Bi3D Diffuser Actor: 3D Policy Diffusion for Bi-manual Robot Manipulation

Anonymous Author(s)
Affiliation
Address
email

Abstract: We present a conceptually simple and general framework for bi-manual manipulation that extends the state-of-the-art 3D diffusion policy 3D Diffuser Actor, by redefining the robot action in a bi-manual form. The method, called Bi3D Diffuser Actor, uses 3D scene feature representations aggregated from posed camera views and sensed depth, conditions on language instructions, and generates 3D trajectories of the left and right robot end effectors jointly. While most baselines struggle with the complexity of two-hand dynamics, our approach not only effectively manages action multimodality but also generates coordinated and synergistic two-hand motions, even in more challenging scenarios. Bi3D Diffuser Actor, trained in a multi-task setting, establishes a new state-of-the-art on PerAct2, with an absolute performance gain of 42.5% over prior approaches that are trained in single-task settings. We hope our simple yet effective approach will serve as a strong baseline and facilitate further research in bi-manual and dexterous manipulation.

Keywords: Diffusion model, 3D policy, Bi-manual manipulation, Imitation learning

1 Introduction

Bi-manual manipulation can unlock more potential for robots to solve more tasks and more effectively, essentially closing the gap between human and robot manipulation capabilities. However, the bi-manual setup is more challenging compared to single-arm manipulation. The two-hand dynamics introduce higher complexity, requiring the motion of both arms to be coordinated synergistically and precisely to achieve successful manipulation tasks. Past approaches [1, 2, 3, 4, 5] struggle to generalize to many tasks due to either less expressive architectures or limited training domains.

At the same time, recent works on single-arm manipulation have achieved remarkable success in handling action multimodality [6, 7, 8], effectively modeling the 3D structure of the scene [9, 10, 11] and incorporating representations from foundation models [12, 13]. These advancements have not been combined with bi-manual manipulation policies yet.

In this work, we aim to leverage the successful learning paradigms for single-arm manipulation into a bi-manual manipulation policy. We propose Bi3D Diffuser Actor, a novel 3D denoising policy transformer that builds upon the state-of-the-art 3D Diffuser Actor [13]. Similar to its predecessor, Bi3D Diffuser Actor takes as input a tokenized 3D scene representation, a language instruction and two noised end-effector's future translation and rotation trajectories, one for each arm; it predicts the error in translations and rotations for each arm's end-effector simultaneously.

We test Bi3D Diffuser Actor in learning policies from demonstrations on the simulation benchmark of PerAct2 [4]. Bi3D Diffuser Actor sets a new state-of-the-art with a 42.5% absolute gain, outperforming existing 3D and 2D policies.

Submitted to the CoRL 2024 Workshop on Whole-Body Control and Bimanual Manipulation. Do not distribute.

