Smash and Spread! Teaching Robots to Transform Objects via Spatial Progress

Priyanka Mandikal, Jiaheng Hu, Shivin Dass, Sagnik Majumder
Roberto Martin-Martin*, **Kristen Grauman***
The University of Texas at Austin
mandikal@utexas.edu

Abstract: While current robot manipulation often focuses on changing the *positions* of objects (e.g., pick-and-place), a wide range of real-world human manipulation involves non-rigid *object state changes*—such as smashing, spreading, or chopping—where an object's visual and physical state evolve gradually over time. Our core insight is that many of these tasks share a common structural pattern: they involve spatially progressing, object-centric transformations that can be represented as regions transitioning from an *actionable* to a *transformed* state. Building on this insight, we integrate spatially progressing object change segmentation maps to provide dense visual affordance cues, using them both as policy observations and to automatically generate rewards reflecting the extent of visual transformation over time. Our formulation enables highly sample-efficient online reinforcement learning without demonstrations, simulation, or costly manual reward annotation. Furthermore, thanks to the abstraction into spatially transformed areas, our method allows direct generalization to new manipulated objects. We validate our SPARTA approach on a real robot for two challenging and previously unaddressed tasks—spreading and smashing—across 6 diverse real-world objects, achieving improvements of 79% in training time and 41% in accuracy over sparse rewards and visual goal-conditioned baselines. More information at https://sites.google.com/view/sparta-robot

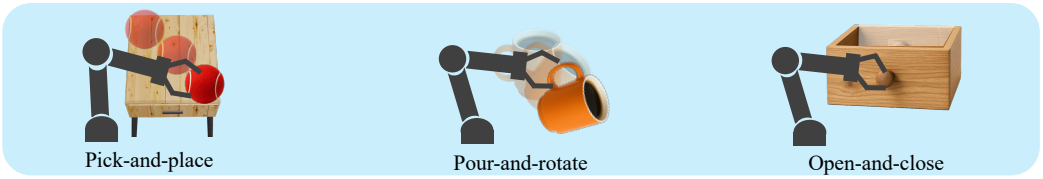
Keywords: object manipulation, visual representation, real-world reinforcement learning

1 Introduction

The status quo in robotic manipulation emphasizes tasks involving object motion, such as pick-and-place [1], opening and closing [2, 3], pushing [4], and rotating [5] objects. While these tasks are foundational, they largely entail changing the kinematic state of rigid or semi-rigid bodies, where progress on the task is readily visible and monitorable via pose changes. In contrast, many real-world tasks demand *object state changes* (OSC)[1] [6, 7, 8]—where the object's visual appearance is gradually transformed, without necessarily altering its pose (see Fig. 1). Examples of manipulation tasks involving OSC are ubiquitous in home domains, such as smashing (e.g., banana into a purée, clay into a disc), spreading (e.g., jam on bread, paint on a chair), or chopping (e.g., avocado). Such operations involve physical interactions that fundamentally alter the object's shape, texture, and color, making the task both mechanically challenging and visually complex. Crucially, the object may remain in place, but its visual and material states evolve gradually—presenting challenges well beyond those in traditional pick-and-place tasks, yet critical in domains like cooking (e.g., grating, peeling, shredding) and household chores (e.g., painting, wiping). See Figure 1.

[1]Here we adopt the term "object state change" (OSC) as in the vision literature: an OSC is a transformation of an object that entails a visually distinct post-condition (e.g., chopped apple) following an action imposed on it (e.g., chopping), often with irreversible changes to the object's morphology, texture, and appearance.

(a) Status quo in robot manipulation: **rigid body motion**



(b) Our goal: learn real-world object manipulation tasks involving **object state changes**

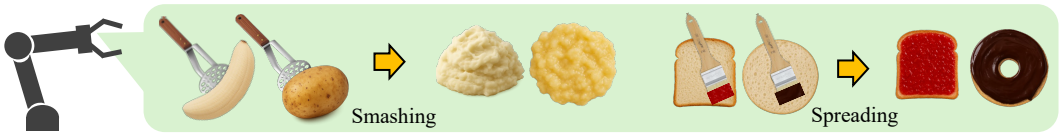


Figure 1: Top: The status quo in robotic manipulation today heavily focuses on rigid-body motion (e.g., pick-and-place, open-and-close, pour-and-rotate, etc). Bottom: However, a wide range of real-world human manipulation involves *object state changes*—such as smashing or spreading —where an object's visual state evolves gradually over time, often in an irreversible way. We introduce a novel vision-based RL approach to capture these fine-grained, spatially-progressing transformations, successfully demonstrating how to guide real robot manipulation for this family of tasks.

What does it take to enable robots to learn such complex object manipulation capabilities? Unlike motion-centric tasks, OSC tasks require the robot to interact with partially transformed, non-rigid objects. This involves: 1) visually distinguishing actionable regions (yet-to-be-transformed) from transformed regions, 2) learning where to act next based on visual cues, and 3) executing precise, progressive transformations to incrementally achieve the desired end state.

