).

Results. Table 1 shows that μ_0 consistently improves multi-sample trace prediction quality. **(1)** In 2D, μ_0 achieves the best Top-5 ADE, FDE, and DTW across all horizons. These gains indicate that its sampled traces contain more accurate goal-directed futures even when Top-1 performance is competitive with strong VLM baselines. **(2)** In 3D, μ_0 obtains the best result on every reported ADE, FDE, and DTW metric across all horizons. **(3)** Beyond accuracy, μ_0 is also efficient: its 0.29s prediction latency is 2.9× faster than the next-fastest reported 2D baseline (Track2Act (Bharadhwaj et al., 2024), 0.85s). **(4)** The qualitative examples in Figure 4 further support these trends.

Table 2: Simulation results in RoboCasa365. We evaluate downstream action generation on 8 representative RoboCasa365 tasks and report success rates (%). **Bold** and underline numbers indicate the best and second-best results in each row, respectively.

Task	No pretraining	Action-labeled pretraining (VLA)		Video-only pretraining	
	Diffusion Policy (Chi et al., 2025)	π_0 (Black et al., 2025)	$\pi_{0.5}$ (Intelligence et al., 2025)	TraceGen (Lee et al., 2026) + action expert	Ours (μ_0 + action expert)
CloseFridge	34	44	34	38	**54**
OpenFridge	28	12	26	**36**	18
CoffeeServeMug	28	34	**48**	42	36
PickPlaceFridgeShelfToDrawer	28	30	**66**	30	40
TurnOnMicrowave	0	2	**12**	0	4
SlideToasterOvenRack	48	46	**76**	28	56
PickPlaceCounterToCabinet	6	18	**54**	0	12
TurnOnToasterOven	10	16	20	10	**22**
Average Success Rate (%)	22.75	25.25	**42**	23	30.25

4.2. Action Generation with Pretrained μ_0 under Both Simulated and Real-World Scenarios

In this section, we evaluate whether pretrained μ_0 can serve as a motion prior for action generation.

Simulated experiment setup. We evaluate each method on 8 representative tasks in RoboCasa365 (Nasiriany et al., 2026), a large-scale simulation benchmark for everyday kitchen manipulation that randomizes scene layouts, object instances, and initial configurations (details are in Appendix D.3). We benchmark our method against three classes of baselines: **(1) Diffusion Policy** (Chi et al., 2025), trained from scratch on target-domain demonstrations; **(2) action-labeled VLAs**, π_0 (Black et al., 2025) and $\pi_{0.5}$ (Intelligence et al., 2025), pretrained with large-scale robot action labels; and **(3) video-only trace models**, TraceGen (Lee et al., 2026), pretrained without proprioceptive or action supervision, like μ_0. For all pretrained methods, we fully finetune only the action expert on the RoboCasa365 data.

Results of simulated scenarios. Table 2 presents the success rates across the 8 selected RoboCasa tasks. **(1)** Overall, μ_0 + action expert achieves a 30.25% average success rate, outperforming π_0 by 5.0 points, despite relying solely on video-only pretraining. **(2)** At the same time, $\pi_{0.5}$ remains stronger on average; however, this comparison is not data-matched: $\pi_{0.5}$ benefits from large-scale action-labeled pretraining, which is costly and difficult to scale, whereas our method uses video-only pretraining. **(3)** Compared with the previous video-only trace baseline (TraceGen), μ_0 improves average success by 7.25 points, which we expect to reflect the benefit of stronger 3D trace prediction.

Real-world experiment setup. Our real-world experiments use a UR3 robot arm equipped with a two-finger gripper, as shown in Figure 5. We evaluate each method on three tasks: two multi-instruction tasks, *Pick <object> into Sink* and *Pour Almonds into <object>*, and a single *Unfold Towel* task. We collect 90, 80, and 50 demonstrations for the three tasks, respectively, and evaluate each task over 20 rollouts. Additional experimental details are reported in Appendix D.4.

Results of real-world scenarios. (1) Figure 6 shows that μ_0 + action expert achieves the highest average success rate of 91.7%, outperforming all baselines across the three real-world tasks on average. **(2)** Compared with the VLM + action expert baseline, which keeps the same policy architecture but removes the trace expert, μ_0 shows an 18.4 percentage-point gap, indicating that frozen trace features provide useful motion guidance beyond generic VLM representations. **(3)** μ_0 + action expert also surpasses the action-labeled VLA baselines π_0 and $\pi_{0.5}$ by 20.0 and 11.7 percentage points. **(4)** Compared with the previous video-only baseline TraceGen, μ_0 improves average success by 10.0 percentage points, which we attribute to the stronger TraceExtract supervision and architecture.

Scaling analysis. Our scaling results offer two main takeaways. First, trace prediction consistently improves with larger models and more pretraining data, yielding the best top5-DTW with the 2.59B model. Second,

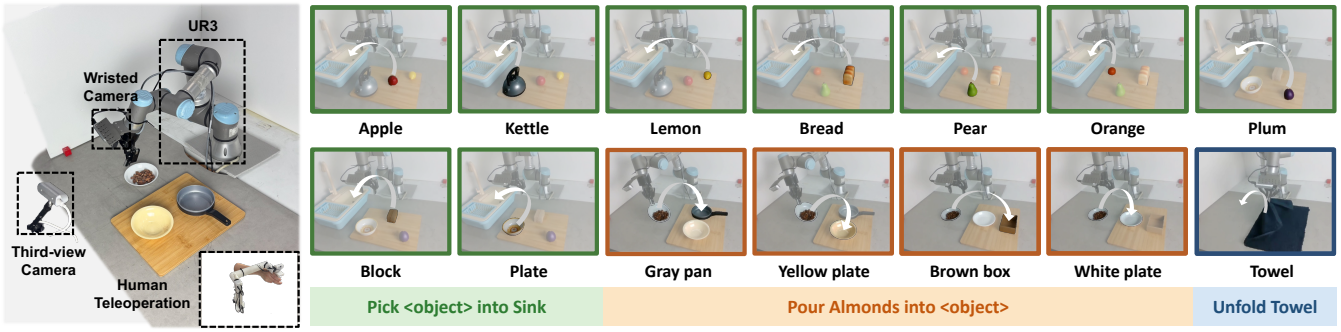

Figure 5: Real-world experimental setup and task visualizations. The setup includes a UR3 robot arm with a two-finger gripper and the three real-world manipulation tasks used for evaluation.

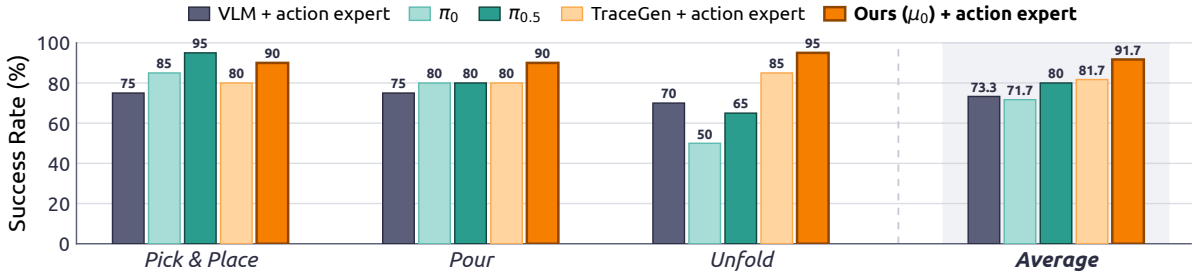

Figure 6: Real-world evaluation results. Bar charts show average success rates (%) for three in-distribution UR3 manipulation tasks. Pick & Place and Pour are averaged over multiple objects.

the trace representation effectively transfers to robot control: the performance gap between our model and the *w/o Trace* variant widens significantly as the action head size decreases, demonstrating that trace-space pretraining provides crucial motion structure that limited policy capacity cannot recover. Full protocols and results are in Appendix E.2 (Tables 7 and 8). **Design ablations.** Component-wise ablations verifying every major choice—including B-spline parameterization, DINOv2 features, rigidity loss, depth input, and historical traces—are detailed in Appendix E.1 (Table 6).

