STEER: Bridging VLMs and Low-Level Control for Adaptable Robotic Manipulation

Anonymous Author(s)
Affiliation
Address
email

Abstract: Recent advances have showcased the opportunity of leveraging the broad semantic understanding learned by vision-language models (VLMs) in robot learning; however, effectively connecting VLMs to robot control remains challenging due to the scarcity of physical robot data compared to internet-scale training data. We propose STEER, a system that learns flexible, low-level manipulation skills, allowing for modulation and adaptation to new situations. By training low-level policies on structured, dense re-annotations of existing robot datasets, we create an intuitive interface for humans or VLMs to guide robots in unfamiliar scenarios and perform new tasks using common-sense reasoning. Our results demonstrate that skills learned through STEER can be synthesized to accomplish held-out tasks without additional training. (Videos[1])

1 Introduction

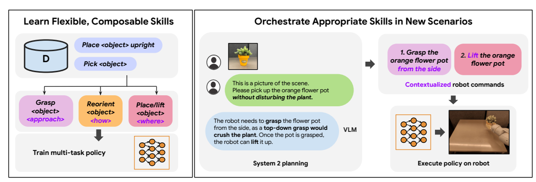


Figure 1: In STEER, we train on a dataset of diverse robot behaviors that is re-annotated to describe the primitive skills used to manipulate objects, with a focus on *how* the robot performed that skill. At inference time, a high-level system (VLM or human) receives a complex language instruction and determines the low-level skills to employ in the given context. Focusing on the "how" enables better contextual behavior.

Designing robots that can handle diverse and nuanced tasks posed by the real world is challenging, as it requires adaptability to complex, dynamic environments. Imitation learning (IL) is a widely-used, data-driven approach that distills expert demonstrations into learned policies, enabling precise manipulation of high-dimensional robot systems in real-world environments [1, 2] and at scale [3, 4, 5, 6, 7, 8, 9]. Despite these advances, robot systems trained with IL remain largely limited to scenarios encountered during training, which are fundamentally narrow as collecting real-world embodied data is costly and constrained by physical limitations. Humans, on the other hand, can adapt to complex, unfamiliar situations with ease, thanks to "common sense" generalization. Humans effortlessly understand high-level concepts like object affordances, intuitive physics, and compositionality—referred to as 'System 2' processing [10], which involves deliberate, analytical thinking. This contrasts with 'System 1' behaviors, which are reactive and particularly useful in contact-rich manipulation. Natural language serves as a key medium for studying System 2 reasoning, not only as a means through which humans understand and describe the world, but also as the primary input-output modality for vision-language models (VLMs) that demonstrate complex, human-like reasoning capabilities [11, 12, 13]. Connecting high-level System 2 plans to low-level System 1 behaviors, however, is not straightforward. Several methods have been proposed to bridge this gap: some enable System 2 reasoning systems to operate in modalities more easily transferable to robotic policies, such as code or semantic keypoints [14, 15], while others account for the lack of System 2 physical grounding by considering robot affordances during planning [16] or jointly training on both internet and embodied data [6, 5]. These approaches typically view System 1 processing as inflexible, seeking to improve System 2 reasoning outputs to better control a fixed System 1 policy. Instead of augmenting System 2 reasoning outputs, we ask: can we improve the System 1 policy to be more flexible and steerable by System 2 processes? Can this combination enable generalizable, end-to-end control?

[1] https://steer-anon.github.io

Submitted to the 8th Conference on Robot Learning (CoRL 2024). Do not distribute.

