Dream2Flow: Bridging Video Generation and Open-World Manipulation with 3D Object Flow

Karthik Dharmarajan, Wenlong Huang, Jiajun Wu, Li Fei-Fei*, Ruohan Zhang*
Stanford University

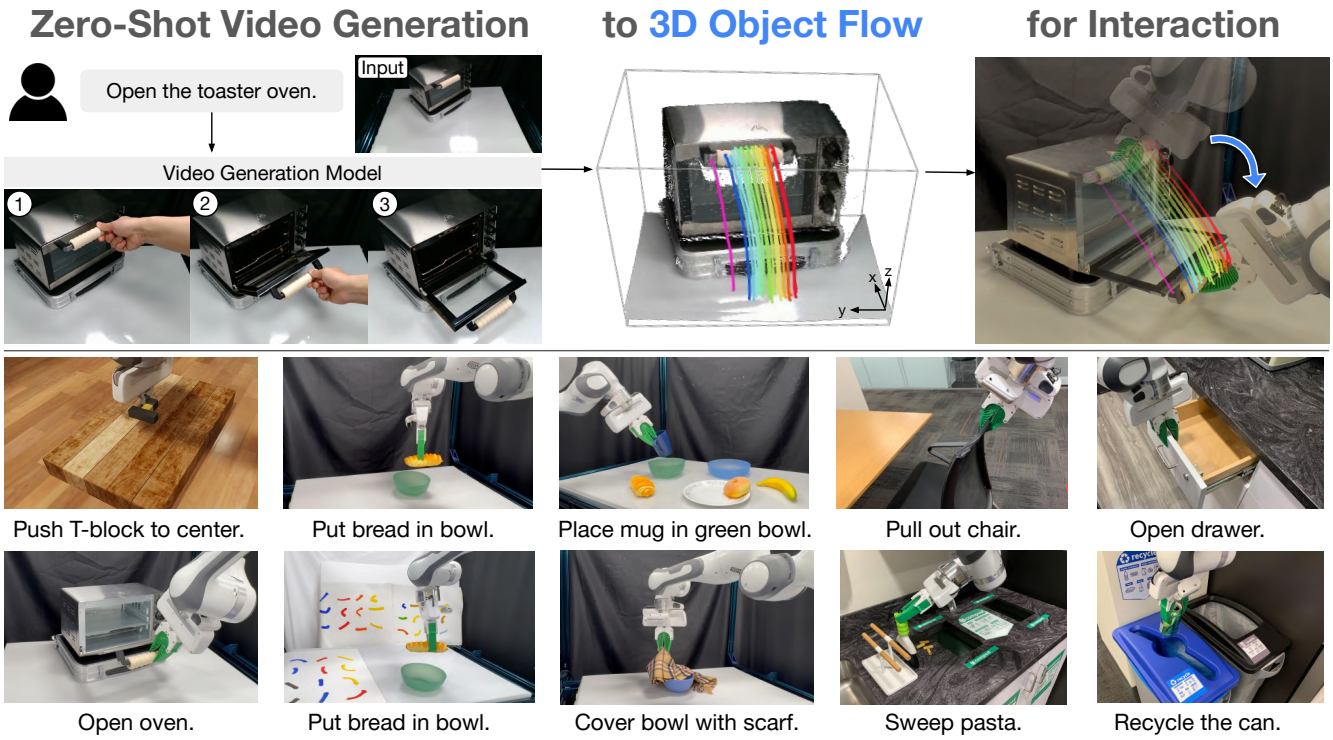

Fig. 1: **Dream2Flow** leverages off-the-shelf video generation models to produce videos of the task being performed in the same scene of the robot. Dream2Flow then extracts a 3D object flow from the motion in the video, allowing for downstream planning and execution with a robot across a wide variety of tasks.

***Abstract*— Generative video modeling has emerged as a compelling tool to zero-shot reason about plausible physical interactions for open-world manipulation. Yet, it remains a challenge to translate such human-led motions into the low-level actions demanded by robotic systems. We observe that given an initial image and task instruction, these models excel at synthesizing sensible object motions. Thus, we introduce Dream2Flow, a framework that bridges video generation and robotic control through 3D object flow as an intermediate representation. Our method reconstructs 3D object motions from generated videos and formulates manipulation as object trajectory tracking. By separating the state changes from the actuators that realize those changes, Dream2Flow overcomes the embodiment gap and enables zero-shot guidance from pre-trained video models to manipulate objects of diverse categories—including rigid, articulated, deformable, and granular. Through trajectory optimization or reinforcement learning, Dream2Flow converts reconstructed 3D object flow into executable low-level commands without task-specific demonstrations. Simulation and real-world experiments highlight 3D object flow as a general and scalable interface for adapting video generation models to open-world robotic manipulation. Videos, visualizations, and appendix are available at https://dream2flow.github.io/.**

I. Introduction

Robotic manipulation in the open world could benefit from visual world models that predict how an environment evolves under interaction. Recent generative video models can synthesize plausible physical interactions from an unseen image and an open-ended task instruction [1], offering rich priors for novel tasks in unseen environments. Despite their promise, it remains unclear what role such models should serve in a robot manipulation system. Most frontier video generators work best with human embodiments where supervision is more abundant, but this poses an embodiment gap for robot control.