5. Related Work

World models and visual motion priors. Embodiment-agnostic robot learning leverages world models to forecast scene dynamics independently of specific action spaces (Du et al., 2023, Xu et al., 2024, Yuan et al., 2024a, Huang et al., 2026). While pixel-space models offer broad visual priors (Wu et al., 2024, Guo et al., 2026) and world-action models jointly predict frames and actions (Li et al., 2026b, Ye et al., 2026b,a), they waste capacity on dense appearance rather than geometry and contact. Intermediate representations like features, 2D tracks, and 3D flow mitigate this (Jang et al., 2026, Zhou et al., 2025b, Hu et al., 2025b, Gu et al., 2024, Bharadhwaj et al., 2024, Wen et al., 2024, Vecerik et al., 2024, Kambara et al., 2026, Zhi et al., 2025, Wang et al., 2026c), yet they carry distinct limitations: latent features lack control, 2D tracks lose metric depth, and fixed grids waste budget on backgrounds. Instead, μ_0 predicts explicit 3D trajectories for query-selected interaction points, providing a compact, metric, and reusable motion interface.

Trace-guided manipulation. Visual motion plans for manipulation generally fall into three categories: (i) VLM-based waypoint or end-effector tracking (Li et al., 2025, Zhou et al., 2025a, Yuan et al., 2024b, Yang et al., 2025), (ii) video generation followed by track extraction (Ko et al., 2024, Bharadhwaj et al., 2025, Li et al., 2026a, Dharmarajan et al., 2026), and (iii) policies directly predicting tracks via diffusion or flow matching (Nguyen et al., 2026, Gao et al., 2025, Lin et al., 2026). TraceGen (Lee et al., 2026) is closest

to our work but relies on fixed-grid traces and requires inference-time depth. Extended discussions are in Appendix F.

6. Conclusion

We introduced μ_0, a query-conditioned 3D trace-space world model for cross-embodiment manipulation. Instead of predicting pixels or embodiment-specific actions, μ_0 predicts smooth future 3D motion for semantically selected interaction keypoints. Its supervision comes from TraceExtract, which turns heterogeneous videos into event-captioned 3D trace tuples through semantic keypoint selection, globally aligned tracking, and motion-centric captioning. After video-only pretraining, the frozen trace model can be reused by action experts, providing an embodiment-agnostic motion prior for downstream robot control. Across trace forecasting, simulation, and real-world robot experiments, our results support 3D interaction traces as a compact and actionable representation for scalable robot world modeling.

Limitations and Future Work. μ_0 inherits errors from the perception stack used to construct traces: failures in semantic clustering, 3D reconstruction, tracking, or captioning can produce noisy supervision. The trace representation captures geometry and motion but does not explicitly model forces, tactile feedback, or contact modes, which may be important for fine manipulation. Our action expert evaluations focus on tabletop manipulation with limited embodiments and task families; broader validation on mobile manipulators, dexterous hands, and longer-horizon tasks remains future work.

Acknowledgements

Lee, Jung and Huang are supported by DARPA HR001124S0029-AIQ-FP-019, National Science Foundation TRAILS Institute (2229885). Private support was provided by Open Philanthropy and Apple. The authors acknowledge the National Artificial Intelligence Research Resource (NAIRR) Pilot for contributing to this research result. We thank Jonguk Cheon and Seokjin Park for their help and support with this project.

References

Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2act: Predicting point tracks from internet videos enables generalizable robot manipulation. In *ECCV*, 2024.

Homanga Bharadhwaj, Debidatta Dwibedi, Abhinav Gupta, Shubham Tulsiani, Carl Doersch, Ted Xiao, Dhruv Shah, Fei Xia, Dorsa Sadigh, and Sean Kirmani. Gen2act: Human video generation in novel scenarios enables generalizable robot manipulation. In *CoRL*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. In *RSS*, 2025.

Remi Cadene, Simon Alibert, Francesco Capuano, Michel Aractingi, Adil Zouitine, Pepijn Kooijmans, Jade Choghari, Martino Russi, Caroline Pascal, Steven Palma, Dana Aubakirova, Mustafa Shukor, Jess Moss, Alexander Soare, Quentin Lhoest, Quentin Gallouédec, and Thomas Wolf. LeRobot: An open-source library for end-to-end robot learning. In *ICLR*, 2026.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Daesol Cho, Youngseok Jang, Danfei Xu, and Sehoon Ha. Egoavflow: Robot policy learning with active vision from human egocentric videos via 3d flow. *arXiv preprint arXiv:2602.22461*, 2026.

Karthik Dharmarajan, Wenlong Huang, Jiajun Wu, Li Fei-Fei, and Ruohan Zhang. Dream2flow: Bridging video generation and open-world manipulation with 3d object flow. In *ICRA*, 2026.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In *NeurIPS*, 2023.

Chongkai Gao, Haozhuo Zhang, Zhixuan Xu, Cai Zhehao, and Lin Shao. Flip: Flow-centric generative planning as general-purpose manipulation world model. In *ICLR*, 2025.

Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, et al. Dreamdojo: A generalist robot world model from large-scale human videos. *arXiv preprint arXiv:2602.06949*, 2026.

Jiayuan Gu, Sean Kirmani, Paul Wohlhart, Yao Lu, Montserrat Gonzalez Arenas, Kanishka Rao, Wenhao Yu, Chuyuan Fu, Keerthana Gopalakrishnan, Zhuo Xu, et al. Rt-trajectory: Robotic task generalization via hindsight trajectory sketches. In *ICLR*, 2024.

Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-world: A controllable generative world model for robot manipulation. In *ICLR*, 2026.

Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. In *ICML*, 2025a.

Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. In *ICML*, 2025b.

Wenlong Huang, Yu-Wei Chao, Arsalan Mousavian, Ming-Yu Liu, Dieter Fox, Kaichun Mo, and Li Fei-Fei. Pointworld: Scaling 3d world models for in-the-wild robotic manipulation. In *CVPR*, 2026.

Adam Hung, Bardienus Pieter Duisterhof, and Jeffrey Ichnowski. 3pointr: 3d point tracks for robot manipulation pretraining from casual videos. *arXiv preprint arXiv:2603.08485*, 2026.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In *CoRL*, 2025.

Yoo Sung Jang, Kanchana Ranasinghe, Cristina Mata, Yichi Zhang, Jorge Mendez-Mendez, and Michael S Ryoo. Lace: Latent visual representation for cross-embodiment learning. *arXiv preprint arXiv:2605.16743*, 2026.

Motonari Kambara, Koki Seno, Tomoya Kaichi, Yanan Wang, and Komei Sugiura. Lilac: Language-conditioned object-centric optical flow for open-loop trajectory generation. *IEEE Robotics and Automation Letters*, 11 (6):6767–6774, 2026.

Jisoo Kim, Jungbin Cho, Sanghyeok Chu, Ananya Bal, Jinhyung Kim, Gunhee Lee, Sihaeng Lee, Seung Hwan Kim, Bohyung Han, Hyunmin Lee, et al. Pri4r: Learning world dynamics for vision-language-action models with privileged 4d representation. *arXiv preprint arXiv:2603.01549*, 2026.

Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B Tenenbaum. Learning to act from actionless videos through dense correspondences. In *ICLR*, 2024.