We introduce STEER: Structured Training for EmbodiEd Reasoning, a framework for training low-level reactive policies that can be flexibly guided by higher-level reasoning systems like humans or VLMs. The key insight of STEER is the use of dense language annotations on robot data, allowing us to train conditional policies based on detailed language instructions. These policies can be conditioned on each step of a plan generated by a high-level model (e.g., VLM or LLM), effectively combining System 1 and System 2 capabilities. This enables the robot to adapt to new situations by synthesizing behaviors not explicitly demonstrated during training. We implement this system using real-world datasets and propose an automated labeling pipeline based on proprioceptive observations to extract basic, object-centric manipulation skills, which are distilled into a low-level policy. Additionally, we present a strategy for using a VLM to generate language-based instructions for this low-level policy. Crucially, our approach enables the repurposing of robot skills in a semantically meaningful way at test time, allowing robots to autonomously handle novel situations.

2 Related Work

Imitation learning (IL) has become the dominant paradigm for training robotic manipulation policies [3, 2, 17]. However, deploying these models in unstructured environments remains challenging, as robot policies trained on human-collected data struggle in "out-of-distribution" scenarios where demonstrations are sparse [18]. This limitation arises from the high cost of collecting large-scale robot data compared to web-scale datasets used for training foundation models [19, 11, 13]. To improve generalization, researchers have leveraged text and vision foundation models to utilize existing datasets. This includes enhancing IL policies for open-world object grasping through open-vocabulary object detection [20] and relabeling episode-level instructions with models like CLIP [21, 22, 23]. Our work aligns with these dataset relabeling approaches, aiming to expand robot capabilities by relabeling behavior modes in existing heterogeneous demonstration datasets. Previous research has also investigated expressive modalities for policy conditioning, such as goal target poses [24], images [25, 26], trajectories [27, 28], and code [14]. However, natural language remains the primary modality for complex planning in state-of-the-art LLMs and VLMs, motivating STEER to improve language-conditioned action prediction. Additionally, many works explore learned skills to accelerate new task learning with temporally extended, semantically meaningful action sequences [29, 30, 31, 32, 33]. These approaches often employ hierarchical policies that learn to compose skills through RL [34, 35, 36]. While EXTRACT [33] uses VLMs to label skills for new tasks, our method leverages VLMs' common-sense reasoning to select appropriate skills without training a separate policy. By reasoning about how humans approach tasks from visual inputs, our framework enables robots to plan longer-horizon tasks and manage novel object configurations. This contrasts with prior work on affordances, which typically relies on keypoint representations in pixel space [37, 38]. Our approach instead reasons about affordances through natural language, allowing for more nuanced interactions with off-the-shelf VLMs or human operators.

3 System Design

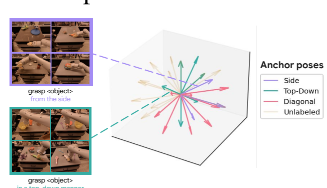


Figure 2: Anchor vectors and their semantic labels. Purple, green, and pink vectors represent side, top-down, and diagonal.

Our goal is to extract language-indexed, object-centric skills that facilitate task execution via foundation models. We achieve this by annotating existing datasets and training a language-conditioned RT-1 policy [4] using segmented and relabeled instructions. We extract semantically identifiable categories linked to language descriptions, focusing on shared, object-relational skills like grasping, lifting, placing, and rotating, originally demonstrated through templates like `pick <object>`, `move <object1> near <object2>`, `knock <object>`, `place <object> upright`, can be executed with varying strategies. Key factors include:

Grasp Angle. Objects can be grasped in multiple stable positions, and the particular way indeed impacts the ability to perform downstream tasks. However, grasp positions are rarely labeled apriori, as they are often implicit. We use a simple approach to label the grasp approach by manually labeling a relatively small set of 'anchor' grasp poses. We then label an arbitrary grasp with the label of its nearest neighbor 'anchor' pose as measured by cosine similarity. We represent a grasp pose as a 3D unit vector, and we identify the time of a grasp where the gripper changed from fully open to fully closed. To define and label the anchor poses, we took 3D unit vectors that are linear combinations of the elementary 3D basis vectors and visually inspected clusters in order to label them. In the grasp data, we identify three distinct modes via inspecting the grasp images: top-down grasps, side grasps, and diagonal grasps (visualized in Figure 2). The sub-trajectory is relabeled to grasp the <object> in a <grasp approach>, where <object> is from the original instruction and <grasp approach> is from the anchor's label.