We address this challenge by extracting actionable signals from visual predictions, which will then be enacted by a robot. Our method, **Dream2Flow**, uses **3D object flow** as an intermediate interface between video generation and robot control. Rather than mimic human motion, we reconstruct and track the task-relevant object motion in 3D. The problem

*Equal Advising. Correspondence: Wenlong Huang

becomes object trajectory tracking: the robot follows the generated object flow while respecting embodiment-specific constraints. This approach separates what should happen in the scene from how a robot executes it, and supports both motion planning and sensorimotor policies.

Using off-the-shelf models and tools, we demonstrate an autonomous pipeline that 1) generates a text-conditioned video of plausible interactions [1], 2) reconstructs 3D object flow via depth estimation and point tracking [2–4], and 3) synthesizes actions with trajectory optimization or reinforcement learning. Since 3D object flow captures task-relevant state changes, it enables manipulation of rigid, articulated, deformable, and granular objects without task-specific demonstrations.

In summary, our key contributions are:

- We propose 3D object flow as an interface for adapting off-the-shelf video generation models for open-world manipulation by formulating it as an object trajectory tracking problem.
- We demonstrate its effectiveness by implementing the approach in both simulated and real domains, which performs diverse tasks given only RGB-D observations and language instructions in a zero-shot manner.
- We examine the properties of 3D object flow by comparing it with alternative intermediate representations and by studying its key design choices as well as generalization properties.

II. Related Works

A. Task Specification in Manipulation

Manipulation tasks have been specified through symbolic goals and constraints [5–7]. Learning-based systems specify tasks often through language and perception, mapping instructions to actions via language-conditioned visuomotor policies and vision–language–action models [8–11]. Object-centric alternatives rely on descriptors or keypoints to capture task-relevant structure [12]. Recently, foundation models enable higher-level interfaces that compile intent into actionable specifications via code [13], 3D value maps akin to potential fields [14], keypoint relations [15], or affordance maps [16].

B. 2D/3D Flow in Robotics

Dense motion fields—optical flow, point tracks, and 3D scene/object flow—provide an embodiment-agnostic, mid-level interface for manipulation [17, 18]. In scene-centric formulations, policies parameterize or condition on motion in 2D or 3D to decide actions [19–26]. In object-centric formulations, desired object motion is specified independently of embodiment and then converted into actions via policy inference, planning, and optimization [27–33]. Our approach follows the this path by reconstructing 3D object flow from language-conditioned generations and tracking it under embodiment constraints, complementing other flow-conditioned policy representations [19, 21, 27, 29, 30].

C. Video Models for Robotics

Recent work increasingly integrates video models across robotic tasks as auxiliary training objectives [34–39], reward models [40–42], policies [43, 44], or as a simulator for the environments [45, 46]. Notably, predictive modeling in robotics can leverage video frame prediction as a form of dynamics model [44, 47–51]. Another notable direction is that video generation can directly provide new training data for robot learning by imagining new trajectories in the form of videos for imitation learning [52, 53].

III. Method

Herein, we introduce the problem formulation of Dream2Flow in Sec. III-A. Leveraging 3D object flow as an interface, we then describe how to extract it from video generations in Sec. III-B and how to infer actions from it in Sec. III-C.

A. Problem Formulation

Given a task instruction ℓ, an initial RGB-D observation (I_0, D_0), and camera projection Π (intrinsics and extrinsics to the robot frame), our goal is to output an action sequence $u_{0:H-1} \in \mathcal{U}^H$ that follows an object motion inferred from a generated video. We make no assumption about a specific action parameterization: $\mathcal{U}$ may denote motion primitives, end-effector poses, or low-level controls.

Extracting 3D Object Flow. From (I_0, ℓ), an image-to-video model produces frames $\{V_t\}_{t=1}^T$, and a video-depth estimator provides $\{Z_t\}_{t=1}^T$. Given a binary mask M of the task-relevant object, we lift masked image points with $Z_{1:T}$ and Π to obtain an object-centric 3D trajectory $P_{1:T} \in \mathbb{R}^{T\times n\times 3}$, which we call the 3D object flow.