Seungjae Lee, Yoonkyo Jung, Inkook Chun, Yao-Chih Lee, Zikui Cai, Hongjia Huang, Aayush Talreja, Tan Dat Dao, Yongyuan Liang, Jia-Bin Huang, and Furong Huang. Tracegen: World modeling in 3d trace space enables learning from cross-embodiment videos. In *CVPR*, 2026.

Hongyu Li, Lingfeng Sun, Yafei Hu, Duy Ta, Jennifer Barry, George Konidaris, and Jiahui Fu. Novaflow: Zero-shot manipulation via actionable flow from generated videos. In *ICRA*, 2026a.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, et al. Causal world modeling for robot control. In *RSS*, 2026b.

Yi Li, Yuquan Deng, Jesse Zhang, Joel Jang, Marius Memmel, Caelan Reed Garrett, Fabio Ramos, Dieter Fox, Anqi Li, Abhishek Gupta, and Ankit Goyal. Hamster: Hierarchical action models for open-world robot manipulation. In *ICLR*, 2025.

Sixu Lin, Junliang Chen, Huaiyuan Xu, Zhuohao Li, Guangming Wang, Yixiong Jing, Sheng Xu, Runyi Zhao, Brian Sheil, Lap-Pui Chau, and Guiliang Liu. Roboflow4d: A lightweight flow world model toward real-time flow-guided robotic manipulation. In *ICML*, 2026.

Xinhang Liu, Yuxi Xiao, Donny Y Chen, Jiashi Feng, Yu-Wing Tai, Chi-Keung Tang, and Bingyi Kang. Trace anything: Representing any video in 4d via trajectory fields. In *ICLR*, 2026.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. RoboCasa: Large-scale simulation of everyday tasks for generalist robots. In *RSS*, 2024.

Soroush Nasiriany, Sepehr Nasiriany, Abhiram Maddukuri, and Yuke Zhu. Robocasa365: A large-scale simulation framework for training and benchmarking generalist robots. In *ICLR*, 2026.

E-Ro Nguyen, Yichi Zhang, Kanchana Ranasinghe, Xiang Li, and Michael S Ryoo. Pixel motion diffusion is what we need for robot control. In *CVPR*, 2026.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

Abraham Savitzky and Marcel JE Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639, 1964.

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

Neerja Thakkar, Shiry Ginosar, Jacob Walker, Jitendra Malik, Joao Carreira, and Carl Doersch. Forecasting motion in the wild. *arXiv preprint arXiv:2604.01015*, 2026.

Mel Vecerik, Carl Doersch, Yi Yang, Todor Davchev, Yusuf Aytar, Guangyao Zhou, Raia Hadsell, Lourdes Agapito, and Jon Scholz. Robotap: Tracking arbitrary points for few-shot visual imitation. In *ICRA*, 2024.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *CVPR*, 2025.

Jiaxu Wang, Yicheng Jiang, Tianlun He, Jingkai Sun, Qiang Zhang, Junhao He, Jiahang Cao, Zesen Gan, Mingyuan Sun, Qiming Shao, et al. Mvista-4d: View-consistent 4d world model with test-time action inference for robotic manipulation. *arXiv preprint arXiv:2602.09878*, 2026a.

Ruixiang Wang, Qingming Liu, Yueci Deng, Guiliang Liu, Zhen Liu, and Kui Jia. Eva: Aligning video world models with executable robot actions via inverse dynamics rewards. *arXiv preprint arXiv:2603.17808*, 2026b.

Xinkai Wang, Chenyi Wang, Yifu Xu, Mingzhe Ye, Fu-Cheng Zhang, Jialin Tian, Xinyu Zhan, Lifeng Zhu, Cewu Lu, and Lixin Yang. Lamp: Learning vision-language-action policies with 3d scene flow as latent motion prior. *arXiv preprint arXiv:2603.25399*, 2026c.

Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. In *RSS*, 2024.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In *ICLR*, 2024.

Mengda Xu, Zhenjia Xu, Yinghao Xu, Cheng Chi, Gordon Wetzstein, Manuela Veloso, and Shuran Song. Flow as the cross-domain manipulation interface. In *CoRL*, 2024.

Jianwei Yang, Reuben Tan, Qianhui Wu, Ruijie Zheng, Baolin Peng, Yongyuan Liang, Yu Gu, Mu Cai, Seonghyeon Ye, Joel Jang, et al. Magma: A foundation model for multimodal ai agents. In *CVPR*, 2025.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. In *NeurIPS*, 2024.

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. *arXiv preprint arXiv:2603.17240*, 2026a.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026b.

Chengbo Yuan, Chuan Wen, Tong Zhang, and Yang Gao. General flow as foundation affordance for scalable robot learning. In *CoRL*, 2024a.

Wentao Yuan, Jiafei Duan, Valts Blukis, Wilbert Pumacay, Ranjay Krishna, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. Robopoint: A vision-language model for spatial affordance prediction for robotics. In *CoRL*, 2024b.

Bowei Zhang, Lei Ke, Adam W. Harley, and Katerina Fragkiadaki. Tapip3d: Tracking any point in persistent 3d geometry. In *NeurIPS*, 2025.

Hongyan Zhi, Peihao Chen, Siyuan Zhou, Yubo Dong, Quanxi Wu, Lei Han, and Mingkui Tan. 3dflowaction: Learning cross-embodiment manipulation from 3d flow world model. *arXiv preprint arXiv:2506.06199*, 2025.

Enshen Zhou, Cheng Chi, Yibo Li, Jingkun An, Jiayuan Zhang, Shanyu Rong, Yi Han, Yuheng Ji, Mengzhen Liu, Pengwei Wang, et al. Robotracer: Mastering spatial trace with reasoning in vision-language models for robotics. *arXiv preprint arXiv:2512.13660*, 2025a.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning. In *ICML*, 2025b.

Appendix

A. Dataset Construction

A.1. Semantic Keypoint Sampling Details

For each chunk, we compute DINOv2 (Oquab et al., 2024) patch descriptors on a small set of representative frames and cluster the descriptors into entity-level groups. Cluster identities are propagated temporally by bipartite matching between adjacent frames, where the matching score combines feature similarity and spatial overlap. Given a per-chunk budget of N keypoints, each entity receives a quota proportional to its visible patch coverage, with a minimum allocation for small entities that remain salient but occupy few patches. Final keypoints are selected by farthest-point sampling within each entity mask on frames where the entity is visible, producing spatially diverse query points that are less likely to fall on background or transient occluders. The resulting DINO cluster identity is stored with each keypoint and reused as the part label for the rigidity loss in Appendix B.3.

A.2. Movement Filter

A substantial fraction of tracked points correspond to background or static structures that contribute no information about the task and bias the model toward zero motion. For each keypoint i, we project its trace through cam_{ref} to obtain (u_i^t, v_i^t, z_i^t), weight depth by $\lambda_z = 0.1$ so pixel motion dominates, and compute the trace diameter $d_i = \max_{t,t' \in \mathcal{V}_i} \|(u_i^t - u_i^{t'},\, v_i^t - v_i^{t'},\, \lambda_z(z_i^t - z_i^{t'}))\|_2$ over the visible frame set $\mathcal{V}_i$. A keypoint is marked *moving* when d_i exceeds $\tau_m = 40$ pixels. Using maximum pairwise displacement, rather than instantaneous velocity, captures the full extent of motion while remaining robust to per-frame tracker jitter.

A.3. Hybrid Global–Local 3D Reconstruction

Globally consistent depth, intrinsics, and extrinsics are the prerequisite for placing every 3D trace in a single reference camera frame. TraceExtract uses a hybrid VGGT (Wang et al., 2025) scheme that combines one global sparse pass with dense local passes, enabling long-horizon manipulation videos to be processed without fitting the entire sequence in memory.