Reorientation. Another mode of behavior identified in the dataset is of reorienting objects. In order to identify and label these reorientations, we first label the wrist orientation for every timestep where the gripper is fully closed. Then if the gripper orientation switches between two of the modes (as labeled in *Grasp Angle*), we label the sub-trajectory preceding it as reorient the <object> <direction>, where <object> again is from the original instruction and <direction> indicates whether the object is rotated from upright to horizontal or vice versa.

Lifting/Placing. Complementing grasping, we label whether the object was lifted or placed at the end of completing the original task. If the object is still held at the end of the episode and the gripper moves vertically upward, we label the final sub-trajectory as hold and lift the <object>. If not, similar to identifying grasps, we identify the time of placing using the gripper state and label this sub-trajectory as place the <object>.

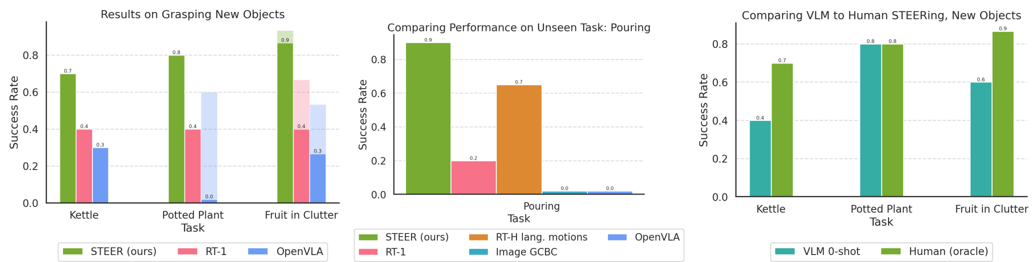
Orchestrating Learned Skills A key capability of the System 2 component is being able to reason about the visual observation of the scene, the task description, and the robot's low-level capabilities to effectively choose and sequence appropriate skills for the task at hand. To implement an automated System 2 component as a code-writing VLM agent in order to autonomously execute the verbalized plans without additional modules or human effort. To facilitate this, we define an API for the action primitives accessible by the VLM to interface with the System 1, reactive low-level RT-1 policy skills as described in Section 3. The API is based on translating the language commands into a simple API that the VLM agent can access. This breakdown is based on *what* the robot should do and *how* to do it. Each primitive skill (i.e. grasping, rotating, lifting, placing) is represented by a function with a keyword argument modifying *how* that primitive is accomplished (i.e. grasp(object, "top-down"). Internally, the API translates this code into the corresponding natural language the RT-1 policy was trained on. We use a system prompt to tell the VLM to control its physical embodiment through code, then provide the robot's visual observation of the scene and a description of the high-level task. The exact system prompt we use in all experiments and example outputs and explanations produced by the model can be found on our project website[1].

4 Experiments

We evaluate STEER by testing its ability to improve grasping in unseen scenarios and perform novel behaviors that require complex reasoning and motor control. We focus on three main questions: (1) Does learning multiple modes of behavior improve adaptability in new situations? (2) Can combining extracted skills from heterogeneous human demos enable entirely new tasks? (3) To what degree can a state-of-the-art VLM plan orchestrate these skills autonomously?

We use a 7 DoF arm, a two-fingered gripper, and a mobile base, as used in RT-1 [4] in a tabletop environment. The experiment involves 70K demonstrations from RT-1's multi-task dataset and 15K grasping demos from MOO [20]. We choose RT-1 [4] for our System 1 component and Gemini 1.5 Pro [39] as our learned System 2 component for those experiments (sample videos[1]).