Action Inference with 3D Object Flow. We represent the state as points on the task-relevant object and robot proprioception, $x_t = (x_t^{\text{obj}}, r_t)$. Let f be a dynamics model and $\hat{x}_{t+1} = f(\hat{x}_t, u_t)$ with $\hat{x}_0 = x_0$. At each planning step t, we use a time-aligned target $\tilde{P}_t \in \mathbb{R}^{n\times 3}$ derived from the video object flow (e.g., via uniform time-warping or nearest-shape matching). We solve:

$$\begin{aligned} \min_{\{u_t\in\mathcal{U}\}} \quad & \sum_{t=0}^{H-1} \lambda_{\text{task}}\big(\hat{x}_t^{\text{obj}}, \tilde{P}_t\big) + \lambda_{\text{control}}(\hat{x}_t, u_t) \\ \text{s.t.} \quad & \hat{x}_{t+1} = f(\hat{x}_t, u_t), \quad \hat{x}_0 = x_0, \\ & \lambda_{\text{task}}\big(\hat{x}_t^{\text{obj}}, \tilde{P}_t\big) = \sum_{i=1}^{n} \big\|\hat{x}_t^{\text{obj}}[i] - \tilde{P}_t[i]\big\|_2^2, \end{aligned}$$

Section III-C instantiates $\mathcal{U}$ and f for different domain.

B. Extracting 3D Object Flows from Videos

Video Generation: Given instruction ℓ and the initial RGB image I_0, Dream2Flow uses an off-the-shelf image-to-video model to generate a video $\{V_t\}_{t=1}^T$ of the task being performed. We exclude the robot from the initial frame and text prompt because current video models often produce less plausible fine-grained robot interactions, which in turn degrades the reconstructed trajectories.

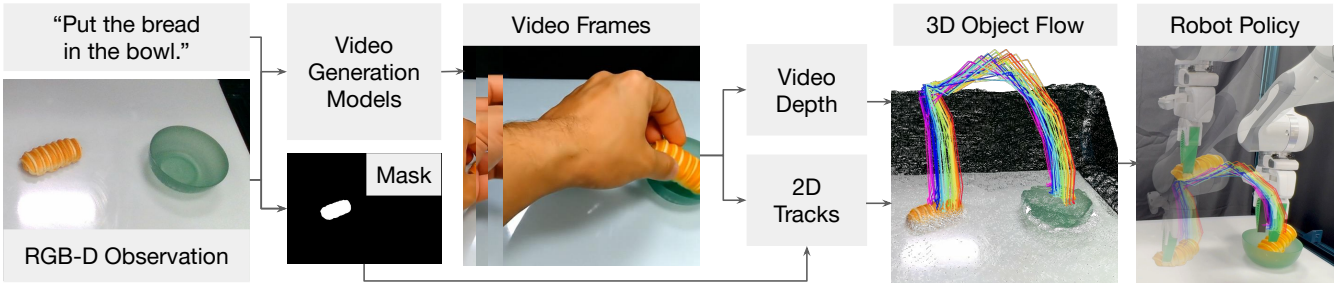

Fig. 2: **An overview of Dream2Flow.** Given a task instruction and an initial RGB-D observation, an image-to-video model synthesizes video frames conditioned on the instruction. We additionally obtain object masks, video depth, and point tracking from vision foundation models, which are used to reconstruct 3D object flow. Finally, a robot policy generates executable actions that track the 3D object flow using trajectory optimization or reinforcement learning.

Video Depth Estimation: We estimate per-frame depth $\{\tilde{Z}_t\}_{t=1}^{T}$ with SpatialTrackerV2 [4, 54]. To resolve monocular scale-shift ambiguity, we align the first frame to the robot depth D_0 and obtain calibrated depths $Z_t = s^\star \tilde{Z}_t + b^\star$.

3D Object Flow Extraction: 3D object flow aims to produce 3D trajectories $P_{1:T} \in \mathbb{R}^{T\times n\times 3}$ with visibilities $V \in \{0,1\}^{T\times n}$ for the task-relevant object. We first localize the task-relevant object using Grounding DINO [55] and SAM 2 [2]. From the mask at $t{=}1$, we sample n pixels and track them with CoTracker3 [3] to obtain 2D trajectories and visibilities. Visible points are then lifted to 3D using the calibrated depths and camera intrinsics/extrinsics, yielding $P_{1:T}$ in the robot frame.

C. Action Inference with 3D Object Flow

Simulated Push-T Domain. For Push-T, Dream2Flow uses a push primitive parameterized by start position, direction, and distance. We learn a particle-based forward dynamics model which takes in as input the feature-augmented scene points, consisting of positions, RGB colors, surface normals, and push parameters. We proceed to use random-shooting to sample r push skill parameters and then select the candidate with the lowest predicted flow-tracking cost. The target points for the cost are selected from the flow L timesteps ahead of the timestep with the closest points to the current observation $t^\star$.