Global sparse pass. Given a video of length T_{total}, we uniformly subsample at most T_{sparse} anchor frames and feed them through VGGT in a single forward call, yielding extrinsics $\{\boldsymbol{E}_t^{\text{sparse}}\}_{t \in \mathcal{S}}$ in a common *global frame* together with a single $\boldsymbol{K}^{\text{global}}$ obtained by averaging the per-frame intrinsics. Per-chunk intrinsics introduce visible discontinuities at chunk boundaries, so a single shared $\boldsymbol{K}^{\text{global}}$ is essential.

Dense passes and SE(3) alignment. The full video is split into non-overlapping chunks of T_{chunk} frames, each producing chunk-local depth $\boldsymbol{D}^{(c)}$ and extrinsics $\{\boldsymbol{E}_t^{(c)}\}$. For every chunk c, the anchor frames in $\mathcal{S} \cap c$ act as shared observations, and we solve for the rigid transform $\boldsymbol{A}^{(c)} \in \text{SE}(3)$ that maps chunk-local poses to global poses,

$$\boldsymbol{A}^{(c)} = \arg\min_{\boldsymbol{A} \in \text{SE}(3)} \sum_{t \in \mathcal{S} \cap c} \left\| \boldsymbol{A}\, \boldsymbol{E}_t^{(c)} - \boldsymbol{E}_t^{\text{sparse}} \right\|^2. \tag{1}$$

Because each chunk aligns *directly* to the same global anchors rather than to its predecessor, alignment errors are independent and bounded across chunks instead of compounding.

A.4. Progressive 3D Tracking Across Chunks

Running a 3D point tracker independently per chunk discards continuity: the same physical point would be re-discovered with a new identity in every chunk, and any object missed by DINO in one chunk would simply

vanish. We instead track *progressively*. The first chunk is processed by feeding peak-frame keypoints through TAPIP3D (Zhang et al., 2025), which produces 3D world-space coordinates $\{\boldsymbol{p}_i^t\}$ and visibility flags. For chunk $c \geq 1$, every active group is propagated by using its last known 3D world position in previous chunk as a 3D query at the first frame of chunk c; because positions live in the same global frame, propagation operates on world-space 3D coordinates and is therefore robust to the large camera motion typical of egocentric video.

A.5. Event-Centric Captioning Details

Given tracked traces, we compute a scalar motion signal by averaging per-frame accelerations over valid moving keypoints. The signal is smoothed into $\tilde{a}_t$ with a Savitzky–Golay filter (Savitzky and Golay, 1964), and prominent local maxima p_i are treated as action anchors. Chunk boundaries are placed at low-motion transition points,

$$b_i = \arg\min_{t \in [p_i, p_{i+1}]} \tilde{a}_t, \tag{2}$$

with minimum- and maximum-duration constraints to avoid degenerate clips. For each chunk, the VLM receives the start, midpoint, and end frames, together with an optional motion mask rendered from the moving traces and an optional episode-level task description. It produces a structured caption describing the object state at the beginning, the interaction that occurs, and the state change at the end. A text-only LLM then merges adjacent chunk captions over sliding windows, yielding paired frame ranges for both fine-grained event captions and coarser task summaries.

B. Architecture and Training Details

B.1. Backbone and Optimization Details

The conditioning backbone begins with a pretrained SmolVLM2-2.2B model acting as a vision-language prefix, truncated to its first $L_{\text{vlm}} = 20$ text-decoder layers. The Trace Expert has the same depth (20 layers) with hidden width scaled to $0.5\times$ that of the VLM. Following Shukor et al. (2025), the Trace Expert interleaves cross-attention against the VLM key-value cache with self-attention every two layers. The inputs to the VLM prefix are an RGB image I_{rgb}, an optional metric depth map rendered as an RGB image I_{dep}, and a tokenized textual instruction l; both image modalities are resized to 512×512.

To incorporate metric depth without disrupting pretrained RGB visual statistics, I_{dep} is normalized through a Turbo colormap and routed through a separate trainable patch-embedding stem cloned from the RGB stem at initialization. RGB and depth tokens then share the subsequent deeper SigLIP layers, allowing the network to adapt to depth statistics while preserving a unified visual representation. RGB frames pass through ColorJitter with strength s=0.3; depth is augmented in the meter domain with zero-mean Gaussian noise of standard deviation σ_d=0.01 m *before* the Turbo colormap, preserving the meter-to-color mapping. We optimize with AdamW at base learning rate 10^{-4} and weight decay 10^{-10}, gradient clipped to norm 10, with a $0.1\times$ multiplier on the VLM parameter group so the pretrained representation drifts slowly while the expert and trace projections adapt quickly. Training runs for 2×10^5 steps with gradient checkpointing, an effective batch size of 24 across two GPUs (6 per GPU), and N uniformly sampled from $[1, 256]$ keypoints per sample. The VLM and SigLIP tower for RGB is frozen from SmolVLM2-2.2B; the action expert, trace projections, embedding tables, the depth-only stem, and the adaLN-Zero heads are randomly initialized, with the adaLN-Zero output Linears and the uv-MLP output Linear zero-initialized so the model begins at a well-conditioned step-zero identity.

B.2. Trace Target and Tokenization Details

The Trace Expert consumes three slices of the TraceExtract trace $\boldsymbol{T}_{\text{ref}}^{t-h:t+H}$ in cam_{ref}: a past history $\boldsymbol{H} \in \mathbb{R}^{N\times h\times 3}$, a current anchor $\boldsymbol{c} \in \mathbb{R}^{N\times 3}$ at frame t, and a future target $\boldsymbol{T}^1 \in \mathbb{R}^{N\times H\times 3}$, with h=8 and H=32. We subtract the anchor from history ($\boldsymbol{H} \leftarrow \boldsymbol{H} - \boldsymbol{c}$) and predict an anchor-relative, per-axis-rescaled future

$$\tilde{\boldsymbol{T}}^1_{n,k} = (\boldsymbol{T}^1_{n,k} - \boldsymbol{c}_n)/\boldsymbol{s}_\Delta, \tag{3}$$

where $\boldsymbol{s}_\Delta$ is a per-axis 95th-percentile scale precomputed once over the training corpus. Anchor-relative targets remove the slow scene-coordinate component and match the variance to the unit-Gaussian noise prior.

Rather than regress the H-step anchor-relative future directly, we re-parameterize each keypoint's future as a degree-3 B-spline with D=10 control points. The anchor-prepended scaled future $[\mathbf{0}; \tilde{\boldsymbol{T}}^1_n] \in \mathbb{R}^{(H+1)\times 3}$ is fit in the dataloader by row-weighted ridge least squares,

$$\boldsymbol{P}^\star_n = \arg\min_{\boldsymbol{P}\in\mathbb{R}^{D\times 3}} \left\|\boldsymbol{M}_n \odot (\boldsymbol{B}\boldsymbol{P} - [\mathbf{0}; \tilde{\boldsymbol{T}}^1_n])\right\|_F^2 + \lambda^2_{\text{bsp}} \left\|\boldsymbol{\Gamma}\boldsymbol{P}\right\|_F^2, \tag{4}$$

where $\boldsymbol{B} \in \mathbb{R}^{(H+1)\times D}$ is a fixed cubic B-spline basis sampled on a uniform grid with the curve pinned at the anchor (t=0), the per-step row weight $\boldsymbol{M}_n$ zeros invalid future steps so they exert no pull on $\boldsymbol{P}$, and $\boldsymbol{\Gamma}$ is the first-order finite-difference operator on consecutive control points. The Tikhonov term with λ_{bsp}=0.2 gently equalizes control-point spacing and compresses the flow-matching target distribution, and an element-wise post-fit clip $|\boldsymbol{P}^\star| \le 1.5$ bounds the target box. A keypoint participates in the flow loss only when at least D of its H future steps are valid; otherwise it is dropped from the flow loss entirely. The network's flow-matching target is $\boldsymbol{P}^\star \in \mathbb{R}^{N\times D\times 3}$, and rollouts decode in a single matrix multiply $\hat{\boldsymbol{T}}^1 = \boldsymbol{B}\hat{\boldsymbol{P}}$ with the anchor row stripped.

