Point Tracking Improves World Action Models

Jiarui Guan[1] **Wenshuai Zhao**[1,2,†] **Yue Pei**[6]
Ziliang Chen[4,5] **Arno Solin**[1,2] **Juho Kannala**[1,3]
[1]Aalto University [2]ELLIS Institute Finland [3]University of Oulu
[4]Sun Yat-sen University [5]Peng Cheng Laboratory [6]Beihang University
{jiarui.guan, wenshuai.zhao}@aalto.fi
{juho.kannala, arno.solin}@aalto.fi c.ziliang@yahoo.com peiyue@buaa.edu.cn
[†]Corresponding author.

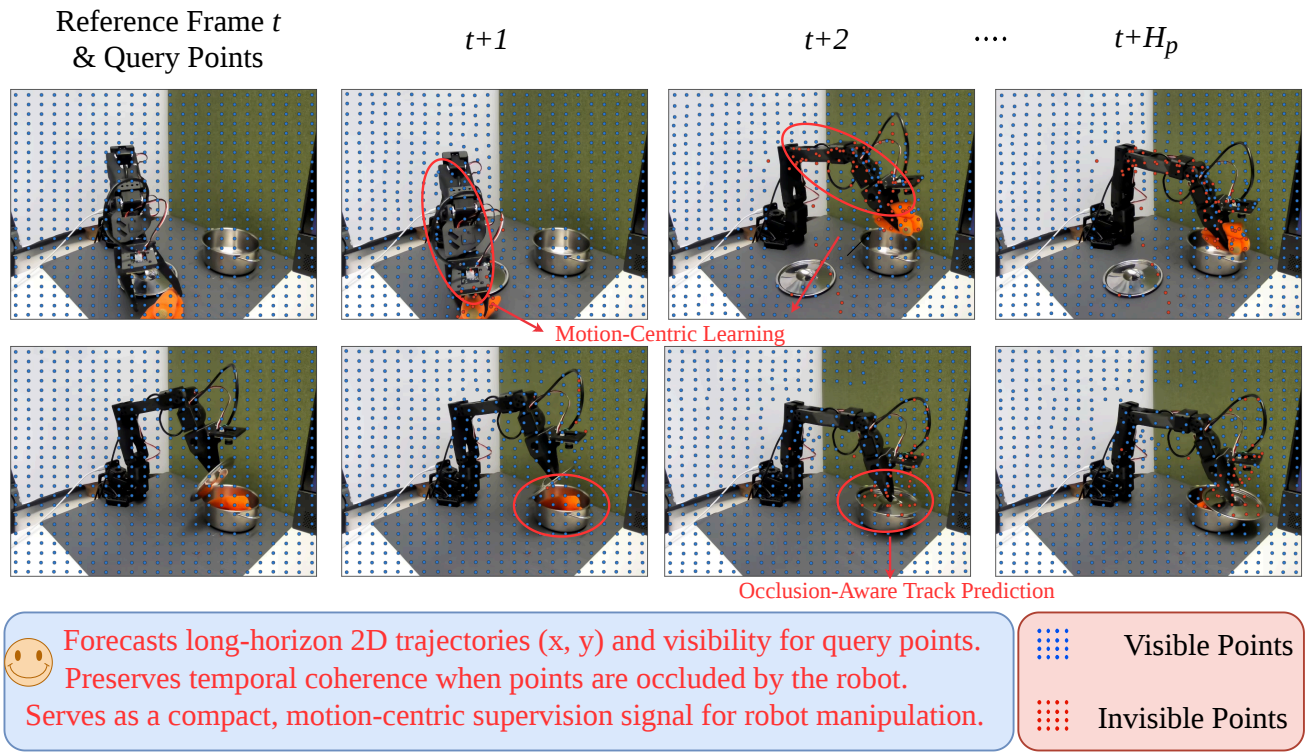

Fig. 1: **JOPAT predicts structured point tracks beyond visible pixels.** Starting from query points on the reference frame, JOPAT forecasts long-horizon 2D trajectories and visibility logits, exposing action-relevant scene motion while representing points that become unobservable or leave the field of view.

***Abstract*—World-action models can transfer video dynamics into robot policies, but pixel-level future prediction entangles controllable motion with lighting, texture, and background variation. We introduce JOPAT, a JOint Pixel-And-Track world-action model that denoises future visual latents, 2D point tracks with visibility, and robot actions in a shared diffusion transformer. Tracks expose correspondence-level motion and missing-evidence state, while visual latents preserve semantic grounding. On 40 LIBERO tasks, JOPAT reaches 97.8% average success and obtains the best Long-suite result. It also improves real-robot LeRobot performance, OOD robustness, and low-data transfer from action-free video.**

I. Introduction

Robot policies must act through occlusion, long-horizon dependencies, and visual distractors. Vision-language-action models benefit from pretrained visual-language representations [1, 2], but they are not primarily trained to model how actions change the world. World-action models couple action prediction with future-state prediction [3]; however, pixel or visual-token futures leave action-relevant variables implicit: which point moved, where it moved, and whether it became hidden.