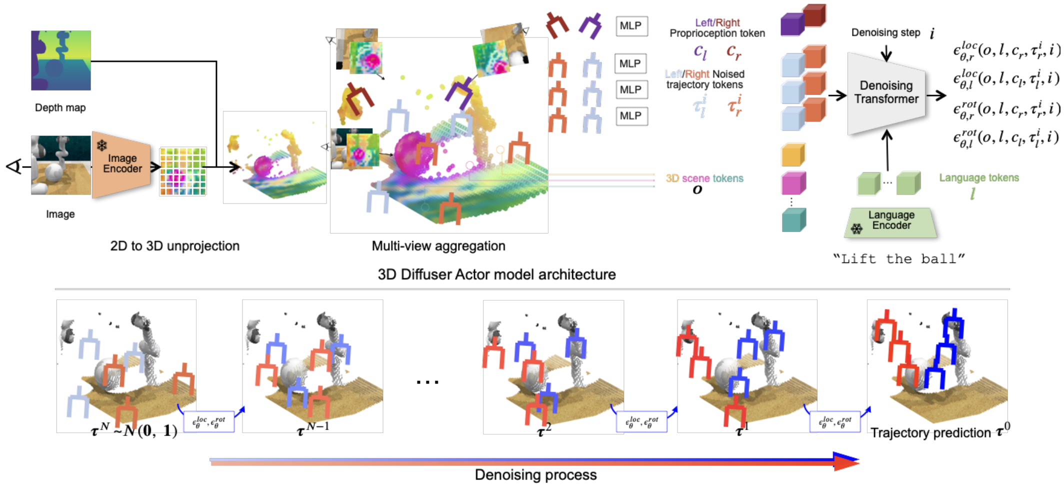

Figure 1: **Overview of Bi3D Diffuser Actor**. **Top:** Bi3D Diffuser Actor is a conditional diffusion model that generates 3D trajectories of two end-effectors. Similar to [13], at each diffusion step i, our model tokenizes the noised estimate of the robot's future end-effector trajectories, posed RGB-D views $\mathbf{o}$, and proprioceptive information c. These tokens are contextualized through attention, using 3D relative positional information, and attend to language tokens l to fuse the instructional information. Our model predicts the noise of left- and right-hand 3D locations $(\epsilon_{\theta,l}^{loc}(\mathbf{o}, l, c_l, \boldsymbol{\tau}_l^i, i)$ and $\epsilon_{\theta,r}^{loc}(\mathbf{o}, l, c_r, \boldsymbol{\tau}_r^i, i))$ and the noise of left- and right-hand 3D rotations $(\epsilon_{\theta,l}^{rot}(\mathbf{o}, l, c_l, \boldsymbol{\tau}_l^i, i)$ and $\epsilon_{\theta,r}^{rot}(\mathbf{o}, l, c_r, \boldsymbol{\tau}^i,_r i)$.). **Bottom:** During inference, Bi3D Diffuser Actor iteratively denoises the estimate of the future bi-manual trajectory.

2 Related Work

Bi-manual manipulation The difficulty of collecting bi-manual data has limited the scope of past works [1, 3]. Recently, [2, 14] propose cost-effective methods to scale data collection in the real world, yet the proposed architectures only absorb RGB observations and do not generalize to multiple tasks or variations. PerAct2 [4] introduces both a new multi-task simulator benchmark and a 3D model based on the Perceiver architecture [15]. VoxAct-B [5] further improves upon this formulation by tasking foundation models to detect the pose of the object of interest. In our work, we address the multimodality of action prediction, an underexplored question for bi-manual manipulation.

Diffusion models in robotics Diffusion models have been recently used as expressive policy representations in imitation learning [8, 7], as well as to model cross-object and object-part arrangements [16, 17, 18, 19, 20], visual image subgoals [21, 22, 23, 24], and in offline reinforcement learning [25, 26, 27]. Most related to our approach is 3D Diffuser Actor [13], a policy scheme that marries 3D scene representations and diffusion models. We show that by generalizing the notation of robot action into the form of bi-manual manipulation and tokenizing the robot action of two arms, we can easily extend 3D Diffuser Actor to tackle bi-manual manipulation.

3 Method

Bi3D Diffuser Actor builds upon the state-of-the-art 3D diffusion policy 3D Diffuser Actor [13], which is trained to generate the robot's end-effector trajectories for single-arm manipulation. We first summarize 3D Diffuser Actor and then describe our extension to bi-manual manipulation.

3.1 3D Diffuser Actor

3D Diffuser Actor is trained to imitate demonstration trajectories of the form of $\{(\mathbf{o}_1, \mathbf{a}_1), (\mathbf{o}_2, \mathbf{a}_2), ...\}$, accompanied with a task language instruction l, where $\mathbf{o}_t$ stands for the visual observation and $\mathbf{a}_t$ stands for robot action at timestep t. Each observation $\mathbf{o}_t$ is one or more posed RGB-D images. Each action $\mathbf{a}_t$ is a single-arm end-effector pose and is decomposed into 3D location, rotation and binary (open/close) state: $\mathbf{a}_t = \{\mathbf{a}_t^{\text{loc}} \in \mathbb{R}^3, \mathbf{a}_t^{\text{rot}} \in \mathbb{R}^6, \mathbf{a}_t^{\text{open}} \in \{0,1\}\}$. Let $\boldsymbol{\tau}_t = (\mathbf{a}_{t:t+T}^{\text{loc}}, \mathbf{a}_{t:t+T}^{\text{rot}})$ denote the trajectory of 3D locations and rotations at timestep t, of temporal horizon T. 3D Diffuser Actor, at each timestep t predicts a trajectory $\boldsymbol{\tau}_t$ and binary states $\mathbf{a}_{t:t+T}^{\text{open}}$.