We adopt a flat tokenization that splits the trace stream into a clean-history segment and a noisy control-point segment, each with its own grouping axis. History is grouped along time at g_{time}=h, yielding G_{hist}=h/g_{time}=1 token per keypoint; control points are grouped along the control-point axis at g_{cp}=D, yielding G_{cp}=D/g_{cp}=1 token per keypoint. Thus, each keypoint contributes $G = G_{\text{hist}} + G_{\text{cp}}$ tokens indexed by j. Two separate linear lifts $W_{\text{hist}} : \mathbb{R}^{3g_{\text{time}}} \to \mathbb{R}^{D_{\text{exp}}}$ and $W_{\text{cp}} : \mathbb{R}^{3g_{\text{cp}}} \to \mathbb{R}^{D_{\text{exp}}}$ produce the base token $\boldsymbol{e}_{n,j}$. We add three positional components: a learned group-index embedding, a binary segment embedding separating history from future, and a 2D Fourier expansion of the current-frame (u, v) passed through a zero-initialized uv-MLP. RoPE positions for all suffix tokens are pinned to the prefix-end index, so rotary attention contributes no positional signal between suffix tokens; all temporal and spatial information lives in the additive embeddings. Within the suffix, attention is causal across the history sub-block and bidirectional elsewhere—control-point tokens attend to each other and to all history—which keeps the keypoint axis exchangeable.

While the VLM encodes global scene-level context, predicting precise traces requires sharp, location-specific cues for each query point. Inspired by Thakkar et al. (2026), we sample a frozen DINO-base feature map at each keypoint's current-frame pixel coordinate via bilinear grid sampling. This yields a localized semantic feature $\boldsymbol{f}^{\text{dino}}_n \in \mathbb{R}^{D_{\text{dino}}}$ for keypoint n. A two-layer MLP fuses this descriptor into each trace token associated with the keypoint,

$$\boldsymbol{e}_{n,j} \leftarrow W_2\,\text{SiLU}\Big(W_1\,\text{concat}(\boldsymbol{e}_{n,j}, \boldsymbol{f}^{\text{dino}}_n)\Big), \tag{5}$$

where W_1 and W_2 are learnable weights and $j \in \{\text{hist}, \text{cp}\}$. This directly injects part-level semantic priors at the token level, bridging global scene context with localized point statistics.

B.3. Flow-Matching Objective and Inference

We train the Trace Expert to generate target B-spline control points $\boldsymbol{P}^\star$ using conditional flow matching. Given standard Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and virtual time $\tau \in [0,1]$, the linear probability path is

$$\boldsymbol{P}^\tau = \tau\boldsymbol{\epsilon} + (1-\tau)\boldsymbol{P}^\star. \tag{6}$$

The network v_θ predicts the constant-in-time target velocity $\boldsymbol{\epsilon} - \boldsymbol{P}^\star$ that transports noise to clean data. To condition the architecture on the flow time step, we route τ through an adaLN-Zero module at each Trace Expert layer. A shared sinusoidal embedding encodes τ and emits layer-specific shift, scale, and gate vectors for both the attention and MLP sublayers. The final linear layer of each conditioning head is initialized to zero, so each expert block acts as an exact identity function at initialization.

The primary flow loss is the masked mean squared error of the predicted velocity in control-point space,

$$\mathcal{L}_{\text{flow}} = \mathbb{E}_{\tau,\boldsymbol{\epsilon}}\left[\left\|v_\theta(\boldsymbol{P}^\tau, \tau, F_{\text{cond}}) - (\boldsymbol{\epsilon} - \boldsymbol{P}^\star)\right\|_2^2\right], \tag{7}$$

computed only over valid and present keypoints. To handle trace truncation, a validity head pools the future control-point tokens per keypoint and predicts an H-dimensional per-step validity logit, trained via sigmoid cross-entropy $\mathcal{L}_{\text{done}}$,

$$\mathcal{L}_{\text{done}} = \frac{\sum_{t=1}^{H} \ell_{\text{BCE}}(\hat{d}_{n,t}, y_{n,t})}{N}, \tag{8}$$

where $\hat{d}_{n,t}$ is the predicted per-step validity logit for keypoint n at future step t, $y_{n,t} \in \{0,1\}$ is the ground-truth per-step validity. At inference, this head provides a stop index to freeze the decoded trace past its predicted end.

We also introduce an auxiliary rigidity loss $\mathcal{L}_{\text{rig}}$ to preserve spatial structural consistency by regularizing the clean control points reconstructed in-flight, $\hat{\boldsymbol{P}}_n = \boldsymbol{P}_n^\tau - \tau v_\theta$. Inspired by Liu et al. (2026), this loss penalizes non-rigid deformations within the same object part. Unlike prior work that relies on ground-truth object segmentation masks available only in synthetic environments, we use the DINO cluster identities produced by TraceExtract. Within each cluster, the pairwise distance between control points of different keypoints should remain invariant across the control-point sequence:

$$\mathcal{L}_{\text{rig}} = \mathbb{E}_{\tau,\boldsymbol{\epsilon}}\left[\frac{1}{|R|}\sum_{(n,n')\in R} \text{Var}_d\left(\left\|\hat{\boldsymbol{P}}_{n,d} - \hat{\boldsymbol{P}}_{n',d}\right\|_2^2\right)\right], \tag{9}$$

where R is the set of unique keypoint pairs sharing a part cluster identity and $d \in \{1,\ldots,D\}$ indexes control points. The joint objective is

$$\mathcal{L} = \mathcal{L}_{\text{flow}} + \lambda_{\text{done}}\mathcal{L}_{\text{done}} + \lambda_{\text{rig}}\mathcal{L}_{\text{rig}}. \tag{10}$$

At inference, we integrate v_θ with a 4-step Euler scheme on $\tau \in [1,0]$ and decode the absolute trace through the B-spline basis.

B.4. Trace-Conditioned Action Expert Details

The action expert conditions on intermediate trace features rather than fully denoised traces. Following partial-denoising schemes used in recent work (Hu et al., 2025a, Wang et al., 2026c), we initialize a pure-noise control-point input $\boldsymbol{P}^0 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and simulate a single step of the 4-step Euler solver. We then extract

the intermediate hidden states of the Trace Expert as the motion descriptor z_{trace}. This single step preserves task-relevant dynamics while avoiding the computational cost of a full rollout.

To inject 3D dynamics without disrupting pretrained VLM representations, we fuse z_{trace} into the last-layer VLM features through a gated cross-attention module. Let $\tilde{h}_{\text{trace}} = \text{LN}(W_{\text{proj}} z_{\text{trace}})$ denote the projected motion features. The guided features are

$$z_{\text{guided}} = z + \sigma(g) \cdot \text{CA}\big(Q = \text{LN}(z),\ K = V = \tilde{h}_{\text{trace}}\big)\,, \tag{11}$$

where CA denotes multi-head cross-attention, z denotes the last-layer VLM features, and g is a learnable scalar gate shared across all heads and spatial positions. We initialize g at zero and pass it through a sigmoid $\sigma(\cdot)$ to bound the gate within $(0,1)$, starting the policy in a weak motion-injection regime that strengthens only when beneficial.