We propose JOPAT, a unified diffusion transformer whose future state combines visual latents, 2D point tracks, and visibility. Visual latents preserve semantic grounding, while point tracks expose persistent correspondences and missing-evidence state. Figure 1 illustrates the key interface: instead of predicting only pixels, JOPAT forecasts where reference-frame query points move and when they become hidden. This makes

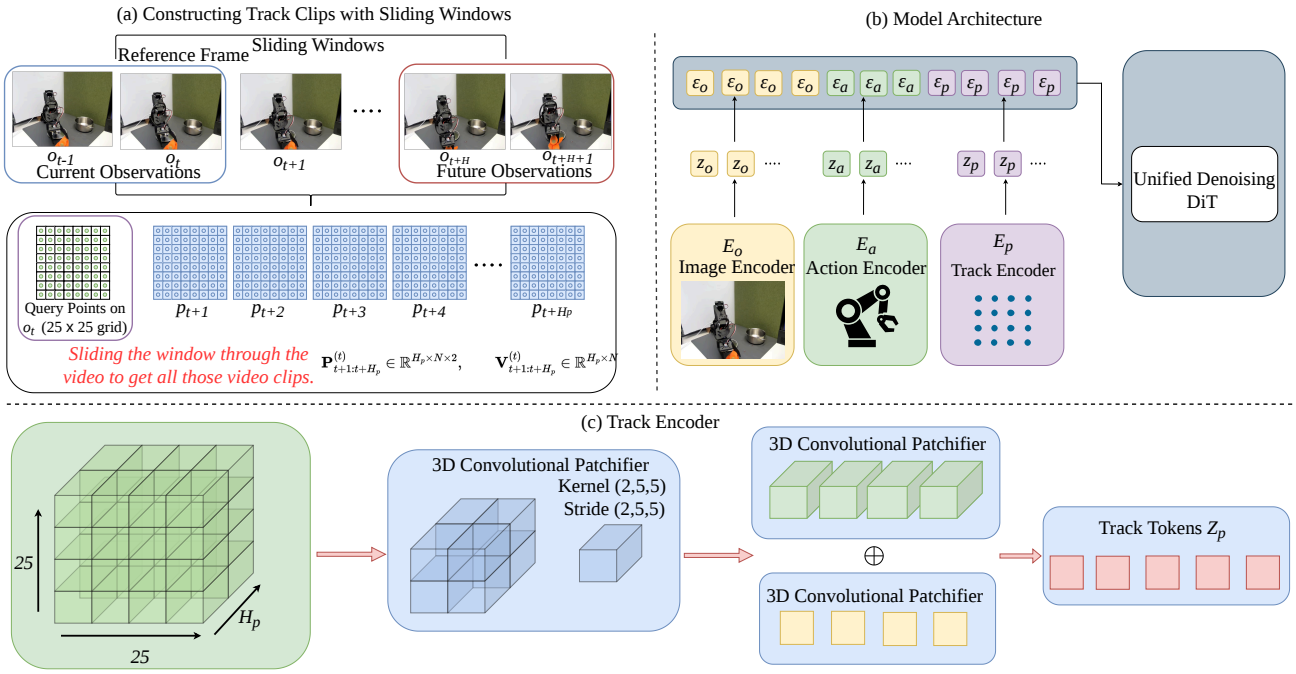

Fig. 2: **Overview of JOPAT. (a)** Sliding-window track construction anchors grid query points on the current frame. **(b)** JOPAT jointly denoises future visual latents, point-track coordinates, and robot actions in a shared Transformer. **(c)** The track encoder treats tracks as a spatiotemporal grid and predicts coordinate noise and visibility logits.

object displacement, contact-induced motion, occlusion, and out-of-frame transitions explicit while retaining appearance and affordance cues through visual latents. Action-free videos supervise this pixel-track future, and action-labeled demonstrations ground it in executable control.

II. Method

Figure 2 summarizes the unified pixel-track-action architecture. At time t, JOPAT conditions on the two most recent RGB frames and predicts an action chunk together with future visual latents and point tracks. The observation encoder produces a global conditioning feature $\mathbf{c}_t$, which is injected into each DiT block through AdaLN together with modality-specific diffusion timesteps. Noisy action tokens, visual-latent tokens, track tokens, and register tokens are concatenated into a single denoising sequence, $\mathbf{Z} = [\tilde{\mathbf{z}}^a_{t:t+K-1}, \tilde{\mathbf{z}}^o_{t+H:t+H+1}, \tilde{\mathbf{z}}^p_{t:t+H_p-1}, \mathbf{r}]$. A DiT-style transformer applies bidirectional full attention over $\mathbf{Z}$ and predicts action noise, visual-latent noise, track-coordinate noise, and visibility logits. Thus, actions interact directly with sampled future pixels and tracks rather than using tracks as a detached auxiliary loss.

