SAMTM: SAM-based Trajectory Modeling for Efficient Policy Learning

Zhangchen Ye
IIIS,Tsinghua University

Mengjie Zhao
IIIS,Tsinghua University

Jiani Wang
IIIS,Tsinghua University

Ruicheng Li
Department of Physics,Tsinghua University

1 Introduction

Robotic manipulation policies are often trained with strong supervision on actions, which limits scalability when action-labeled data is scarce or expensive to obtain. At the same time, large-scale video data, such as human videos from the internet or other robot demonstrations, contains rich information about object motion and task dynamics but lacks explicit action annotations.

Recent work has explored trajectory-based representations (Wen et al., 2023), where future pixel-level movement serves as an intermediate abstraction between perception and control. Such representations are largely embodiment-agnostic and can potentially bridge pre-trained vision models with robot learning.

However, a key limitation of the existing trajectory modeling method is that it only relies on uniformly sampled grid points across the image. In manipulation tasks, most of these points lie on the background and contribute little to task execution, while the motion of the robot's end effector and task-relevant objects is more informative.

In this project, we propose a trajectory modeling pipeline that incorporates semantic segmentation to guide point sampling. By using Grounded SAM 2 (Ren et al., 2024) with text prompts, we obtain masks for robot arms and relevant objects, then sample points within these regions. The predicted trajectories of these points are then used to condition a low-level policy. We evaluate this approach on a standard pick-and-place benchmark and a self-defined dual-arm articulated object task.

2 Related Work

Trajectory, also called point flow representations, have emerged as a promising interface for policy learning. Any-point Trajectory Modeling (ATM) (Wen et al., 2023) formulates policy learning as a two-stage process: first predicting future trajectories of image points, and then using these trajectories to guide action generation. This framework allows learning from diverse data sources without requiring accurate action labels.

Gao et al. (2024) treats flow as an action representation within a world model, performing model-based planning to derive the optimal flow. Gu et al. (2023) further explores using manually drawn or LLM-generated trajectory sketches to provide effective guidance.

Other works treat motion or flow as a universal manipulation interface across domains. Methods such as Hudor Guzey et al. (2025), GenFlowRL (Yu et al., 2025), and HinFlow (Zheng et al., 2025) demonstrate that trajectory or flow representations can be applied to reinforcement learning and online imitation learning. A common insight from these works is that object-centric representations are often more informative than uniform spatial sampling.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

Recent advances in segmentation foundation models further enable object-centric perception. SAM 2 (Ravi et al., 2024) extends the Segment Anything (Kirillov et al., 2023) framework to videos with a memory attention mechanism, making it suitable for online processing and long-horizon tasks. Its text-guided variant, Grounded SAM 2 (Ren et al., 2024), allows segmentation of task-relevant entities using natural language prompts, which is particularly convenient in robotic manipulation scenarios.

Our work builds on these ideas by combining trajectory modeling with task-relevant segmentation, and by evaluating the resulting framework in simulation settings.

3 Methodology: SAM-Based Task-Relevant Trajectory Modeling

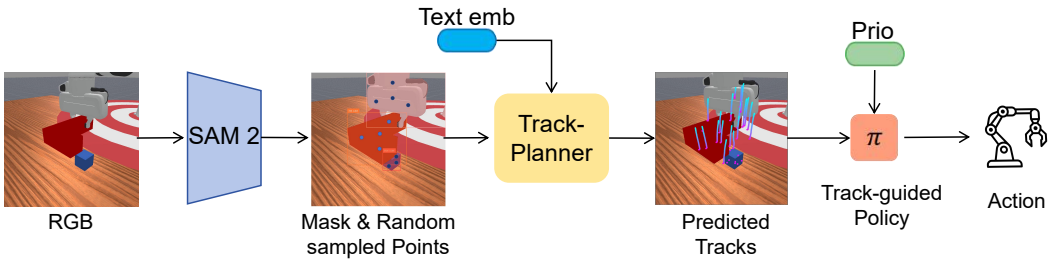

Figure 1: Overall Pipeline.