The action expert adopts the self-attention architecture of $\pi_{0.5}$ (Intelligence et al., 2025) and generates continuous actions via flow matching. Beyond the guided features z_{guided}, which serve as the conditioning prefix, the expert receives three additional inputs and tokenizes each through a dedicated stem: a gripper-camera image encoded by DINOv2 (Oquab et al., 2024), robot proprioception mapped through an MLP, and the language instruction. The noisy action sequence enters the expert as the query. We define a linear action path $a^\tau = (1-\tau)\epsilon_a + \tau a$ with $\epsilon_a \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, and train the velocity field v_ϕ with

$$\mathcal{L}_{\text{action}} = \mathbb{E}_{\tau, a, \epsilon_a} \left\| v_\phi\big(a^\tau, \tau, z_{\text{guided}}, c\big) - (a - \epsilon_a) \right\|_2^2, \tag{12}$$

where c collects the proprioception, gripper-camera, and language conditions.

C. Model Training

C.1. Training Strategy

To ensure μ_0 can robustly predict traces even when historical traces or metric depth are unavailable at inference, we apply a two-level history dropout where we either drop the historical traces of all keypoints simultaneously with probability 0.2, or drop each keypoint's history independently with probability 0.3. The metric depth channel is randomly omitted with probability 0.7 to allow the model to flexibly fall back on static RGB observations. The pretrained VLM backbone is kept frozen to preserve its generalist visual-language representations, allowing only the trace expert and projection layers to adapt to 3D kinematics.

D. Experiment Details

D.1. Metric

For each evaluation slot, the policy predicts 16 keypoint traces over $T = \{8, 16, 32\}$ future timesteps in normalized image–depth space $(u, v, z) \in [-1, 1]^2 \times \mathbb{R}_{\geq 0}$ (UV in the resized 256×256 camera frame, z in metric meters). Let $\hat{\tau}_k^{(s)} \in \mathbb{R}^{T \times 3}$ denote the s-th sample for keypoint k, and $\tau_k^\star$ its ground-truth future. All distances below use the Euclidean pointwise cost in (u, v, z) space and are reported as means over valid future steps.

- **minADE and minFDE.** Average Displacement Error (ADE) and Final Displacement Error (FDE) compute the mean Euclidean distance over all predicted timesteps and the final timestep, respectively. We take

the minimum over the S samples:

$$\text{minADE} \;=\; \mathbb{E}_{\text{slot}}\left[\frac{1}{K}\sum_{k=1}^{K}\min_{s\in[S]}\frac{1}{T}\sum_{t=1}^{T}\|\hat{\tau}_{k,t}^{(s)}-\tau_{k,t}^{\star}\|_2\right],$$

$$\text{minFDE} \;=\; \mathbb{E}_{\text{slot}}\left[\frac{1}{K}\sum_{k=1}^{K}\min_{s\in[S]}\|\hat{\tau}_{k,T}^{(s)}-\tau_{k,T}^{\star}\|_2\right].$$

- **minDTW.** For each keypoint we compute the Dynamic Time Warping distance between each sample and the GT, and take the minimum over samples:

$$\text{minDTW} \;=\; \mathbb{E}_{\text{slot}}\left[\frac{1}{K}\sum_{k=1}^{K}\min_{s\in[S]}\text{DTW}\left(\hat{\tau}_k^{(s)},\,\tau_k^{\star}\right)\right].$$

  DTW allows monotonic time warping, so it scores the *shape* of the predicted path independent of small temporal misalignments.
- **minFD.** Identical aggregation, with the discrete Fréchet distance replacing DTW:

$$\text{minFD} \;=\; \mathbb{E}_{\text{slot}}\left[\frac{1}{K}\sum_{k=1}^{K}\min_{s\in[S]}\text{FD}\left(\hat{\tau}_k^{(s)},\,\tau_k^{\star}\right)\right].$$

  Whereas DTW averages pointwise displacement after alignment, the Fréchet distance is the *maximum* pointwise displacement over the optimal monotonic alignment, so it is sensitive to large excursions and endpoint errors that DTW averages away. We report it alongside model parameters in this appendix, as downstream consumers care about worst-case deviations along the path.
- **Inference time and Parameters.** Inference time represents the mean wall-clock latency per slot on a single A6000 GPU (reported in the main text). Total parameter counts for the respective models are provided alongside the FD results.

D.2. Additional results on Trace Prediction

Fréchet distance comparison. Beyond the metrics in the main paper, we further evaluate trace prediction quality using the Fréchet distance (FD), which measures the geometric similarity between predicted and ground-truth traces while accounting for their ordering along the path. As shown in Table 3, we report Top-1 FD and Top-5 FD across time horizons $T \in \{8, 16, 32\}$ against both 2D and 3D baselines, where all methods receive the same image–text pairs except for depth-conditioned baselines. Our method achieves the strongest overall FD performance across both metrics, indicating that our predicted traces are accurate on average, geometrically faithful to the ground-truth motion, and consistent as the prediction horizon grows.

Parameter efficiency. Table 3 also reports performance relative to model size. Our method attains strong trace prediction performance while maintaining a favorable parameter-efficiency trade-off compared with competing baselines. For a fair comparison, we report the parameter count of each baseline at trace inference time. Specifically, we count every component that participates in the forward pass producing the predicted trace, including frozen pretrained backbones, diffusion U-Nets, vision encoders, and trace prediction heads. Components used only during training, such as teacher networks or auxiliary heads, and components used only in downstream action execution, such as separate residual policies or optimization-based action solvers, are excluded. For methods with released checkpoints, parameter counts are obtained by summing `numel()`

Table 3: 2D and 3D trace prediction evaluation (Fréchet Distance and parameters). Comparison over time horizons $T \in \{8, 16, 32\}$. All baselines receive the same image and text pairs, except ‡ which requires depth input.

	Method	top1-FD ↓			top5-FD ↓			Params
		$T = 8$	16	32	8	16	32	
2D	Gemini-3.1-pro	0.324	0.467	0.505	0.269	0.385	0.416	?
	Gemini-3-flash	0.324	0.467	**0.504**	0.266	0.387	0.417	?
	GPT-5.5	0.342	0.476	0.511	0.299	0.415	0.449	?
	Track2Act (Bharadhwaj et al., 2024)	0.363	0.543	0.631	0.304	0.420	0.451	0.47B
	Hamster (Li et al., 2025)	0.339	0.462	0.505	0.291	0.390	0.429	13.5B
	μ_0 **(Ours)**	**0.314**	**0.446**	0.517	**0.200**	**0.306**	**0.370**	2.59B
3D	3DFlowAction (Zhi et al., 2025)	0.765	0.843	0.866	0.664	0.747	0.772	2.04B
	Dream2Flow‡ (Dharmarajan et al., 2026)	0.547	0.710	0.787	0.325	0.464	0.530	11.3B
	TraceGen‡ (Lee et al., 2026)	0.450	0.560	0.642	0.291	0.395	0.457	0.67B
	μ_0 **(Ours)**	**0.329**	**0.455**	**0.527**	**0.210**	**0.319**	**0.384**	2.59B

over all loaded parameters. For closed-source models or methods without public checkpoints, we use reported model sizes when available and otherwise mark the parameter count as undisclosed.

Qualitative results. Figure 7 presents additional qualitative comparisons across a diverse set of manipulation tasks. Consistent with the quantitative results above, μ_0 consistently produces coherent, task-relevant traces that better align with the intended manipulation dynamics, whereas baselines often generate sparse, noisy, overly dense, or spatially misaligned traces.