Real-World Domain. We use absolute end-effector poses as the action space and a rigid-grasp dynamics model. We first proceed to grasp the desired part on the relevant object, and then use the dynamics model and point-flow following objective to move the end-effector such that the grasped part motion is similar to the video. Candidate grasps come from AnyGrasp [56], and we choose the one closest to the thumb detected in the generated video via HaMeR [57], which indicates the intended interaction point, such as a handle. The rigid-grasp dynamics model assumes that grasped points move with the end-effector while non-grasped points remain fixed, allowing us to produce an end-effector trajectory with PyRoki [58] using flow-tracking, smoothness, and reachability costs.

Simulated Door Opening Domain. For Door Opening, we use reinforcement learning to learn a sensimotor policy which moves the object according to the 3D object flow with SAC [59]. This approach can be viewed as using the simulator as a dynamics model for compiling the optimization process prescribed in Eq. III-A into a parametric policy in an offline manner. The reward function involves a contact term and 3D object flow progress term.

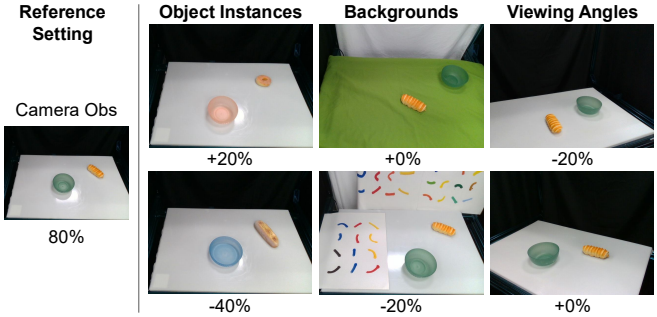

Fig. 3: **Robustness evaluations.** Relative performance across instance, background, and task variations, showing Dream2Flow remains robust under various different settings.

IV. Experiments

We seek to answer the following research questions through our experiments: $\mathcal{Q}\mathbf{1}$: What are properties of 3D object flow when used as an interface to bridge videos and robot control? $\mathcal{Q}\mathbf{2}$: How does Dream2Flow perform compared to alternative interfaces? $\mathcal{Q}\mathbf{3}$: How effective is 3D object flow as a reward for learning sensimotor policies?

We evaluate Dream2Flow on simulated and real manipulation tasks spanning rigid, articulated, deformable, and granular objects, including Push-T, Bread in Bowl, Oven Opening, Bowl Covering, and Door Opening as seen in Figures 1 and 4.

A. Properties of 3D Object Flow as a Video-Control Interface

For the simulated Push-T environment, we use Wan2.1 [60] with a goal image prompt and evaluate 10 initial states with 10 seeds each for 100 total trials. Six generated videos exhibited severe T-block morphing, which corrupted tracking and execution. In the real world, we use Veo 3 [61] and run 10 trials per task across the Bread in Bowl, Oven Opening, and Bowl Covering tasks as shown in Table I.

Task	AVDC	RIGVID	Dream2Flow
Push-T	-	-	**52/100**
Bread in Bowl	7/10	6/10	**8/10**
Open Oven	0/10	6/10	**8/10**
Cover Bowl	2/10	1/10	**3/10**

TABLE I: **Comparisons of intermediate representations on real robot.** Dream2Flow outperforms AVDC and RIGVID across three tasks by following 3D object flow rather than rigid transforms alone.

For evaluating certain axes of generalization, we run five trials each across variations in object instance, background, and viewpoint as in Fig. 3. Performance remains similar to the original setting except on the large-bread variation. We additionally observe that different language prompts on the same scene can lead to performing different tasks as shown on the website. We additionally show case studies demonstrating completion of in-the-wild tasks in Fig. 1 .

Across 60 real-world trials, failures mainly stem from video artifacts, tracking loss under occlusion or rotation, and failed executions (Fig. 5). Among video failures, common modes are object morphing and hallucination, while execution failures mostly occur in the Cover Bowl task involving a deformable object.

B. How does Dream2Flow perform compared to alternative interfaces?

For the three real-world tasks, we compare against AVDC [62], which uses dense optical flow and depth to estimate a sequence of rigid object transforms from the generated video as well as RIGVID [63] which tracks rigid object poses. For deformable settings, we adapt RIGVID by solving for rigid transforms between initial and visible 3D points.