As shown in Fig 1, our framework follows a hierarchical design consisting of two main components: a Track-Planner (Sec 3.2) and a Track-guided Policy (Sec 3.3). Given RGB observations, the Track-Planner predicts future trajectories of sampled points within task-relavent masks from Grounded SAM 2 (Ren et al., 2024), while the Track-guided Policy maps RGB observations, point tracks, together with proprioception, to robot actions.

3.1 Task-Relevant Point Sampling

Instead of uniformly sampling points on a fixed image grid, we use Grounded SAM 2 to segment task-relevant regions, such as robot grippers and manipulated objects. Within the region of each mask, we randomly sample a fixed number of points, ensuring that all predicted trajectories correspond to semantically meaningful regions. This design significantly reduces the proportion of background points and focuses the representation on task-critical motions. For the wrist-mounted camera, since the task-relevant objects may not consistently appear in this view, we simply use a fixed set of points on a grid. To improve the robustness of the resulting trajectories, we apply this sampling strategy multiple times per image during training phases.

3.2 Track-Planner

The track-planner predicts the future trajectories of task-relevant points based on the current image input o_t. This process can be formally described as: $\mathcal{G}_t = \mathbf{F}(o_t, \mathbf{p}_t)$ where $\mathbf{F}$—parameterized by ξ—denotes the trajectory prediction function, and $\mathbf{p}_t$ represents the current coordinates of the query points. By forecasting the positions of these points over the subsequent H frames, the model ensures that the high-level planner delivers comprehensive guidance for task execution.

To implement the track-planner, we first collect a suite of demonstration videos using the motion planner in ManiSkill3 (Tao et al., 2025). Ground-truth trajectories are extracted via CoTracker3 (Karaev et al., 2024), a state-of-the-art tracking model. These image-trajectory pairs are then stored in an annotated video dataset denoted as $\mathcal{D}_h$.

Following the acquisition of flow annotations, we proceed with the training of the track-planner. Drawing on the Track Transformer architecture proposed in ATM (Wen et al., 2023), we tokenize both the images and query points before feeding them into a multi-modal transformer model. The model is trained using a flow prediction loss, which is formally defined as:

$$\min_{\xi} \mathbb{E}_{(o_t, \mathbf{p}_t^{t+H}) \sim \mathcal{D}_h} \left[\mathcal{L} \left(\mathbf{F}(o_t, \mathbf{p}_t; \xi), \mathbf{p}_{t+1}^{t+H} \right) \right] \tag{1}$$

where $\mathcal{L}$ stands for the flow prediction loss function, and $\mathcal{D}_h$ refers to the annotated video dataset.

3.3 Track-Guided Policy Learning

Upon completing the training of the Track-Planner, we freeze its parameters and train a track-guided policy using an action-labeled dataset $\mathcal{D}_r$. Building on the high-level planner, our goal is to map the implicit motion information encapsulated in the trajectories into an executable low-level policy. This low-level policy is formulated as a track-guided policy $\pi(o_t, \mathbf{F}(o_t, \mathbf{p}_t))$ parameterized by θ.

The architecture of the track-guided policy adheres to a transformer-based design (Kim et al., 2021; Wen et al., 2023). It takes RGB observations, predicted trajectories, and the robot's proprioceptive state as inputs, and outputs low-level control actions. Specifically, images and proprioceptive data are encoded into spatial tokens via a spatial transformer. Track tokens are then fused with these spatial tokens to fully integrate the guiding information from the trajectories. The combined token set is subsequently passed through a series of MLPs to generate control actions. The optimization objective is formally expressed as:

$$\min_{\theta} \mathbb{E}_{(o_t, a_t)\sim\mathcal{D}_r} \left[\mathcal{L}_2 \left(\pi(o_t, \mathbf{F}(o_t, \mathbf{p}_t); \theta), a_t\right)\right] \tag{2}$$

where $\mathcal{L}_2$ denotes the action prediction loss function, and $\mathcal{D}_r$ represents the action-labeled dataset.

Furthermore, we integrate action chunking and temporal ensemble into policy learning — two techniques widely adopted in prior work (Zhao et al., 2023; Chi et al., 2023; Zhao et al., 2024). Specifically, the agent generates the next k actions every k steps based on current observations and flow predictions, while our temporal ensemble mechanism computes a weighted average of these predictions using an exponential weighting scheme: $w_i = \exp(-m \times i)$.