3D Diffuser Actor is a conditional diffusion probabilistic model [28, 29] of trajectories given the visual scene and a language instruction; it predicts a whole trajectory $\boldsymbol{\tau}$ at once, non autoregressively, through iterative denoising, by inverting a process that gradually adds noise to a sample $\boldsymbol{\tau}^0$. 3D Diffuser Actor models a learned gradient of the denoising process with a 3D relative transformer $\hat{\epsilon} = \epsilon_\theta(\boldsymbol{\tau}_t^i; i, \mathbf{o}_t, l, c_t)$ that takes as input the noisy trajectory $\boldsymbol{\tau}_t^i$ at timestep t, diffusion step i, and conditioning information from the language instruction l, the visual observation $\mathbf{o}_t$ and proprioception c_t of timestep t, to predict the noise component $\hat{\epsilon}$. At each timestep t and diffusion step i, the visual observations $\mathbf{o}_t$, proprioception c_t and noised trajectory estimate $\boldsymbol{\tau}_t^i$ are converted to a set of 3D tokens. Each 3D token is represented by a latent embedding and a 3D position.

The model fuses all 3D tokens using a 3D Relative Denoising Transformer. This applies relative self-attentions among all 3D tokens and cross-attentions to the language tokens. The final trajectory tokens are fed to MLPs to predict: (1) the noise $\epsilon_\theta^{loc}(\mathbf{o}, l, c, \boldsymbol{\tau}^i, i)$ and $\epsilon_\theta^{rot}(\mathbf{o}, l, c, \boldsymbol{\tau}^i, i)$ added to $\boldsymbol{\tau}^0$'s sequence of 3D translations and 3D rotations, respectively, and (2) the end-effector opening $f_\theta^{\text{open}}(\mathbf{o}, l, c, \boldsymbol{\tau}^i, i) \in [0,1]^T$.

3.2 Bi3D Diffuser Actor

To extend 3D Diffuser Actor to bi-manual manipulation, we first redefine the robot action in a bi-manual form: $\mathbf{a}_{t,l}$ and $\mathbf{a}_{t,r}$ denote the robot action at timestep t, of the left and right robot arm respectively. Our goal is to predict the corresponding trajectory $\boldsymbol{\tau}_{t,l} = (\mathbf{a}_{t:t+T,l}^{\text{loc}}, \mathbf{a}_{t:t+T,l}^{\text{rot}})$ and $\boldsymbol{\tau}_{t,r} = (\mathbf{a}_{t:t+T,r}^{\text{loc}}, \mathbf{a}_{t:t+T,r}^{\text{rot}})$ of temporal horizon T for both arms.

We follow the same 3D tokenization procedure to map (1) the noisy estimate of pose $\mathbf{a}_l^i$ of $\boldsymbol{\tau}_l^i$ and $\mathbf{a}_r^i$ of $\boldsymbol{\tau}_r^i$ at diffusion step i, and (2) the left- and right-hand proprioceptive information c_l and c_r, into 3D tokens. We use the same 3D Relative Denoising Transformer architecture to contextualize these tokens and predict the translation and rotation noise as well as the end-effector opening for both arms.

Training and inference During training, we randomly sample a time step t and a diffusion step i and add noise $(\epsilon_l^{\text{loc}}, \epsilon_r^{\text{loc}}, \epsilon_l^{\text{rot}}, \epsilon_r^{\text{loc}})$ to a ground-truth left- and right-hand trajectory $(\boldsymbol{\tau}_{t,l}^0, \boldsymbol{\tau}_{t,r}^0)$. We use the $L1$ loss for reconstructing the sequence of 3D locations and 3D rotations. We use binary cross-entropy (BCE) loss to supervise the end-effector opening, we use the prediction from i=1 at inference time. Let $\epsilon_{\theta,l}^{loc}(\mathbf{o}, l, c_l, \boldsymbol{\tau}_l^i, i)$ and $\epsilon_{\theta,r}^{loc}(\mathbf{o}, l, c_r, \boldsymbol{\tau}_r^i, i)$ be the predicted noise of 3D translation, $\epsilon_{\theta,l}^{rot}(\mathbf{o}, l, c_l, \boldsymbol{\tau}_l^i, i)$ and $\epsilon_{\theta,r}^{rot}(\mathbf{o}, l, c_r, \boldsymbol{\tau}_r^i, i)$ be the predicted noise of 3D rotation, and $f_{\theta,l}^{\text{open}}(\mathbf{o}, l, c_l, \boldsymbol{\tau}_l^i, i)$ and $f_{\theta,r}^{\text{open}}(\mathbf{o}, l, c_r, \boldsymbol{\tau}_r^i, i)$ be the end-effector opening of the left and the right robot arm. Our objective reads:

$$
\begin{aligned}
\mathcal{L}_\theta = {} & w_1 [\|(\epsilon_{\theta,l}^{\text{loc}}(\mathbf{o}, l, c_l, \boldsymbol{\tau}_l^i, i) - \epsilon_l^{\text{loc}}\| + \|(\epsilon_{\theta,r}^{\text{loc}}(\mathbf{o}, r, c_r, \boldsymbol{\tau}_r^i, i) - \epsilon_l^{\text{loc}}\|] && (1) \\
& + w_2 [\|(\epsilon_{\theta,l}\text{rot}(\mathbf{o}, l, c_l, \boldsymbol{\tau}_l^i, i) - \epsilon_l^{\text{rot}}\| + \|(\epsilon_{\theta,r}\text{rot}(\mathbf{o}, l, c_r, \boldsymbol{\tau}_r^i, i) - \epsilon_r^{\text{rot}}\|] && (2) \\
& + [\text{BCE}(f_{\theta,l}^{\text{open}}(\mathbf{o}, l, c_l, \boldsymbol{\tau}_l^i, i), \mathbf{a}_{1:T,l}^{\text{open}}) + \text{BCE}(f_{\theta,r}^{\text{open}}(\mathbf{o}, l, c_r, \boldsymbol{\tau}_r^i, i), \mathbf{a}_{1:T,r}^{\text{open}})],
\end{aligned}
$$

where w_1, w_2 are hyperparameters estimated using cross-validation. To draw a sample from the learned distribution $p_\theta(\boldsymbol{\tau}_l, \boldsymbol{\tau}_r | \mathbf{o}, l, c)$, we start by drawing a sample of bi-manual trajectories $\boldsymbol{\tau}_l^N \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$ and $\boldsymbol{\tau}_r^N \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$. Then, we iteratively denoise the sample using the predicted noise according to a specified sampling schedule [30, 31].

Implementation details Following PerAct2 [4], we segment demonstrations and train our model to predict end-effector *keyposes*. During inference, we predict the next keypose and use a motion planner to reach it [9, 32, 10]. We use the same model architecture as 3D Diffuser Actor, except that our model has two sets of 3D trajectory tokens, one for each arm. We closely follow the hyper-parameters of 3D Diffuser Actor except that we train our model for $200,000$ iterations and use 5 camera views. Please check Table 7 in the paper of 3D Diffuser Actor for more details in the hyper-parameters.

	multi-task training	Avg. Success	push box	lift ball	push buttons	pick up plate	put item into drawer	put bottle into fridge
ACT	✗	5.9	0	36	4	0	13	0
RVT-LF	✗	10.5	52	17	39	3	10	0
PerAct-LF	✗	17.5	57	40	10	2	27	0
PerAct^2	✗	16.8	6	50	47	4	10	3
Bi3DDA (ours)	✓	**59.3**	**74**	**92**	**96**	**66**	**32**	**79**

	multi-task training	handover item	pick up laptop	straighten rope	sweep dust	lift tray	handover item (easy)	take tray out of oven
ACT	✗	0	0	16	0	6	0	2
RVT-LF	✗	0	3	3	0	6	0	3
PerAct-LF	✗	0	11	21	28	14	9	8
PerAct^2	✗	11	12	24	0	1	**41**	9
Bi3DDA (ours)	✓	**19**	**71**	**50**	**98**	**59**	20	**15**

Table 1: **Evaluation on PerAct2.** Our model is trained under a multi-task setting, while all other baselines are trained under single-task settings. Unlike baselines that report the best checkpoint on separate tasks, we only evaluate the final checkpoint across all tasks. **Bi3D Diffuser Actor outperforms all prior arts on most tasks by a large margin under a more challenging setup.**

4 Experiments

We evaluate Bi3D Diffuser Actor on PerAct2 [4], a recently-introduced learning-from-demonstrations benchmark for multi-task bi-manual manipulation. PerAct2 is based on RLBench [33] and uses two Franka Panda Robots to manipulate the scene. It has a suite of 13 bimanual tasks, each of which has 1-5 variations that concern the variability across object poses, appearance and semantics.