Table I shows that Dream2Flow outperforms AVDC and RIGVID. AVDC can track bread reasonably well, but the optical flow could not capture the motion of the oven door. AVDC and RIGVID become brittle when visible points are sparse or occluded, making rigid transform estimation (and consequently execution) noisy. In contrast, Dream2Flow moves smoothly between points of high visibility since it does not perform rigid transform estimation.

C. How effective is 3D object flow as a reward for learning sensimotor policies?

We also use 3D object flow as an RL reward. SAC [59] policies trained with the hand-crafted object state reward and with the 3D object flow based reward achieve comparable success across a Franka Panda, a floating base Spot, and a GR1 over 100 random door positions (Table II). Figure 4 illustrates that different embodiments discover distinct yet effective strategies, with Spot using base motion for reachability and GR1 relying more on palm-finger contact for stability.

V. Conclusion

We presented Dream2Flow, which converts text-conditioned video generations into executable robot behavior by reconstructing and tracking 3D object flow. By separating task-relevant object motion from embodiment-specific control, the method supports planning and policy learning across simulated and real tasks involving rigid, articulated, deformable, and granular objects. Results show stronger performance than rigid-trajectory baselines and highlight current limits from video artifacts, tracking loss, and grasp selection. Overall, 3D object flow provides a practical bridge from open-ended video generation to robot control.

Reward Type	Franka	Spot	GR1
Object State	99/100	99/100	**96/100**
3D Object Flow	**100/100**	**100/100**	94/100

TABLE II: **Comparison of policies trained using different rewards.** The policies trained using the 3D object flow reward perform comparably to those trained with the object state reward across different embodiments.

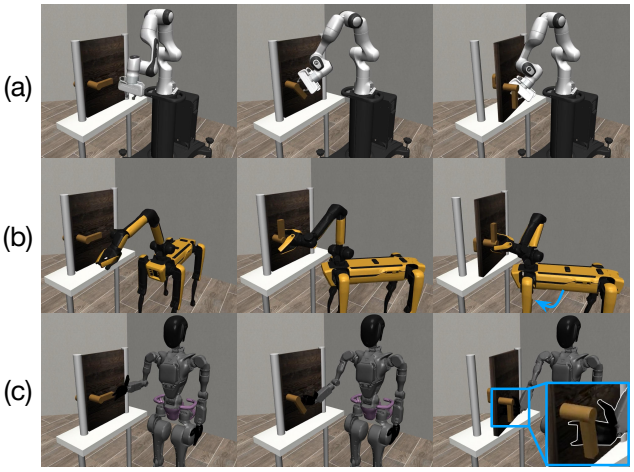

Fig. 4: **Rollouts from policies trained using 3D object flow as a reward.** Different embodiments such as the (a) Panda, (b) Spot, or (c) GR1 use different strategies to open the door. The Spot is able move its base for better reachability while the GR1 uses the area between its fingers and palm to pull for better stability.

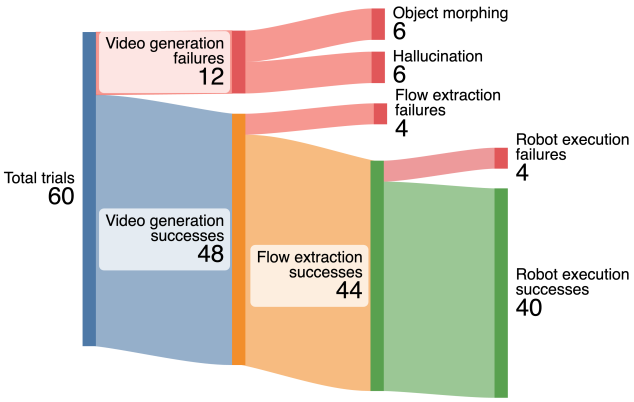

Fig. 5: **Failure breakdown on real-robot experiments.** Common causes include video artifacts (object morphing, object hallucination), tracking errors, and grasp selection mismatches.

REFERENCES

[1] T. Brooks, B. Peebles, C. Holmes, W. DePue, Y. Guo, L. Jing, D. Schnurr, J. Taylor, T. Luhman, E. Luhman *et al.*, "Video generation models as world simulators," *OpenAI Blog*, vol. 1, no. 8, p. 1, 2024.

[2] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer, "Sam 2: Segment anything in images and videos," *arXiv preprint arXiv:2408.00714*, 2024. [Online]. Available: https://arxiv.org/abs/2408.00714

[3] N. Karaev, I. Makarov, J. Wang, N. Neverova, A. Vedaldi, and C. Rupprecht, "CoTracker3: Simpler and better point tracking by pseudo-labelling real videos," *arxiv*, 2024.

[4] Y. Xiao, J. Wang, N. Xue, N. Karaev, I. Makarov, B. Kang, X. Zhu, H. Bao, Y. Shen, and X. Zhou, "Spatialtrackerv2: 3d point tracking made easy," in *ICCV*, 2025.