4 Experiment Results

4.1 Experimental Setup

We conduct all experiments in ManiSkill3 (Tao et al., 2025), which provides GPU-parallelized simulation, flexible task configuration, and support for diverse robot models, such as Franka Emika Panda arm and SO-101 dual-arm (Cadene et al., 2024).

Tasks. Our experiments includes multiple tasks:

- A pick-and-place in implemented by ManiSkill3 benchmark. (Sec 4.2)
- A custom task involving a SO-101 dual-arm robot with an articulated drawer. (Sec 4.3)
- SO-101 dual-arm benchmark tasks required by the project. (Sec 4.4)

Metrics. For simulation experiments, Performance is primarily measured by task success rate over 40 evaluation episodes across random initializations. For real-world deployment, success rates are computed over a fixed number of trials following the course evaluation protocol.

4.2 Pick-and-Place

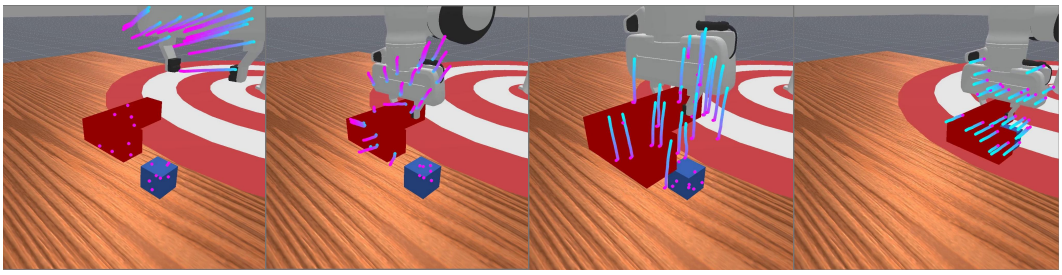

Figure 2: Visualizations of the *pullcubetool* task.

This is a standard manipulation task implemented by ManiSkill3. As show in Fig 2, the task involves a Franka Panda arm picking up the "L" shape tool to pull the blue cube to the circle range. The

robot's observations consist of images from an external camera and a wrist-mounted camera (both at 128x128 resolution), along with its end effector pose as proprioceptive states. We sample 32 points in total, with 16 points on the robot gripper, 8 points on the stick, and 8 points on the cube.

Our baselines include: behaviour cloning (BC) and ATM.

BC denotes traditional behavior cloning strategy, which trains an observation-to-action policy only on the limited action-labeled demonstrations. It shares the same model architecture as ATM except that the predicted flow is zero-out in both training and evaluation.

ATM first pretrains a track transformer on large-scale video dataset to predict the future flow of the query points. The official implementation queries a set of 32 points on a fixed grid.

We use the motion planner provided by ManiSkill3 (Tao et al., 2025) to collect 250 demos. We use dataset mixure following ATM (Wen et al., 2023), where the number of videos data is 50 times that of action-labeled demonstrations. Therefore, 250 videos are used to train Track-Planner and 5 action-labeled demonstrations are used to train Track-guided Policy for both ATM and our method. For BC, we only use 5 action-labeled demonstrations for training. The simulation evaluation results are shown in table 1.

Table 1: Pick-and-Place Task Success Rates

Method	BC	ATM	Ours
Success Rate ↑	0.225	0.550	0.650

On this task, trajectory-guided policies consistently outperform standard behavior cloning under the same policy architecture. Both ATM-style and our task-relevant trajectory methods show higher success rates when only a small number of action-labeled demonstrations are available. Moreover, our method achieves better performance than uniform grid-based trajectory modeling, indicating that focusing on task-relevant regions leads to more informative guidance.