We follow PerAct2's experimental setup and use 100 demonstrations per task for model training and 100 episodes for evaluation. We use the same set of five RGB-D cameras, including the front, left/right wrist and left/right shoulder cameras. The input image resolution of 256×256. Similar to [4, 9], we extract keyposes from demonstrations and employ the low-level motion planner BiRRT [34] to reach the next keypose. We also note two major differences from the setup in [4]:

1. **We train our model under a multi-task setting, while [4] trains baselines under single-task settings.** Multi-task learning is essential towards building a robot generalist [35, 36].
2. **We test the final checkpoint on all tasks, instead of evaluating the best checkpoint for each task.** PerAct2 [4] saves intermediate checkpoints during training and selects the best one for each task, which is impractical when the number of tasks grows. We instead consistently use the final checkpoint for evaluation across all tasks.

We compare our model to the following baselines: i) ACT [2], a 2D transformer architecture that is trained as a conditional VAE to predict a sequence of actions; ii) RVT-LF [11, 4], that unprojects 2D views to form a point cloud, renders virtual views and feeds them to a transformer to predict the 3D actions for each arm in sequence; iii) PerAct-LF [9, 4], that vozelizes the 3D space and uses to a Perceiver [15] architecture to predict the 3D actions for each arm in sequence; iv) PerAct^2 [4], which shares the same architecture as PerAct-LF but predicts the actions for the two arms jointly.

Results We show quantitative results in Table 1. Bi3D Diffuser Actor achieves an average 59.3% success rate among all tasks, an absolute improvement of 42.5% over PerAct^2, even solving tasks that previous approaches are unable to solve, such *put bottle into fridge*.

5 Conclusion

We present Bi3D Diffuser Actor, a policy that extends 3D Diffuser Actor to bi-manual manipulation. Our method sets a new state-of-the-art on PerAct2 by a large margin, using a more challenging setup compared to all other baselines. Our future work includes to further extend the method to tackle bi-manual multi-fingered manipulation tasks.

References

[1] J. Grannen, Y. Wu, S. Belkhale, and D. Sadigh. Learning bimanual scooping policies for food acquisition. In *Conference on Robot Learning*, 2022.

[2] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *RSS*, 2023.

[3] J. Grannen, Y. Wu, B. Vu, and D. Sadigh. Stabilize to act: Learning to coordinate for bimanual manipulation. *CoRL*, 2023.

[4] M. Grotz, M. Shridhar, T. Asfour, and D. Fox. Peract2: A perceiver actor framework for bimanual manipulation tasks. *arXiv preprint arXiv:2407.00278*, 2024.

[5] I.-C. A. Liu, S. He, D. Seita, and G. Sukhatme. Voxact-b: Voxel-based acting and stabilizing policy for bimanual manipulation. *CoRL*, 2024.

[6] T. Pearce, T. Rashid, A. Kanervisto, D. Bignell, M. Sun, R. Georgescu, S. V. Macua, S. Z. Tan, I. Momennejad, K. Hofmann, and S. Devlin. Imitating human behaviour with diffusion models, 2023.

[7] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

[8] M. Reuss, M. Li, X. Jia, and R. Lioutikov. Goal-conditioned imitation learning using score-based diffusion policies. *arXiv preprint arXiv:2304.02532*, 2023.

[9] M. Shridhar, L. Manuelli, and D. Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2023.

[10] T. Gervet, Z. Xian, N. Gkanatsios, and K. Fragkiadaki. Act3d: 3d feature field transformers for multi-task robotic manipulation. *CoRL*, 2023.

[11] A. Goyal, J. Xu, Y. Guo, V. Blukis, Y.-W. Chao, and D. Fox. Rvt: Robotic view transformer for 3d object manipulation. *arXiv preprint arXiv:2306.14896*, 2023.

[12] M. Shridhar, L. Manuelli, and D. Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on Robot Learning*, pages 894–906. PMLR, 2022.

[13] T.-W. Ke, N. Gkanatsios, and K. Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. *arXiv preprint arXiv:2402.10885*, 2024.

[14] Z. Fu, T. Z. Zhao, and C. Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation, 2024.