Improving Test-Time Adaptability We present three challenging, unseen grasping scenarios: a kettle, a potted plant, and grasping in clutter. We compare STEER with the baseline RT-1 [4] (trained on original instructions) and OpenVLA [8], which fine-tunes a VLM on robot data from Open X-Embodiment [40]. We report the success rates in Figure 3a. RT-1 occasionally succeeds, but exhibits different strategies and we observe that failures are often caused by a sub-optimal approach. OpenVLA performed similarly to RT-1, demonstrating that additional web data does not lead to sufficiently strong embodied reasoning about how to grasp in a new scenario where a particular approach is evidently necessary. For example, we find that OpenVLA often picks the potted plant up, but does not respect the language instruction of picking up the flower pot *without disturbing*



(a) Grasping results. Full successes are in solid colors. Partial successes are in light colors. We do 20 trials of `Kettle`, 10 of `Potted Plant`, and 15 of `Fruit in Clutter`.

(b) New task results. We run each method 10 times, comparing the low-level capabilities afforded by each model to perform the task usin human guidance.

(c) We compare a VLM to a human in wielding the learned policies. The VLM can effectively recover most of the performance.

Figure 3: Results on grasping in unseen scenarios and performing a new task, with human or VLM guidance. We find that by having access to and being able to reason about extracted low-level strategies enables higher success in OOD scenarios than the baseline RT-1 model and a state-of-the-art VLA.

the plant and grasps from above around the plant leaves. Decomposing the grasp strategies and exploiting the most suitable one as we do in STEER reduces this failure mode.

Performing Novel Behaviors We study whether we can engineer behavior for a new everyday task without collecting new demonstrations or additional fine-tuning. Pouring is out of the distribution of demonstrated tasks but should be achievable with the motions that exist in the data. We compare against the **best-case** version of each of 4 baselines and comparisons: baseline **RT-1** [4], **Language motions from RT-H** [41], defined by narrating end-effector movement to give language like `move arm left` and `rotate arm right`, **a goal-image conditioned variant of RT-1**, which tests whether language is a better abstraction layer than goal images, and **OpenVLA** [8]. As seen in Figure 3b, human orchestration with a STEER policy achieves a 90% success rate on pouring as compared to 70% with a policy trained with language motions from RT-H (whose orchestration is significantly more cumbersome as it requires tight closed-loop guidance). In comparison, baseline RT-1 cannot complete the task because it is not trained to reorient objects. The goal image conditioned baseline, despite having demonstration sub goal images from the same starting positions, fails and appears to mimic the exact arm positions in the subgoals rather than manipulate the object state as prescribed by the goal image. OpenVLA, despite having access to the same underlying demo data, does not generalize to the new motion by stitching together the appropriate motions.

VLM Orchestration Now, we test whether a *VLM* can effectively select or sequence appropriate skills afforded by STEER by reasoning about the context, in the visual observation and task description, as well as the skills exposed through the API *without any examples* (i.e. 0-shot). For these experiments, we compare the VLM to human orchestration of the same low-level policy to serve as an upper bound on performance. Exact inputs and outputs can be found on the project website[1].

Seen task, new scenarios. We see that the VLM successfully produces the same high-level plans as the human expert very reliably for the grasping tasks. However, as shown in Figure 3c we see that there is a degradation in end-to-end task performance compared to human orchestration when executing the code produced by the VLM, and we analyze these failures. For the kettle picking task, we note that the low-level policy appears to be sensitive to the specific naming of objects. That is, the VLM often produced code to grasp the ‘black and white kettle‘ from the top instead of grasping the ‘black and white object‘ from the top, and with further analysis find that this instruction has a noticeable degradation across all low-level language-conditioned policies. So, while the VLM reasonably commands the policy to grasp from above, the low-level policy is less reliable. We expect this to be improved with denser annotation or augmentation on the entity-level, whereas STEER is concerned with the motion-level. For the `Fruit in Clutter` grasping task, the VLM did not always command the appropriate action and we suspect that similar object naming references (‘red apple’ instead of ‘apple’) impact the low-level policy performance.