For track construction, each sliding window uses the latest observation o_t as the reference frame for a 25×25 grid of query points. CoTrackerV3 [4] provides 2D trajectories $\mathbf{P}^{(t)} \in \mathbb{R}^{H_p \times N \times 2}$ and binary visibility labels $\mathbf{V}^{(t)} \in \{0,1\}^{H_p \times N}$ over horizon H_p, where $N = 625$. We reshape the point dimension back to its spatial query grid, forming a track-as-video tensor with coordinate channels over time, height, and width. A 3D convolutional patchifier encodes this tensor into track tokens, so nearby query points and adjacent future frames are modeled jointly. The coordinate branch is trained with a diffusion loss, while the visibility branch is supervised with BCE and is predicted only as an output, preventing leakage of ground-truth visibility into the track encoder.

Training uses the same model for action-free video and action-labeled robot demonstrations. For each modality, Gaussian noise is added at an independently sampled diffusion timestep, and the model predicts the corresponding denoising target. On demonstrations, the objective combines action, visual-latent, track-coordinate, and visibility losses; on action-free videos, the action branch is masked and only the pixel-track future state is supervised. At inference, action, visual, and track tokens are initialized from Gaussian noise and denoised together with DDIM sampling [5]. Only the action chunk is executed, but the visual and track tokens remain active latent future-state variables during sampling, so each action is generated under an implicitly sampled pixel-track future. Appendix A-A provides the full hyperparameters, compute resources, and deployment details.

III. Experiments

We evaluate the state-interface hypothesis along four axes: main benchmark performance, representation design, robustness under distribution shift, and scalable action-free pretraining. Simulation uses LIBERO [17], including the *Spatial*, *Object*, *Goal*, and *Long* suites with 10 tasks and 50 demonstrations per task. JOPAT is pretrained on LIBERO-90 for 100K steps and finetuned on each target suite for 10K steps. Real-world experiments use the LeRobot SO-101 platform with 50 demonstrations per task after DROID [18] action-free pretraining. JOPAT uses a 12-layer DiT with hidden size 768, 12 attention heads, offset $H = 16$, and action/track horizon $K = H_p = 19$.

TABLE I: **Simulation experimental results.** Comparison of task success rates (SR) and ranks (RK) on the LIBERO benchmark across four task types. "†" indicates our reproduced results.

Method	Spatial		Object		Goal		Long		Average	
	SR ↑	RK ↓	SR ↑	RK ↓	SR ↑	RK ↓	SR ↑	RK ↓	SR ↑	RK ↓
Diffusion Policy [6]	78.3	15	92.5	10	68.3	15	50.5	15	72.4	15
Octo fine-tuned [7]	78.9	14	85.7	15	84.6	12	51.1	14	75.1	14
OpenVLA [2]	84.7	11	88.4	14	79.2	13	53.7	13	76.5	13
π_0 fine-tuned [1]	96.8	5	98.8	2	95.8	4	85.2	7	94.2	7
π_0-Fast [8]	96.4	6	96.8	7	88.6	8	60.2	11	85.5	8
$\pi_{0.5}$-KI [9]	98.0	2	97.8	6	95.6	5	85.8	6	96.0	4
OpenVLA-OFT [10]	97.6	3	98.4	4	97.9	2	94.5	3	97.1	3
SpatialVLA [11]	88.2	9	89.9	13	78.6	14	55.5	12	78.1	12
PD-VLA† [12]	95.5	7	96.7	8	94.9	7	91.7	4	94.7	5
STAR [13]	95.5	7	98.3	5	95.0	6	88.5	5	94.3	6
Dita [14]	84.2	12	96.3	9	85.4	11	63.8	10	82.4	11
CoT-VLA [15]	87.5	10	91.6	12	87.6	9	69.0	9	83.9	10
CogVLA [16]	**98.6**	**1**	98.8	2	96.6	3	95.4	2	97.4	2
UWM [3]	82.3	13	92.2	11	86.8	10	77.6	8	84.7	9
JOPAT	97.2	4	**98.9**	**1**	**98.4**	**1**	**96.4**	**1**	**97.8**	**1**

A. Simulation results

Table I reports the main LIBERO results. JOPAT achieves the best average SR and obtains the strongest *Long* result, the suite where maintaining action-relevant state over temporally extended interactions is most important. This supports the motivation of JOPAT: point tracks provide a compact motion-centric interface for long-horizon dynamics, while visual latents retain semantic and contextual information needed for task grounding. Compared with UWM, the gain is especially large on *Long*, indicating that explicit correspondences help beyond pixel-centric future prediction. The consistent ranking across suites suggests that the track-augmented state is not specialized to one task family. Instead, track prediction appears to provide a general motion prior that complements visual semantics across spatial, object, goal, and long-horizon settings.