D.3. RoboCasa365

Environment details. We evaluate simulated action generation in RoboCasa365 (Nasiriany et al., 2026), a large-scale household manipulation benchmark built on the RoboCasa kitchen simulation platform (Nasiriany et al., 2024). The benchmark provides diverse kitchen layouts, object assets, and task initializations, making it well suited for testing whether policies generalize across scene and object variation rather than memorizing a fixed setup. We use the PandaOmron mobile manipulator, which consists of a Franka Panda arm mounted on an Omron mobile base and equipped with a gripper. Each policy observes two 256×256 RGB inputs, a left third-person camera image and a wrist/gripper camera image, together with a language instruction and a 16-dimensional proprioceptive state. The action space is 12-dimensional, including arm motion, gripper control, and mobile-base control.

We evaluate 8 representative atomic tasks from RoboCasa365: *CloseFridge*, *OpenFridge*, *CoffeeServeMug*, *PickPlaceFridgeShelfToDrawer*, *TurnOnMicrowave*, *SlideToasterOvenRack*, *PickPlaceCounterToCabinet*, and *TurnOnToasterOven*. For each task, we use 100 demonstrations, resulting in 800 demonstrations in total. All methods use the same demonstrations, RGB observations, language instructions, and proprioceptive states. TraceGen additionally requires depth input, so we estimate depth from RGB observations using Depth Anything V2 (Yang et al., 2024) and provide the predicted depth images only to TraceGen.

Training details. We use the LeRobot (Cadene et al., 2026) implementations of Diffusion Policy (Chi et al., 2025), π_0 (Black et al., 2025), and $\pi_{0.5}$ (Intelligence et al., 2025). Diffusion Policy is trained from scratch on the target RoboCasa365 demonstrations using the multi-task DiT policy. For π_0, $\pi_{0.5}$, TraceGen + action expert, and μ_0 + action expert, we freeze the pretrained backbone and train only the action expert on the target demonstrations. Thus, our method uses the frozen μ_0 trace model as a motion-prior feature extractor while optimizing a RoboCasa365-specific action expert for control. All training runs are performed on 4 NVIDIA L40S GPUs. Table 4 summarizes the shared training hyperparameters.

Figure 7: Additional qualitative comparisons. We show predicted traces across all methods on additional manipulation tasks, one per row, with the language instruction shown below each example. Columns follow the same ordering as the main paper: ground truth (GT), our method (μ_0), general-purpose VLMs (Gemini 3.1 Pro, Gemini 3 Flash, GPT 5.5), and trace prediction baselines (Track2Act, Hamster, 3DFlowAction, Dream2Flow, TraceGen). Across diverse tasks, our method consistently produces coherent, task-relevant traces that better align with the intended manipulation dynamics, while baselines often generate sparse, noisy, overly dense, or spatially misaligned traces.

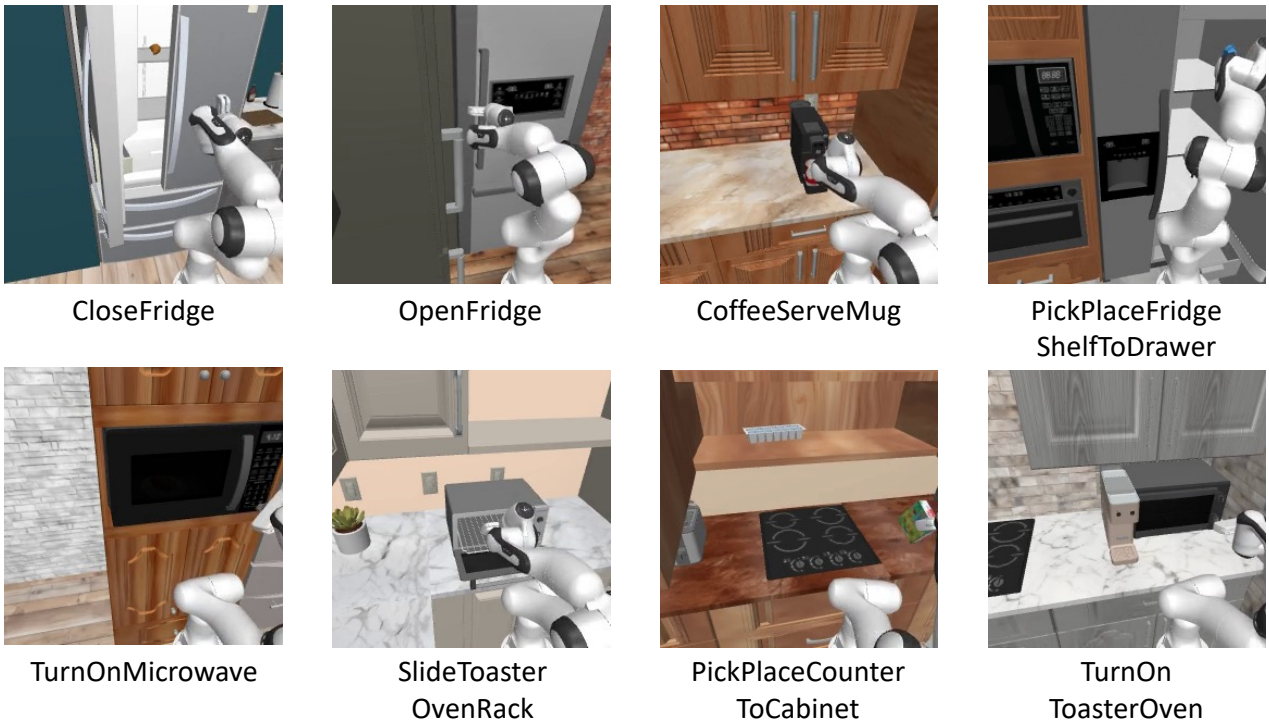

Figure 8: RoboCasa365 simulation examples. Example evaluation scenes from the 8 RoboCasa365 tasks used in our simulated experiments. The benchmark randomizes scene layouts, object instances, and initial configurations across rollouts, emphasizing policy generalization rather than memorization of a fixed scene.

Table 4: RoboCasa365 training hyperparameters. We use the same hyperparameters for all methods.

Hyperparameter	Value
Action dimension	12
Action horizon	16
Execution horizon	8
Batch size	32
Optimizer	AdamW
Learning rate	1×10^{-4}
Warmup steps	1,000
Training steps	50,000

We evaluate each trained policy over 50 rollouts per task. During evaluation, RoboCasa365 randomizes scene layouts, object instances, and initial configurations across rollouts. We use the default sparse task-completion signal from the environment and report the success rate (%) for each task, together with the average success rate across all eight tasks.

D.4. Real-world Robot

Hardware setup. Figure 5 shows the real-robot platforms that we use for both demonstration collection and policy evaluation. A fixed-base UR3 manipulator with a two-finger gripper executes all manipulation tasks. Two RGB cameras, mounted respectively at a third-person viewpoint and on the wrist, provide 224×224 visual observations. The robot proprioception comprises the 6D end-effector pose and the gripper state,

Table 5: Real-robot training hyperparameters. We use the same task-specific hyperparameters for all methods. Each subtable corresponds to one real-world task.

Pick <object> into Sink

Hyperparameter	Value
Action dimension	7
Action horizon	50
Execution horizon	25
Batch size	32
Optimizer	AdamW
Learning rate	5×10^{-5}
Warmup steps	400
Training steps	8,000

Pour Almonds into <object>

Hyperparameter	Value
Action dimension	7
Action horizon	50
Execution horizon	25
Batch size	32
Optimizer	AdamW
Learning rate	5×10^{-5}
Warmup steps	300
Training steps	6,000

Unfold Towel

Hyperparameter	Value
Action dimension	7
Action horizon	50
Execution horizon	25
Batch size	32
Optimizer	AdamW
Learning rate	5×10^{-5}
Warmup steps	300
Training steps	6,000

yielding a 7D state vector. A human teleoperator collects demonstrations by controlling the end-effector pose and gripper command through a custom teleoperation interface.