Seen objects, new task. Without any examples, the VLM correctly identifies that in order to pour from the cup, the robot ought to grasp it from the side as if a human were performing the task. It then recognizes that it must reorient it, then reorient it back in order to place it back upright on the table. The VLM succeeded in *6 out of 10 trials for zero-shot synthesizing of pouring behavior*.

References

[1] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *Robotics: Science and Systems (RSS)*, 2023.

[2] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 2024.

[3] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn. BC-z: Zero-shot task generalization with robotic imitation learning. In *5th Annual Conference on Robot Learning*, 2021. URL https://openreview.net/forum?id=8kbp23tSGYv.

[4] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, K.-H. Lee, S. Levine, Y. Lu, U. Malla, D. Manjunath, I. Mordatch, O. Nachum, C. Parada, J. Peralta, E. Perez, K. Pertsch, J. Quiambao, K. Rao, M. Ryoo, G. Salazar, P. Sanketi, K. Sayed, J. Singh, S. Sontakke, A. Stone, C. Tan, H. Tran, V. Vanhoucke, S. Vega, Q. Vuong, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich. Rt-1: Robotics transformer for real-world control at scale. In *arXiv preprint arXiv:2212.06817*, 2022.

[5] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

[6] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

[7] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, C. Xu, J. Luo, T. Kreiman, Y. Tan, L. Y. Chen, P. Sanketi, Q. Vuong, T. Xiao, D. Sadigh, C. Finn, and S. Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

[8] M. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[9] R. Doshi, H. Walke, O. Mees, S. Dasari, and S. Levine. Scaling cross-embodied learning: One policy for manipulation, navigation, locomotion and aviation. *arXiv preprint arXiv:2408.11812*, 2024.

[10] D. Kahneman. *Thinking, fast and slow*. Farrar, Straus and Giroux, New York, 2011. ISBN 9780374275631 0374275637. URL https://www.amazon.de/Thinking-Fast-Slow-Daniel-Kahneman/dp/0374275637/ref=wl_it_dp_o_pdT1_nS_nC?ie=UTF8&colid=151193SNGKJT9&coliid=I3OCESLZCVDFL7.

[11] OpenAI et al. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

[12] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL https://arxiv.org/abs/2201.11903.

[13] G. Team et al. Gemini: A family of highly capable multimodal models, 2024. URL https://arxiv.org/abs/2312.11805.

[14] 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)*, pages 9493–9500. IEEE, 2023.

[15] N. Di Palo and E. Johns. Keypoint action tokens enable in-context imitation learning in robotics. *arXiv preprint arXiv:2403.19578*, 2024.

[16] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *6th Annual Conference on Robot Learning*, 2022.

[17] S. Lee, Y. Wang, H. Etukuru, H. J. Kim, N. M. M. Shafiullah, and L. Pinto. Behavior generation with latent actions, 2024. URL https://arxiv.org/abs/2403.03181.

[18] O. Mees, L. Hermann, and W. Burgard. What matters in language conditioned robotic imitation learning over unstructured data. *IEEE Robotics and Automation Letters (RA-L)*, 7(4):11205–11212, 2022.

[19] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

[20] A. Stone, T. Xiao, Y. Lu, K. Gopalakrishnan, K.-H. Lee, Q. Vuong, P. Wohlhart, S. Kirmani, B. Zitkovich, F. Xia, C. Finn, and K. Hausman. Open-world object manipulation using pre-trained vision-language models, 2023.

[21] T. Xiao, H. Chan, P. Sermanet, A. Wahid, A. Brohan, K. Hausman, S. Levine, and J. Tompson. Robotic skill acquistion via instruction augmentation with vision-language models. In *Proceedings of Robotics: Science and Systems*, 2023.

[22] V. Myers, B. C. Zheng, O. Mees, S. Levine, and K. Fang. Policy adaptation via language optimization: Decomposing tasks for few-shot imitation, 2024. URL https://arxiv.org/abs/2408.16228.