B. Real-world experiments

We further evaluate four LeRobot SO-101 tasks: *Cook-Soup* tests long-horizon sequencing under repeated interactions, *Insert-Peg* tests precision alignment and contact-sensitive execution, *Push-Tomato* tests contact-rich object displacement, and *Pick-Grocery* tests object-level OOD generalization. These tasks introduce unmodeled contact dynamics, arm-induced occlusion, lighting variation, and object-layout changes that are absent from the simulator. All methods use the same camera setup, demonstration budget, and rollout protocol so that differences reflect the learned policy representation. Appendix A-B gives the full real-robot training, inference, and evaluation protocol, and Figure 5 visualizes the task setup.

Table II shows that JOPAT achieves the best average real-robot SR, outperforming ACT and UWM by 17.5 and 25.0 points. Against ACT, the largest gain appears on *Push-Tomato*, where JOPAT reaches 100% SR, suggesting that explicit track prediction provides a useful motion interface for contact-rich displacement. JOPAT also improves strongly on *Cook-Soup*, consistent with better robustness to occlusion and long-horizon state drift. *Insert-Peg* remains difficult for all methods, likely requiring finer 3D geometry or contact-aware sensing beyond RGB-based track-and-latent modeling.

TABLE II: **Real-robot performance on LeRobot.** SR on four real-world tasks are reported.

Method	Cook	Peg	Tomato	Grocery	Avg.
ACT [19]	40%	0%	70%	50%	40%
UWM [3]	10%	0%	80%	40%	32.5%
JOPAT (Ours)	**60%**	**10%**	**100%**	**60%**	**57.5%**

C. Ablation on joint pixel-track modeling

We compare three ablations under the same data and optimization protocol: **Latent-only** removes track prediction, **Track-only** removes visual-latent prediction, and **Joint** predicts visual latents, tracks, visibility, and actions with the shared denoising transformer. This isolates whether the gains come from the pixel-track future-state interface rather than from data, model size, or optimization changes. Since all variants share the backbone and pretraining schedule, performance differences reflect which future modalities are available during denoising.

Table III shows that joint modeling improves over Latent-only by 19.0 points and Track-only by 70.2 points on LIBERO-Long. Latent-only retains object appearance but leaves contact-relevant motion implicit, while Track-only exposes motion structure but lacks the visual grounding needed to identify the correct object, affordance, or task state. The joint model succeeds by coupling semantic context and correspondence-level motion in the same denoising state.

The same effect appears in Table IV, where Joint reaches 57.5% real-robot SR, improving average performance by 22.5 points over Latent-only and 42.5 points over Track-only. JOPAT improves every task under the same data budget, although *Insert-Peg* remains difficult for all variants.

TABLE III: **Joint vs. single-modality ablation on LIBERO-Long.** Success rates (SR, %) across ten long-horizon tasks. All variants are pretrained on DROID action-free videos under identical settings.

Task	Latent-only	Track-only	Joint	Task	Latent-only	Track-only	Joint
Mug-Mug	58	2	**92**	Tomato-Sauce	93	48	**100**
Soup-Cheese	78	14	**94**	Cream-Butter	91	66	**100**
Moka-Moka	67	26	**89**	Microwave	94	32	**100**
Book-Candy	63	14	**100**	Chocolate-Pudding	73	46	**91**
Bowl-Drawer	95	2	**100**	Moka-Stove	62	12	**98**
Average SR: Latent-only 77.4		Track-only 26.2		Joint **96.4**			

TABLE IV: **Real-robot modality ablation.** SR are reported on four LeRobot SO-101 tasks.

Method	Cook-Soup	Insert-Peg	Push-Tomato	Pick-Grocery	Avg.
Latent-only	40	0	70	30	35.0
Track-only	10	0	30	20	15.0
Joint (Ours)	**60**	**10**	**100**	**60**	**57.5**

D. Visibility, pretraining, and OOD robustness

We next test whether the point-track interface preserves object correspondence and motion intent under incomplete perception, action-free supervision, and distribution shift.

Table V shows that visibility improves real-robot average SR from 47.5% to 57.5%, mainly helping occlusion-heavy tasks. This supports the role of visibility supervision as a missing-evidence constraint: the model learns when a correspondence is unobservable instead of forcing every point to explain a visible coordinate. The unchanged *Pick-Grocery* result suggests that visibility helps most when self-occlusion, transient out-of-view motion, or long-horizon object motion would otherwise cause track drift.