Training details. Table 5 reports the task-specific training hyperparameters, which we share across all methods to ensure a fair comparison. For our method, we freeze the VLM backbone and the trace expert of μ_0, and we train only the action expert from scratch on the collected demonstrations.

E. Additional Results

E.1. Ablation Studies

To validate the architectural and optimization design choices in μ_0, we evaluate the impact of core components on trace prediction quality by systematically disabling or modifying them.

Architectural and Optimization Design. We first isolate the contributions of our specific modeling choices:

- **w/o B-spline parameterization:** Instead of predicting $D = 10$ B-spline control points, the model directly regresses the raw $H = 32$ anchor-relative future steps.
- **w/o DINOv2 features:** We remove the per-keypoint patch-feature injection (Eq. 5), forcing the model to rely solely on the global vision–language prefix without explicit part-level semantics.
- **Rigidity loss variations:** We experiment with modifying the weight of the auxiliary rigidity loss (λ_{rig}) and completely removing it ($\lambda_{\text{rig}} = 0$) to measure its impact on preserving intra-part physical consistency.

Input Modality Robustness. Furthermore, we analyze the model's robustness to missing or degraded input modalities. Specifically, we train variants where metric depth is omitted (**w/o Depth**) and where the short past trajectory is removed (**w/o Historical Trace**), forcing the model to predict future motion from a static RGB observation alone.

E.2. Scaling Analysis

A critical property of an effective world model is its ability to scale predictably with increased model capacity and training data. We evaluate this property with three controlled studies. For **model scaling**, we keep the pretraining dataset fixed and vary model capacity from 342M to 568M and 2.59B parameters. For **data scaling**, we fix the 2.59B model and train on 5%, 20%, and 100% of the TraceExtract dataset. For **action-head scaling**, we train downstream action heads at two capacities and compare policies with and

Table 6: Ablation Studies. We evaluate the effect of individual design choices and input modalities on trace prediction quality.

	top5-DTW ↓		
Model Variant	$T = 8$	16	32
Architecture Variations			
Full μ_0	**0.127**	**0.187**	**0.223**
w/o B-spline (Raw Trace)	0.156	0.222	0.258
w/o DINOv2 features	0.139	0.193	0.230
w/o Rigidity Loss	0.138	0.193	0.227
Input Robustness			
w/ Depth & Trace history	0.107	0.160	0.203
w/o Depth	0.112	0.168	0.207
w/o Trace history	0.126	0.183	0.224
w/o Depth & Trace history	0.127	0.187	0.223

without frozen trace features. Table 7 reports the full model- and data-scaling results, and Table 8 reports the action-head scaling comparison.

Data scaling. When the model size is fixed at 2.59B parameters, increasing the pretraining set from 5% to 100% improves top5-DTW from 0.134/0.200/0.235 to 0.127/0.187/0.223 for T=8/16/32. The gains are most consistent at longer horizons, where more diverse interaction videos help the model predict temporally extended motion rather than only short-term displacement.

Model scaling. With the full dataset fixed, larger models improve trace prediction across all horizons: the 342M model obtains 0.143/0.205/0.240, the 568M model improves to 0.136/0.191/0.227, and the 2.59B model reaches 0.127/0.187/0.223. This monotonic trend indicates that the trace-prediction objective remains capacity-limited at our current scale.

Action-head scaling. Table 8 shows that frozen trace features improve downstream policy learning for both action-head sizes. With a 200M action head, using trace features raises success from 10.675% to 25.625%; with a 400M action head, the gain narrows from 28.25% to 30.25%. The larger gap at the smaller 200M head suggests that limited action-head capacity benefits most from trace-space features that provide structured motion information, whereas a larger head can recover much of this signal on its own.

F. Additional Related Work Discussion

Extended Discussion on Embodiment-Agnostic World Models. While the main text briefly outlines the limitations of existing world models, we provide a more granular breakdown here. Pixel-space video models (Wu et al., 2024, Guo et al., 2026) and world-action models (Li et al., 2026b, Ye et al., 2026b,a) excel at learning broad visual dynamics. However, they expend the majority of their representational capacity on dense appearance details that are often irrelevant to the geometry and contact structure required for manipulation.

To alleviate this burden, intermediate representations have been explored (Jang et al., 2026, Zhou et al., 2025b, Hu et al., 2025b, Gu et al., 2024, Bharadhwaj et al., 2024, Wen et al., 2024, Vecerik et al., 2024, Kambara et al., 2026, Zhi et al., 2025, Wang et al., 2026c), yet each distinct prior carries inherent trade-offs:

Table 7: Scaling Analysis. Evaluating the performance impact when scaling model parameters and training data volume.

	top5-DTW ↓		
Scale Factor	**T=8**	**16**	**32**
Model Scaling (100% Data)			
342M Model	0.143	0.205	0.240
568M Model	0.136	0.191	0.227
2.59B Model	**0.127**	**0.187**	**0.223**
Data Scaling (2.59B Model)			
5% Dataset	0.134	0.200	0.235
20% Dataset	0.138	0.195	0.227
100% Dataset	**0.127**	**0.187**	**0.223**

Table 8: Action Head Scaling. μ_0 with an action expert remains robust across action-head capacities compared to *w/o Trace*.

Model Variant	**Success Rate (%) ↑**
200M Action Head	
w/o Trace	10.675
μ_0 + action expert (Ours)	**25.625**
400M Action Head	
w/o Trace	28.25
μ_0 + action expert (Ours)	**30.25**

(1) *Latent features* avoid pixel reconstruction but are notoriously difficult to inspect, control, or translate into precise robot motion. (2) *2D tracks and optical flow* provide a more geometric interface but lack metric depth, often obscuring crucial 3D contacts and spatial object motion. (3) *Recent 3D flow methods* restore metric structure but typically predict dense flow fields over fixed grids (wasting compute budget on static backgrounds), condition rollouts on labeled actions, or relegate motion to an auxiliary policy prior rather than building a standalone world model. μ_0 bypasses these issues by making a sparse set of semantic 3D traces the explicit prediction target. This yields a world-model output that is compact, metric, and directly reusable as a motion interface for downstream policies.

Extended Comparison: Trace Representations and TraceGen. As noted in the main text, visual motion plans for manipulation generally fall into three families: VLM-based waypoints (Li et al., 2025, Zhou et al., 2025a, Yuan et al., 2024b, Yang et al., 2025), post-hoc track extraction from generated video (Ko et al., 2024, Bharadhwaj et al., 2025, Li et al., 2026a, Dharmarajan et al., 2026), and direct track prediction via diffusion or flow matching (Nguyen et al., 2026, Gao et al., 2025, Lin et al., 2026). While these choices provide useful auxiliary guidance, they are less suited to learning a universal dynamics model because raw action semantics often depend heavily on specific robot kinematics and control frequencies.

TraceGen (Lee et al., 2026) represents the closest prior work to ours, but μ_0 introduces fundamental changes to both the supervision pipeline and the model interface. Specifically, TraceGen relies on fixed-grid traces over short clips, necessitates depth information at inference time, and utilizes a hand-designed trace replay mechanism. It does not provide a reusable, query-conditioned 3D world model. In contrast, μ_0 addresses these limitations end-to-end: First, our data pipeline, TraceExtract, replaces fixed grids with semantic interaction keypoints, global 3D tracking, event-level captions, and movement filtering. Second, the Trace Expert in μ_0 predicts query-conditioned B-spline futures via semantic flow matching. Finally, our Action Expert directly consumes frozen trace-denoising features rather than relying on raw trace replay. Through this design, actionable 3D traces are elevated from merely an auxiliary visual cue to the central, video-pretrained motion interface that drives cross-embodiment manipulation.