[23] J. Zhang, K. Pertsch, J. Zhang, and J. J. Lim. Sprint: Scalable policy pre-training via language instruction relabeling. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9168–9175. IEEE, 2024.

[24] A. Gupta, V. Kumar, C. Lynch, S. Levine, and K. Hausman. Relay policy learning: Solving long horizon tasks via imitation and reinforcement learning. *Conference on Robot Learning (CoRL)*, 2019.

[25] 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*, abs/2310.10639, 2023. URL https://api.semanticscholar.org/CorpusID:264172455.

[26] K. Fang, P. Yin, A. Nair, and S. Levine. Planning to practice: Efficient online fine-tuning by composing goals in latent space. *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4076–4083, 2022. URL https://api.semanticscholar.org/CorpusID:248834175.

[27] J. Gu, S. Kirmani, P. Wohlhart, Y. Lu, M. G. Arenas, K. Rao, W. Yu, C. Fu, K. Gopalakrishnan, Z. Xu, P. Sundaresan, P. Xu, H. Su, K. Hausman, C. Finn, Q. Vuong, and T. Xiao. Rt-trajectory: Robotic task generalization via hindsight trajectory sketches. In *Robotics: Science and Systems (RSS)*, 2024.

[28] C. Wen, X. Lin, J. So, K. Chen, Q. Dou, Y. Gao, and P. Abbeel. Any-point trajectory modeling for policy learning. *Robotics: Science and Systems (RSS)*, 2024.

[29] Y. Lee, S.-H. Sun, S. Somasundaram, E. S. Hu, and J. J. Lim. Composing complex skills by learning transition policies. In *International Conference on Learning Representations*, 2019.

[30] A. Gupta, V. Kumar, C. Lynch, S. Levine, and K. Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. *CoRL*, 2019.

[31] M. Dalal, D. Pathak, and R. R. Salakhutdinov. Accelerating robotic reinforcement learning via parameterized action primitives. *Neural Information Processing Systems (NeurIPS)*, 34: 21847–21859, 2021.

[32] S. Nasiriany, H. Liu, and Y. Zhu. Augmenting reinforcement learning with behavior primitives for diverse manipulation tasks. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

[33] J. Zhang, M. Heo, Z. Liu, E. Biyik, J. J. Lim, Y. Liu, and R. Fakoor. Extract: Efficient policy learning by extracting transferrable robot skills from offline data. *arXiv preprint arXiv:2406.17768*, 2024.

[34] P.-L. Bacon, J. Harb, and D. Precup. The option-critic architecture. In *Proceedings of the AAAI conference on artificial intelligence*, 2017.

[35] O. Nachum, S. S. Gu, H. Lee, and S. Levine. Data-efficient hierarchical reinforcement learning. *Neural Information Processing Systems (NeurIPS)*, 31, 2018.

[36] X. B. Peng, M. Chang, G. Zhang, P. Abbeel, and S. Levine. Mcp: Learning composable hierarchical control with multiplicative compositional policies. *Advances in Neural Information Processing Systems*, 32, 2019.

[37] T. Nagarajan, C. Feichtenhofer, and K. Grauman. Grounded human-object interaction hotspots from video. In *International Conference on Computer Vision (ICCV)*, 2019.

[38] S. Bahl, R. Mendonca, L. Chen, U. Jain, and D. Pathak. Affordances from human videos as a versatile representation for robotics. 2023.

[39] G. Team. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024. URL https://arxiv.org/abs/2403.05530.

[40] O. X. E. Collaboration et al. Open x-embodiment: Robotic learning datasets and rt-x models : Open x-embodiment collaboration0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[41] S. Belkhale, T. Ding, T. Xiao, P. Sermanet, Q. Vuong, J. Tompson, Y. Chebotar, D. Dwibedi, and D. Sadigh. Rt-h: Action hierarchies using language. *arXiv preprint arXiv:2403.01823*, 2024.