Table VI isolates the effect of action-free video pretraining on LIBERO-Long. Both DROID and OpenVid-1M pretraining outperform training from scratch across demonstration budgets, showing that the pixel-track future-state objective can turn action-free videos into useful motion priors. The low-data setting shows the clearest practical benefit: with only 10 demonstrations, DROID improves SR from 11.9% to 64.2%, while generic OpenVid-1M still reaches 48.5%. Appendix A-C further reports real-robot DROID pretraining and horizon sensitivity analyses in Figures 3 and 4.

For OOD robustness, we modify LIBERO-Long with three changes: the object initialization range is expanded by 5 cm, one background object is replaced with an unseen object, and an unseen distractor is placed in front of the target object. This setting increases failures caused by occlusion, distractor interference, larger object displacement, and delayed errors in long-horizon execution.

Table VII shows the best average OOD SR, improving over Diffusion Policy, UWM, and fine-tuned $\pi_{0.5}$ by 34, 32, and 13 points. Compared with UWM, JOPAT improves on all five tasks, with especially large gains on *Book-Candy*, *Mug-Mug*, and *Soup-Cheese*. Fine-tuned $\pi_{0.5}$ achieves strong peak performance on several tasks but is less stable across task families; JOPAT is not best on every individual task, yet gives the strongest average under distractors and occlusions.

TABLE V: **Track visibility ablation.** We report the SR on LeRobot SO-101 tasks. The visibility head improves robustness under self-occlusion and transient out-of-view motion.

Method	Cook	Peg	Tomato	Grocery	Avg.
w/o visibility	40	0	90	60	47.5
w/ visibility (Ours)	**60**	**10**	**100**	**60**	**57.5**

TABLE VI: **Action-free video pretraining on LIBERO-Long.** SR are reported.

Method	10 demos	25 demos	50 demos
JOPAT w/o PT	11.9%	31.6%	66.1%
JOPAT (DROID PT)	**64.2**%	82.7%	**96.4**%
JOPAT (OpenVid-1M PT)	48.5%	**84.6**%	95.1%

TABLE VII: **OOD evaluation on modified LIBERO-Long.** SR are reported.

Task	DP	$\pi_{0.5}$	UWM	JOPAT
Book-Candy	0.24	**0.95**	0.38	0.88
Soup-Cheese	0.21	**0.82**	0.44	0.78
Bowl-Drawer	0.56	0.12	0.52	**0.68**
Moka-Moka	0.44	**0.70**	0.22	0.48
Mug-Mug	0.18	0.08	0.14	**0.50**
Avg.	0.32	0.53	0.34	**0.66**

IV. Conclusion and Discussion

We introduced JOPAT, a unified world-action model that jointly predicts visual latents, 2D point tracks, and actions. Explicit correspondences expose action-relevant motion and occlusion state, improving robustness over pixel-centric futures. Across LIBERO and real-robot LeRobot tasks, JOPAT improves performance, data efficiency, and transfer from action-free video.

a) Limitations: Grid-based tracks may miss fine-grained deformations needed for sub-centimeter dexterity, and off-the-shelf trackers (e.g., CoTracker) impose a supervision ceiling when they fail to capture the relevant dynamics.

Acknowledgments

AS and JK acknowledge funding from the Research Council of Finland (352788, 362407, 373999, 373780, 362408, and 339730). This work was supported by the Research Council of Finland Flagship programme: Finnish Center for Artificial Intelligence FCAI. We acknowledge the computational resources provided by the Aalto Science-IT project.

References

[1] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2026. URL https://arxiv.org/abs/2410.24164.

[2] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model, 2024. URL https://arxiv.org/abs/2406.09246.

[3] Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets, 2025. URL https://arxiv.org/abs/2504.02792.

[4] Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos, 2024. URL https://arxiv.org/abs/2410.11831.

[5] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022. URL https://arxiv.org/abs/2010.02502.

[6] 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, 2024. URL https://arxiv.org/abs/2303.04137.

[7] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy, 2024. URL https://arxiv.org/abs/2405.12213.

[8] 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, 2025. URL https://arxiv.org/abs/2501.09747.

[9] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025. URL https://arxiv.org/abs/2504.16054.

[10] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success, 2025. URL https://arxiv.org/abs/2502.19645.

[11] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, and Xuelong Li. Spatialvla: Exploring spatial representations for visual-language-action model, 2025. URL https://arxiv.org/abs/2501.15830.

[12] Wenxuan Song, Jiayi Chen, Pengxiang Ding, Han Zhao, Wei Zhao, Zhide Zhong, Zongyuan Ge, Jun Ma, and Haoang Li. Accelerating vision-language-action model integrated with action chunking via parallel decoding, 2025. URL https://arxiv.org/abs/2503.02310.

[13] Hao Li, Qi Lv, Rui Shao, Xiang Deng, Yinchuan Li, Jianye Hao, and Liqiang Nie. Star: Learning diverse robot skill abstractions through rotation-augmented vector quantization, 2025. URL https://arxiv.org/abs/2506.03863.