[5] C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez, "Integrated task and motion planning," *Annual review of control, robotics, and autonomous systems*, vol. 4, no. 1, pp. 265–293, 2021.

[6] Z. Zhao, S. Cheng, Y. Ding, Z. Zhou, S. Zhang, D. Xu, and Y. Zhao, "A survey of optimization-based task and motion planning: From classical to learning approaches," *IEEE/ASME Transactions on Mechatronics*, 2024.

[7] M. Toussaint, "Logic-geometric programming: An optimization-based approach to combined task and motion planning." in *IJCAI*, 2015, pp. 1930–1936.

[8] M. Shridhar, L. Manuelli, and D. Fox, "Cliport: What and where pathways for robotic manipulation," in *Conference on robot learning*. PMLR, 2022, pp. 894–906.

[9] ——, "Perceiver-actor: A multi-task transformer for robotic manipulation," in *Conference on Robot Learning*. PMLR, 2023, pp. 785–799.

[10] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in *Conference on Robot Learning*. PMLR, 2023, pp. 2165–2183.

[11] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. P. Foster, P. R. Sanketi, Q. Vuong *et al.*, "Openvla: An open-source vision-language-action model," in *Conference on Robot Learning*. PMLR, 2025, pp. 2679–2713.

[12] A. Simeonov, Y. Du, A. Tagliasacchi, J. B. Tenenbaum, A. Rodriguez, P. Agrawal, and V. Sitzmann, "Neural descriptor fields: Se (3)-equivariant object representations for manipulation," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6394–6400.

[13] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, "Code as policies: Language model programs for embodied control," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 9493–9500.

[14] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, "Voxposer: Composable 3d value maps for robotic manipulation with language models," *Proceedings of Machine Learning Research*, vol. 229, 2023.

[15] W. Huang, C. Wang, Y. Li, R. Zhang, and L. Fei-Fei, "Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation," in *Conference on Robot Learning*. PMLR, 2025, pp. 4573–4602.

[16] Y. Tang, W. Huang, Y. Wang, C. Li, R. Yuan, R. Zhang, J. Wu, and L. Fei-Fei, "Uad: Unsupervised affordance distillation for generalization in robotic manipulation," *arXiv preprint arXiv:2506.09284*, 2025.

[17] M. Xu, Z. Xu, Y. Xu, C. Chi, G. Wetzstein, M. Veloso, and S. Song, "Flow as the cross-domain manipulation interface," in *Conference on Robot Learning*. PMLR, 2025, pp. 2475–2499.

[18] C. Yuan, C. Wen, T. Zhang, and Y. Gao, "General flow as foundation affordance for scalable robot learning," in *Conference on Robot Learning*. PMLR, 2025, pp. 1541–1566.

[19] T. Weng, S. Bajracharya, Y. Wang, K. Agrawal, and D. Held, "Fabricflownet: Bimanual cloth manipulation with a flow-based policy," *arXiv preprint arXiv:2111.05623*, 2021.

[20] A. Goyal, A. Mousavian, C. Paxton, Y.-W. Chao, B. Okorn, J. Deng, and D. Fox, "Ifor: Iterative flow minimization for robotic object rearrangement," *arXiv preprint arXiv:2202.00732*, 2022.

[21] D. Seita, Y. Wang, S. J. Shetty, E. Y. Li, Z. Erickson, and D. Held, "Toolflownet: Robotic manipulation with tools via predicting tool flow from point clouds," *arXiv preprint arXiv:2211.09006*, 2022.

[22] T. Chen, Y. Mu, Z. Liang, Z. Chen, S. Peng, Q. Chen, M. Xu, R. Hu, H. Zhang, X. Li, and P. Luo, "G3flow: Generative 3d semantic flow for pose-aware and generalizable object manipulation," *arXiv preprint arXiv:2411.18369*, 2024.

[23] S. Wang, J. You, Y. Hu, J. Li, and Y. Gao, "Skil: Semantic keypoint imitation learning for generalizable data-efficient manipulation," in *Robotics: Science and Systems (RSS)*, 2025.

[24] S. Haldar and L. Pinto, "Point policy: Unifying observations and actions with key points for robot manipulation," *arXiv preprint arXiv:2502.20391*, 2025.

[25] S. Guo, X. Liang, J. Lin, Y. Zhuang, L. Lin, and X. Liang, "Actionsink: Toward precise robot manipulation with dynamic integration of action flow," *arXiv preprint arXiv:2508.03218*, 2025.

[26] Y. Yang, Z. Cai, Y. Tian, J. Zeng, and J. Pang, "Gripper keypose and object pointflow as interfaces for bimanual robotic manipulation," *arXiv preprint arXiv:2504.17784*, 2025.