[15] A. Jaegle, F. Gimeno, A. Brock, A. Zisserman, O. Vinyals, and J. Carreira. Perceiver: General perception with iterative attention, 2021.

[16] W. Liu, T. Hermans, S. Chernova, and C. Paxton. Structdiffusion: Object-centric diffusion for semantic rearrangement of novel objects. *arXiv preprint arXiv:2211.04604*, 2022.

[17] A. Simeonov, A. Goyal, L. Manuelli, L. Yen-Chen, A. Sarmiento, A. Rodriguez, P. Agrawal, and D. Fox. Shelving, stacking, hanging: Relational pose diffusion for multi-modal rearrangement. *arXiv preprint arXiv:2307.04751*, 2023.

[18] U. A. Mishra and Y. Chen. Reorientdiff: Diffusion model based reorientation for object manipulation. *arXiv preprint arXiv:2303.12700*, 2023.

[19] X. Fang, C. R. Garrett, C. Eppner, T. Lozano-Pérez, L. P. Kaelbling, and D. Fox. Dimsam: Diffusion models as samplers for task and motion planning under partial observability. *arXiv preprint arXiv:2306.13196*, 2023.

[20] N. Gkanatsios, A. Jain, Z. Xian, Y. Zhang, C. Atkeson, and K. Fragkiadaki. Energy-based models as zero-shot planners for compositional scene rearrangement. *arXiv preprint arXiv:2304.14391*, 2023.

[21] I. Kapelyukh, V. Vosylius, and E. Johns. Dall-e-bot: Introducing web-scale diffusion models to robotics. *IEEE Robotics and Automation Letters*, 2023.

[22] Y. Dai, M. Yang, B. Dai, H. Dai, O. Nachum, J. Tenenbaum, D. Schuurmans, and P. Abbeel. Learning universal policies via text-guided video generation. *arXiv preprint arXiv:2302.00111*, 2023.

[23] A. Ajay, S. Han, Y. Du, S. Li, G. Abhi, T. Jaakkola, J. Tenenbaum, L. Kaelbling, A. Srivastava, and P. Agrawal. Compositional foundation models for hierarchical planning. *arXiv preprint arXiv:2309.08587*, 2023.

[24] K. Black, M. Nakamoto, P. Atreya, H. Walke, C. Finn, A. Kumar, and S. Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models. *arXiv preprint arXiv:2310.10639*, 2023.

[25] H. Chen, C. Lu, C. Ying, H. Su, and J. Zhu. Offline reinforcement learning via high-fidelity generative behavior modeling, 2023.

[26] B. Yang, H. Su, N. Gkanatsios, T.-W. Ke, A. Jain, J. Schneider, and K. Fragkiadaki. Diffusion-es: Gradient-free planning with diffusion for autonomous driving and zero-shot instruction following. *ArXiv*, abs/2402.06559, 2024.

[27] P. Hansen-Estruch, I. Kostrikov, M. Janner, J. G. Kuba, and S. Levine. Idql: Implicit q-learning as an actor-critic method with diffusion policies. *arXiv preprint arXiv:2304.10573*, 2023.

[28] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015.

[29] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[30] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *CoRR*, abs/2006.11239, 2020. URL https://arxiv.org/abs/2006.11239.

[31] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[32] P.-L. Guhur, S. Chen, R. G. Pinel, M. Tapaswi, I. Laptev, and C. Schmid. Instruction-driven history-aware policies for robotic manipulations. In *Conference on Robot Learning*, pages 175–187. PMLR, 2023.

[33] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

[34] J. J. Kuffner and S. M. LaValle. Rrt-connect: An efficient approach to single-query path planning. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 2, pages 995–1001. IEEE, 2000.

[35] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, P. Florence, C. Fu, M. G. Arenas, K. Gopalakrishnan, K. Han, K. Hausman, A. Herzog, J. Hsu, B. Ichter, A. Irpan, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, L. Lee, T.-W. E. Lee, S. Levine, Y. Lu, H. Michalewski, I. Mordatch, K. Pertsch, K. Rao, K. Reymann, M. Ryoo, G. Salazar, P. Sanketi, P. Sermanet, J. Singh, A. Singh, R. Soricut, H. Tran, V. Vanhoucke, Q. Vuong, A. Wahid, S. Welker, P. Wohlhart, J. Wu, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023.

[36] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.