[14] Zhi Hou, Tianyi Zhang, Yuwen Xiong, Haonan Duan, Hengjun Pu, Ronglei Tong, Chengyang Zhao, Xizhou Zhu, Yu Qiao, Jifeng Dai, and Yuntao Chen. Dita: Scaling diffusion transformer for generalist vision-language-action policy, 2025. URL https://arxiv.org/abs/2503.19757.

[15] Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, Ankur Handa, Ming-Yu Liu, Donglai Xiang, Gordon Wetzstein, and Tsung-Yi Lin. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models, 2025. URL https://arxiv.org/abs/2503.22020.

[16] Wei Li, Renshan Zhang, Rui Shao, Jie He, and Liqiang Nie. CogVLA: Cognition-aligned vision-language-action model via instruction-driven routing & sparsification. *arXiv preprint, arxiv:2508.21046*, 2025.

[17] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

[18] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

[19] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware, 2023. URL https://arxiv.org/abs/2304.13705.

[20] Dustin Podell, Zion English, Kyle Lacey, Andreas

Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis, 2023. URL https://arxiv.org/abs/2307.01952.

[21] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

APPENDIX A
SUPPLEMENTARY MATERIAL

This appendix provides implementation details and supplementary analyses that complement the main paper. We include the real-robot experimental protocol, the real-robot action-free pretraining result, prediction-horizon sensitivity, and task-level failure modes.

A. Implementation details

a) Architecture: JOPAT uses a DiT-style transformer backbone with modality-specific encoders and decoders for visual latents, point tracks, and actions. The conditioning encoder processes the two most recent RGB observations and injects the resulting global feature into each Transformer block through AdaLN. Future observations are encoded with a frozen SDXL VAE, while point tracks are represented on a regular query grid and patchified with a 3D convolutional encoder. The Transformer output is split into modality-specific heads for action noise, visual-latent noise, track-coordinate noise, and visibility logits, matching the unified pixel-track-action architecture in Section II and the track construction in Section II.

b) Training data: For action-free videos, the action branch is masked and the model is trained only with visual-latent, track-coordinate, and visibility losses. For action-labeled robot demonstrations, all branches are supervised. Point-track and visibility targets are extracted with CoTrackerV3 [4] using the sliding-window construction described in Section II. Simulation and real-robot experiments use the same joint denoising formulation, with dataset-specific pretraining and finetuning protocols described in the corresponding experiment sections.

c) Inference: At test time, JOPAT performs receding-horizon action generation with DDIM sampling. Action, visual, and track tokens are initialized from Gaussian noise and denoised conditioned on the current observation history. The model can also sample future visual latents and tracks for analysis, but policy execution uses only the decoded action chunk; the first 8 actions are executed before replanning.

d) Compute resources.: Action-free pretraining was performed on 4 NVIDIA H200 GPUs for approximately 5 days. Task-specific finetuning used 1 NVIDIA H200 GPU for approximately 1 day. During deployment, inference runs on a single NVIDIA RTX 4090 GPU at approximately 10 Hz.

B. Real-robot experimental protocol

a) Platform and tasks: We evaluate on the LeRobot SO-101 platform with four manipulation tasks: *Cook-Soup*, which requires long-horizon sequencing and repeated object interactions; *Insert-Peg*, which requires precise geometric alignment and contact-sensitive execution; *Push-Tomato*, which tests contact-rich object displacement under pose and interaction variation; and *Pick-Grocery*, which tests object-level OOD generalization to unseen objects. Together, these tasks evaluate robustness to partial observability, arm-object occlusion, contact-induced dynamics, and distribution shift.

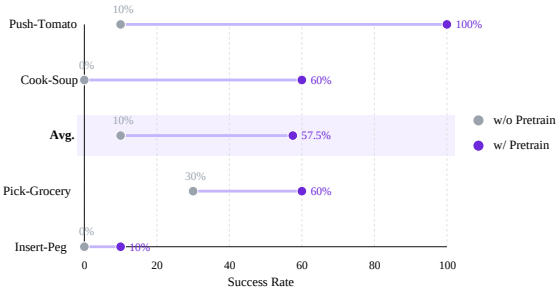

Fig. 3: **Action-free pretraining ablation.** Average real-robot success rate with and without DROID action-free video pretraining.

b) Training data and finetuning: For real-robot transfer, JOPAT is first pretrained on DROID action-free videos [18]. During action-free pretraining, the action branch is masked and only the visual-latent, track-coordinate, and visibility objectives are applied. Point tracks and visibility labels are extracted with CoTrackerV3 [4] using the same sliding-window construction as in the simulation experiments. The pretrained model is then finetuned separately on each LeRobot task using 50 action-labeled demonstrations. During finetuning, all branches are supervised so that the pretrained pixel-track motion priors are aligned with executable SO-101 actions.

c) Inference and evaluation: At test time, JOPAT uses receding-horizon action generation with DDIM sampling. The policy conditions on the two most recent RGB observations, denoises action, visual, and track tokens jointly, and executes only the decoded action chunk. Unless otherwise stated, we use a future-observation offset of $H = 16$, an action/track horizon of 19 steps, 10 DDIM denoising steps, and execute the first 8 actions before replanning. We report success rate over 10 evaluation rollouts per task for each method.

d) Task setup: Figure 5 visualizes the four LeRobot SO-101 tasks. Each column corresponds to one task, and the three rows show the initial configuration, an intermediate state, and a successful final state.