4.3 Self-Deifined Dual-Arm Task

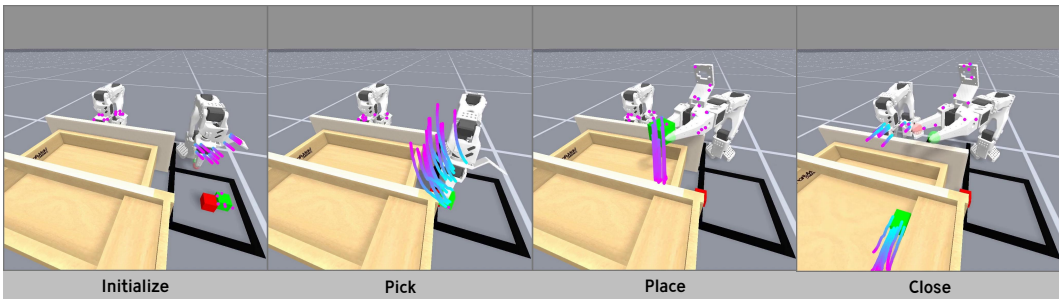

Figure 3: Visualizations of the self-defined task

This is a custom task involving a SO-101 dual-arm robot interacting with an articulated drawer. The right arm need to pick up target object in the table, put it in the drawer, and then the left arm need to close the drawer. We develop a series of tasks to include different colors and shapes of target objects distinguished by task instruction. The instruction-related subtasks includes manipulating three categories of objects: green cube, green sphere, and red cube.

For this dual-arm task, the robot's observations includes one external camera and a wrist-mounted camera on its right arm. All cameras capture images at a resolution of 128×128, complemented by the joint pose as the proprioceptive states and a language embedding processed by a pre-trained BERT (Devlin et al., 2019).

For self-defined dual-arm task, we use totally 150 videos of all 3 tasks to train Track-Planner and only 50 action-labeled demonstrations of green cube task to train Track-guided Policy. Our method can solve this task with 55% success rate in simulation environment.

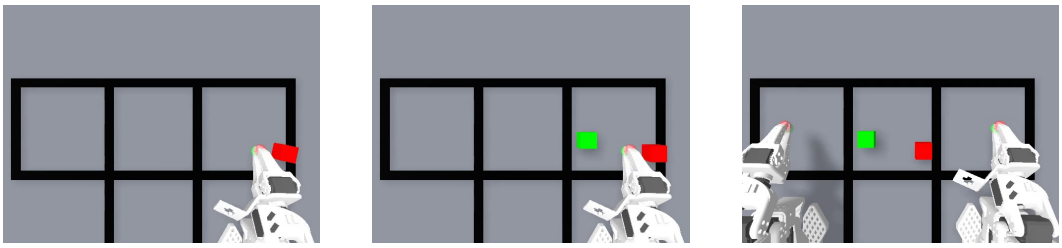

Figure 4: Reproduce Tasks in ManiSkill3

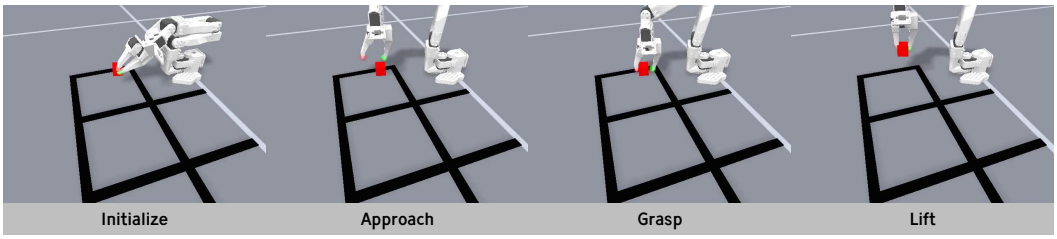

Figure 5: Visualizations of the motion planning demonstrations.

4.4 Simulation Reproduction and Real-World Deployment

As shown in Fig 4, we successfully reproduce benchmark tasks required by the project, including lifting, stacking, and sorting in ManiSkill3 (Tao et al., 2025). As shown in Fig 5, we also implemented motion planners for each task to collect high quality expert demonstrations efficiently. Each of these motion planner can achieve a success rate higher than 50%. For these tasks, we collect 50, 100, and 100 demos respectively, and use these demos to train ACT policy (Zhao et al., 2023), which provided by LeRobot (Cadene et al., 2024).