[27] B. Eisner, H. Zhang, and D. Held, "Flowbot3d: Learning 3d articulation flow to manipulate articulated objects," in *Robotics: Science and Systems (RSS)*, 2022.

[28] H. Zhang, B. Eisner, and D. Held, "Flowbot++: Learning generalized articulated objects manipulation via articulation projection," 2024. [Online]. Available: https://arxiv.org/abs/2306.12893

[29] C. Gao, H. Zhang, Z. Xu, Z. Cai, and L. Shao, "Flip: Flow-centric generative planning as general-purpose manipulation world model," *arXiv preprint arXiv:2412.08261*, 2024.

[30] J. Guo, X. Ma, Y. Wang, M. Yang, H. Liu, and Q. Li, "Flowdreamer: A rgb-d world model with flow-based motion representations for robot manipulation," *arXiv preprint arXiv:2505.10075*, 2025.

[31] H. Zhi, P. Chen, S. Zhou, Y. Dong, Q. Wu, L. Han, and M. Tan, "3dflowaction: Learning cross-embodiment manipulation from 3d flow world model," *arXiv preprint arXiv:2506.06199*, 2025.

[32] Y. He and Q. Nie, "Manitrend: Bridging future generation and action prediction with 3d flow for robotic manipulation," *arXiv preprint arXiv:2502.10028*, 2025.

[33] Z.-H. Yin, S. Yang, and P. Abbeel, "Object-centric 3d motion field for robot learning from human videos," *arXiv preprint arXiv:2506.04227*, 2025.

[34] H. Wu, Y. Jing, C. Cheang, G. Chen, J. Xu, X. Li, M. Liu, H. Li, and T. Kong, "Unleashing large-scale video generative pre-training for visual robot manipulation," *arXiv preprint arXiv:2312.13139*, 2023.

[35] Y. Seo, K. Lee, S. L. James, and P. Abbeel, "Reinforcement learning with action-free pre-training from videos," in *International Conference on Machine Learning*. PMLR, 2022, pp. 19 561–19 579.

[36] J. Wu, H. Ma, C. Deng, and M. Long, "Pre-training contextualized world models with in-the-wild videos for reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 36, pp. 39 719–39 743, 2023.

[37] J. Yang, B. Liu, J. Fu, B. Pan, G. Wu, and L. Wang, "Spatiotemporal predictive pre-training for robotic motor control," *arXiv preprint arXiv:2403.05304*, 2024.

[38] Y. Hu, Y. Guo, P. Wang, X. Chen, Y.-J. Wang, J. Zhang, K. Sreenath, C. Lu, and J. Chen, "Video prediction policy: A generalist robot policy with predictive visual representations," *arXiv preprint arXiv:2412.14803*, 2024.

[39] S. Li, Y. Gao, D. Sadigh, and S. Song, "Unified video action model," *arXiv preprint arXiv:2503.00200*, 2025.

[40] T. Huang, G. Jiang, Y. Ze, and H. Xu, "Diffusion reward: Learning rewards via conditional video diffusion," in *European Conference on Computer Vision*. Springer, 2024, pp. 478–495.

[41] A. Escontrela, A. Adeniji, W. Yan, A. Jain, X. B. Peng, K. Goldberg, Y. Lee, D. Hafner, and P. Abbeel, "Video prediction models as rewards for reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 36, pp. 68 760–68 783, 2023.

[42] A. S. Chen, S. Nair, and C. Finn, "Learning generalizable robotic reward functions from" in-the-wild" human videos," *arXiv preprint arXiv:2103.16817*, 2021.

[43] A. Ajay, S. Han, Y. Du, S. Li, A. Gupta, T. Jaakkola, J. Tenenbaum, L. Kaelbling, A. Srivastava, and P. Agrawal, "Compositional foundation models for hierarchical planning," *Advances in Neural Information Processing Systems*, vol. 36, pp. 22 304–22 325, 2023.

[44] Y. Du, S. Yang, B. Dai, H. Dai, O. Nachum, J. Tenenbaum, D. Schuurmans, and P. Abbeel, "Learning universal policies via text-guided video generation," *Advances in neural information processing systems*, vol. 36, pp. 9156–9172, 2023.

[45] D. Valevski, Y. Leviathan, M. Arar, and S. Fruchter, "Diffusion models are real-time game engines," *arXiv preprint arXiv:2408.14837*, 2024.

[46] J. Bruce, M. D. Dennis, A. Edwards, J. Parker-Holder, Y. Shi, E. Hughes, M. Lai, A. Mavalankar, R. Steigerwald, C. Apps *et al.*, "Genie: Generative interactive environments," in *Forty-first International Conference on Machine Learning*, 2024.