C. Additional ablations

The main paper reports the LIBERO-Long joint-versus-single-modality ablation in Table III, the real-robot modality ablation in Table IV, the visibility ablation in Table V, and the LIBERO-Long pretraining scaling study in Table VI. Here we provide two supplementary real-robot analyses: the effect of DROID action-free video pretraining and the sensitivity to the future-observation offset. Both follow the real-robot protocol in Appendix A-B.

1) Action-free pretraining: Figure 3 summarizes the effect of action-free video pretraining. With the same 50 real-robot finetuning demonstrations per task, DROID pretraining improves the average success rate from 10.0% to 57.5%. The largest gains occur on Push-Tomato and Cook-Soup, suggesting that action-free videos provide useful priors for object motion, occlusion handling, and contact-induced dynamics. Insert-Peg remains difficult, indicating that high-precision

TABLE VIII: **JOPAT hyperparameters.** Unless otherwise stated, the same configuration is used for simulation and real-robot experiments.

Parameter	Value
Data and representation	
Observation history	2 RGB frames
Image resolution	224×224
Future-observation offset	$H = 16$ steps
Action / track horizon	$K = H_p = 19$ steps
Executed action prefix	8 steps
SDXL latent shape	$28 \times 28 \times 4$
Query-point grid	25×25 ($N = 625$)
Track patch size	$(2, 5, 5)$ over time and grid dimensions
Model	
Conditioning encoder	ResNet-18, ImageNet initialization
Visual target encoder	Frozen SDXL VAE [20]
Transformer depth	12 layers
Hidden dimension	768
Attention heads	12
MLP ratio	4
Register tokens	8
Conditioning mechanism	AdaLN
Diffusion and optimization	
Noise schedule	squaredcos_cap_v2
Training diffusion steps	100
Inference denoising steps	10
Sampler	DDIM [5]
Optimizer	AdamW [21]
Learning rate	1×10^{-4}
Weight decay	1×10^{-6}
Adam betas	$(0.9, 0.999)$
Batch size	144 for pretraining; 72 for finetuning
LR schedule	Constant for pretraining; cosine for finetuning
Warmup steps	1000
Loss weights	$\lambda_a = \lambda_o = \lambda_p = \lambda_{\text{vis}} = 1$

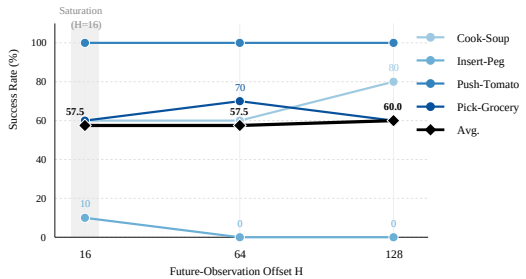

Fig. 4: **Horizon sensitivity.** Average real-robot success rate for different future-observation offsets.

insertion also requires fine-grained 3D geometry or contact sensing beyond 2D tracks and RGB latents.

2) Prediction-horizon sensitivity: Figure 4 shows that moderate future-observation offsets already capture the dynamics needed for most tasks. The $H = 16$ setting matches the performance of longer offsets on average, while avoiding the higher uncertainty associated with very long future-track prediction. Longer offsets can help multi-stage tasks such as Cook-Soup, but may hurt precision-sensitive tasks such as Insert-Peg, where future uncertainty produces noisy supervision for fine contact alignment.

D. Qualitative behavior and failure modes

a) Occlusion and off-screen motion: In qualitative rollouts, the predicted tracks often remain temporally coherent when the robot arm occludes the target or when objects move near the image boundary. This behavior is consistent with the quantitative visibility ablation: the model learns to maintain a belief over hidden object motion rather than relying only on currently visible pixels. Such object permanence is particularly useful for long-horizon tasks in which the policy must continue acting through temporary perceptual dropout.

b) Task-level failure modes: The remaining failures are task dependent:

- **Cook-Soup.** Failures often occur around the pot lid, where specular reflections and unusual lid orientations weaken visual grounding. Stronger illumination augmentation or depth cues may improve robustness.
- **Insert-Peg.** The task requires millimeter-level 3D alignment between the peg and socket. Two-dimensional tracks reduce perceptual drift but do not fully resolve contact geometry, leading to jamming or repeated failed insertions.
- **Push-Tomato.** Failures are usually caused by small localization or approach-angle errors that compound during pushing. Better contact dynamics or closed-loop state estimation may further improve reliability.
- **Pick-Grocery.** Failures mainly reflect limited spatial generalization in clutter. The policy can overfit to frequent grasp locations and struggle when object layouts differ from the demonstration distribution.