For the lifting and stacking, the robot's observations includes one external camera and a wrist-mounted camera on its right arm. For the dual arm task, a wrist-mounted camera on each of its two arms are included, making a total of three cameras. All cameras capture images at a resolution of 640×480, complemented by the joint pose.

The evaluation results are shown in table 2 with all three tasks get success rates higher than 50%. For real-world deployment, we directly train our policy with real-data provided by TA. All three tasks are successfully executed in the real-world setup provided by TA.

Table 2: Success Rates of Reproduce Tasks

Task	Lifting	Stacking	Sorting
Success Rate ↑	0.55	0.65	0.50

5 Conclusion and Discussion

In this project, we propose a task-relevant trajectory modeling framework for embodied manipulation that integrates semantic segmentation with trajectory-guided policy learning. We demonstrate that focusing on task-relevant trajectories improves performance and data efficiency over prior methods. Through experiments in simulation and real-world deployment, we finished all requirements of this project assignment.

However, we observe several limitations. First, points are randomly sampled within masks at each frame, which may introduce temporal inconsistency. Tracking fixed semantic keypoints, such as gripper tips or object handles, could further improve stability. Second, trajectories from different camera views are predicted independently, which may lead to inconsistencies in multi-view settings. A unified multi-view trajectory representation is a promising direction for future work.

6 Appendix

Our code is hosted in private GitHub repositories at the following addresses `https://github.com/yzc0731/eai_project` and `https://github.com/yzc0731/lerobot`. Both repositories remain private, and we have added the GitHub accounts of all TAs as collaborators to ensure access.

7 Acknowledgments

We would like to express our sincere gratitude to the teaching team for their invaluable guidance and continuous support throughout this project. We also thank Ruicheng Li (lirc23@mails.tsinghua.edu.cn) for his discussions.

References

Remi Cadene, Simon Alibert, Alexander Soare, Quentin Gallouedec, Adil Zouitine, Steven Palma, Pepijn Kooijmans, Michel Aractingi, Mustafa Shukor, Dana Aubakirova, Martino Russi, Francesco Capuano, Caroline Pascal, Jade Choghari, Jess Moss, and Thomas Wolf. Lerobot: State-of-the-art machine learning for real-world robotics in pytorch. `https://github.com/huggingface/lerobot`, 2024.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

Chongkai Gao, Haozhuo Zhang, Zhixuan Xu, Zhehao Cai, and Lin Shao. Flip: Flow-centric generative planning as general-purpose manipulation world model. *arXiv preprint arXiv:2412.08261*, 2024.

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. *arXiv preprint arXiv:2311.01977*, 2023.

Irmak Guzey, Yinlong Dai, Georgy Savva, Raunaq Bhirangi, and Lerrel Pinto. Bridging the human to robot dexterity gap through object-oriented rewards. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3344–3351. IEEE, 2025.

Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. *arXiv preprint arXiv:2410.11831*, 2024.

Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *International conference on machine learning*, pp. 5583–5594. PMLR, 2021.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded sam: Assembling open-world models for diverse visual tasks, 2024.

Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse kai Chan, Yuan Gao, Xuanlin Li, Tongzhou Mu, Nan Xiao, Arnav Gurha, Viswesh Nagaswamy Rajesh, Yong Woo Choi, Yen-Ru Chen, Zhiao Huang, Roberto

Calandra, Rui Chen, Shan Luo, and Hao Su. Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai. *Robotics: Science and Systems*, 2025.

Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. *arXiv preprint arXiv:2401.00025*, 2023.

Kelin Yu, Sheng Zhang, Harshit Soora, Furong Huang, Heng Huang, Pratap Tokekar, and Ruohan Gao. Genflowrl: Shaping rewards with generative object-centric flow in visual reinforcement learning. *arXiv preprint arXiv:2508.11049*, 2025.

Tony Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. 07 2023. doi: 10.15607/RSS.2023.XIX.016.

Yitian Zheng, Zhangchen Ye, Weijun Dong, Shengjie Wang, Yuyang Liu, Chongjie Zhang, Chuan Wen, and Yang Gao. Translating flow to policy via hindsight online imitation. *arXiv preprint arXiv:2512.19269*, 2025.