[47] M. Yang, Y. Du, K. Ghasemipour, J. Tompson, D. Schuurmans, and P. Abbeel, "Learning interactive real-world simulators," *arXiv preprint arXiv:2310.06114*, vol. 1, no. 2, p. 6, 2023.

[48] S. Zhou, Y. Du, J. Chen, Y. Li, D.-Y. Yeung, and C. Gan, "Robodreamer: Learning compositional world models for robot imagination," *arXiv preprint arXiv:2404.12377*, 2024.

[49] O. Rybkin, K. Pertsch, K. G. Derpanis, K. Daniilidis, and A. Jaegle, "Learning what you can do before doing anything," *arXiv preprint arXiv:1806.09655*, 2018.

[50] R. Mendonca, S. Bahl, and D. Pathak, "Structured world models from human videos," *arXiv preprint arXiv:2308.10901*, 2023.

[51] H. Che, X. He, Q. Liu, C. Jin, and H. Chen, "Gamegen-x: Interactive open-world game video generation," *arXiv preprint arXiv:2411.00769*, 2024.

[52] J. Jang, S. Ye, Z. Lin, J. Xiang, J. Bjorck, Y. Fang, F. Hu, S. Huang, K. Kundalia, Y.-C. Lin *et al.*, "Dreamgen: Unlocking generalization in robot learning through neural trajectories," *arXiv e-prints*, pp. arXiv–2505, 2025.

[53] J. Liang, R. Liu, E. Ozguroglu, S. Sudhakar, A. Dave, P. Tokmakov, S. Song, and C. Vondrick, "Dreamitate: Real-world visuomotor policy learning via video generation," *arXiv preprint arXiv:2406.16862*, 2024.

[54] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao, "Depth anything v2," *Advances in Neural Information Processing Systems*, vol. 37, pp. 21 875–21 911, 2024.

[55] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," *arXiv preprint arXiv:2303.05499*, 2023.

[56] H.-S. Fang, C. Wang, H. Fang, M. Gou, J. Liu, H. Yan, W. Liu, Y. Xie, and C. Lu, "Anygrasp: Robust and efficient grasp perception in spatial and temporal domains," *IEEE Transactions on Robotics (T-RO)*, 2023.

[57] G. Pavlakos, D. Shan, I. Radosavovic, A. Kanazawa, D. Fouhey, and J. Malik, "Reconstructing hands in 3D with transformers," in *CVPR*, 2024.

[58] C. M. Kim*, B. Yi*, H. Choi, Y. Ma, K. Goldberg, and A. Kanazawa, "Pyroki: A modular toolkit for robot kinematic optimization," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025.

[59] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 80. PMLR, 10–15 Jul 2018, pp. 1861–1870.

[60] T. Wan, A. Wang, B. Ai, B. Wen, C. Mao, and et al., "Wan: Open and advanced large-scale video generative models," *arXiv preprint arXiv:2503.20314*, 2025.

[61] Google DeepMind, "Veo-3 Technical Report," 2025. [Online]. Available: https://storage.googleapis.com/deepmind-media/veo/Veo-3-Tech-Report.pdf

[62] P.-C. Ko, J. Mao, Y. Du, S.-H. Sun, and J. B. Tenenbaum, "Learning to act from actionless videos through dense correspondences," *arXiv preprint arXiv:2310.08576*, 2023.

[63] S. Patel, S. Mohan, H. Mai, U. Jain, S. Lazebnik, and Y. Li, "Robotic manipulation by imitating generated videos without physical demonstrations," *arXiv preprint arXiv:2507.00990*, 2025.

[64] C. Li, R. Zhang, J. Wong, C. Gokmen, S. Srivastava, and et al., "Behavior-1k: A human-centered, embodied ai benchmark with 1,000 everyday activities and realistic simulation," *arXiv preprint arXiv:2403.09227*, 2024.

[65] Y. Zhu, J. Wong, A. Mandlekar, R. Martín-Martín, A. Joshi, K. Lin, S. Nasiriany, and Y. Zhu, "robosuite: A modular simulation framework and benchmark for robot learning," in *arXiv preprint arXiv:2009.12293*, 2020.

[66] X. Wu, L. Jiang, P.-S. Wang, Z. Liu, X. Liu, Y. Qiao, W. Ouyang, T. He, and H. Zhao, "Point transformer v3: Simpler, faster, stronger," in *CVPR*, 2024.

[67] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," http://pybullet.org, 2016–2021.

[68] Y. Zhu, A. Joshi, P. Stone, and Y. Zhu, "Viola: Imitation learning for vision-based manipulation with object proposal priors," *arXiv preprint arXiv:2210.11339*, 2022.