Overall, JOPAT is strongest when success depends on

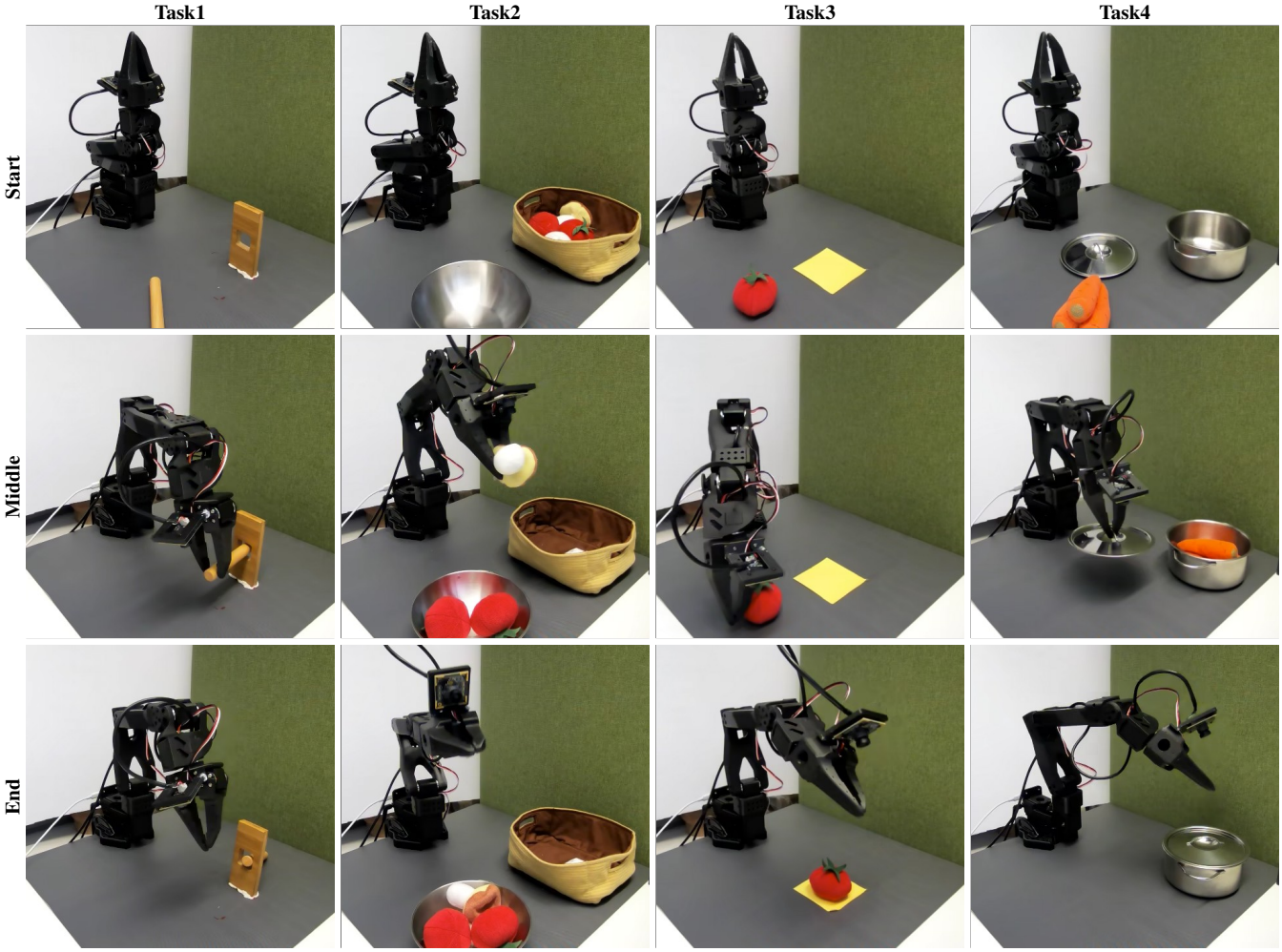

Fig. 5: **Real-robot task setup.** Insert-Peg, Cook-Soup, Push-Tomato, and Pick-Grocery on the LeRobot SO-101 platform. The first row shows the initial configuration, the second row shows an intermediate state, and the third row shows successful task completion.

temporal consistency, object permanence, and robustness to visual dropout. Its main limitation is fine-grained contact geometry, suggesting that future extensions should combine the proposed pixel-track world-action model with explicit 3D sensing, tactile feedback, or contact-aware objectives.