To tackle this challenge, we propose SPARTA (Spatial Progress-Aware Robotic object TransformAtion)—a robotic system that leverages progress-aware object state change segmentation maps to guide learning and execution of OSC tasks. Inspired by the spatially-progressing object state change (SPOC) task [9] in computer vision, we explore how fine-grained visual affordances geared toward object states can inform robot learning. Specifically, we focus on two representative OSC tasks—spreading (coating) and smashing—and develop an online real-world reinforcement learning (RL) paradigm to train agents to perform them. In SPARTA, a transforming object is segmented into two regions: actionable and transformed. For example, in smashing a potato, unsmashed chunks represent actionable areas, while the smashed portions are transformed. We use these SPOC affordance[2] maps to generate two things: 1) an additional visual observation channel that accelerates training and 2) a temporally dense, spatially-informed reward signal, namely the percentage of newly transformed object area relative to the total object area after each policy action. Our formulation produces task-oriented visual information and dense, smooth, monotonic reward curves that make real-world RL possible and *highly* sample-efficient compared to existing alternatives.

In our experiments, we show that with just 3 hours of online RL training and *no human demonstrations*, SPARTA successfully learns manipulation policies. We evaluate on a variety of real-world objects, demonstrating the robustness and generality of our approach. In contrast, baselines using sparse task rewards fail to learn due to the lack of intermediate supervision. On the other hand, goal-conditioned representation learning approaches (e.g., LIV [10]), struggle to model the fine-grained, visually complex OSC tasks—often collapsing to near-sparse reward behavior.

These results highlight the importance of dense visual affordance representations tailored to object state transformations, and lay the foundation for a broader class of robotic manipulation skills that go beyond rigid-body motion and can be treated with a common, holistic approach. Furthermore, our work is the first to successfully translate the visual effects of human action—as gleaned from widespread real-world video—to real-robot policies for such challenging manipulation tasks.

[2]Here "affordance" means regions requiring robot interaction, distinct from conventional grasp points.

2 Related Work

Non-rigid object manipulation. Recent advances tackle individual tasks requiring more complex manipulation than traditional pick-and-place-style tasks, such as cutting [11, 12, 13, 14], peeling [15, 16, 17], and stir-frying [18]. Cutting has been extensively studied through physical modeling of tool-material interactions, e.g., RoboNinja [12], DiSECt [11], RoboCook [14], and SliceIt! [13]. Peeling tasks rely on tactile sensing to detect peeled regions [15, 17], or learn reorientation policies for dexterous in-hand manipulation [16]. However, these efforts often focus on the task's mechanical aspects, lack general-purpose vision feedback, or rely heavily on simulation. In contrast, our work targets a broad class of spatially transformative tasks that require reasoning over visual state changes rather than contact dynamics alone. Recent work uses food state classifiers trained with *manual* labels to guide cooking plans [19], but operates on discrete state changes (e.g., raw to cooked), lacks spatial awareness, and does not offer continuous feedback to control policies. Across the board, prior approaches either depend on high-fidelity simulation, contact dynamics, or manually defined state changes—limiting their scalability and generalization.

Visual representations for robot learning. Another way to accelerate downstream policy learning is by pretraining visual representations [20, 21, 22, 23, 24, 10]. Early efforts used labeled Internet images [21] or robot-specific data [20], while recent methods pretrain on large-scale human activity videos [8, 25]. More relevant to our novel visual rewards, VIP [24] learns an implicit value function over egocentric videos, while its extension LIV [10] further incorporates language-goal embeddings. There is also growing interest in using LLMs [26] and video-language models (VLMs) [27, 28] for robotic reasoning, typically using coarse frame-level goal matching or symbolic planning. In contrast, SPARTA leverages a VLM for spatial reasoning over localized object regions, enabling dense reward generation and efficient online RL for visually complex manipulation.

Affordances in robotics. Understanding *how* and *where* to interact with objects has driven a surge of interest in affordance-based functional grasping [29, 30, 31, 32], including dexterous grasping [30], tool use [33], and drawer manipulation [34]. Parallel efforts in computer vision predict hand-object interactions [35, 36, 37, 38, 39], but they emphasize pick-and-place or grasping-style tasks. In contrast, we tackle a fundamentally different class of affordance—spatially evolving, visual object state transformations that generalize across tasks and robot embodiments. To our knowledge, this represents the first affordance reasoning approach for such manipulations, and drawing on in-the-wild human video we achieve non-rigid object interactions on a real robot.

Object state change understanding. Object state change is explored in computer vision for video-level state classification [6, 7], object segmentation [40, 41, 9], and modeling procedural changes [42]. Our work is inspired in part by the *spatially progressing object state change* (SPOC) task [9], which segments state-changing objects into actionable and transformed regions. Trained on large-scale instructional video [43], SPOC demonstrates robust spatial reasoning across diverse objects and transformations. However, all these models are vision-only [6, 7, 40, 41, 9]: they passively analyze state changes but are not used to inform robot control. Our work bridges that gap. By integrating vision-based OSC understanding into robot manipulation, we show how robots can learn to act using SPOC-style affordance maps capturing gradual visual progress, which is difficult to address with tactile sensing [17], force models, or binary state classifiers [2, 19].

3 Approach

Our goal is to enable robots to perform object state transformation where the object's visual and physical properties change gradually over time, and often in an irreversible, non-rigid way. This goes beyond traditional manipulation focused on object motion, requiring the robot to perceive, reason about, and progressively transform specific regions of the object. To tackle this, we propose a sample-efficient online reinforcement learning framework guided by novel dense visual affordances that indicate where and how much of the object has changed, enabling structured policy learning through spatially grounded reward signals.

Spatially-progressing object state changes occur in a variety of real-world domains (cooking, cleaning, arts and crafts, construction, etc.). We focus our study on actions from the cooking domain,

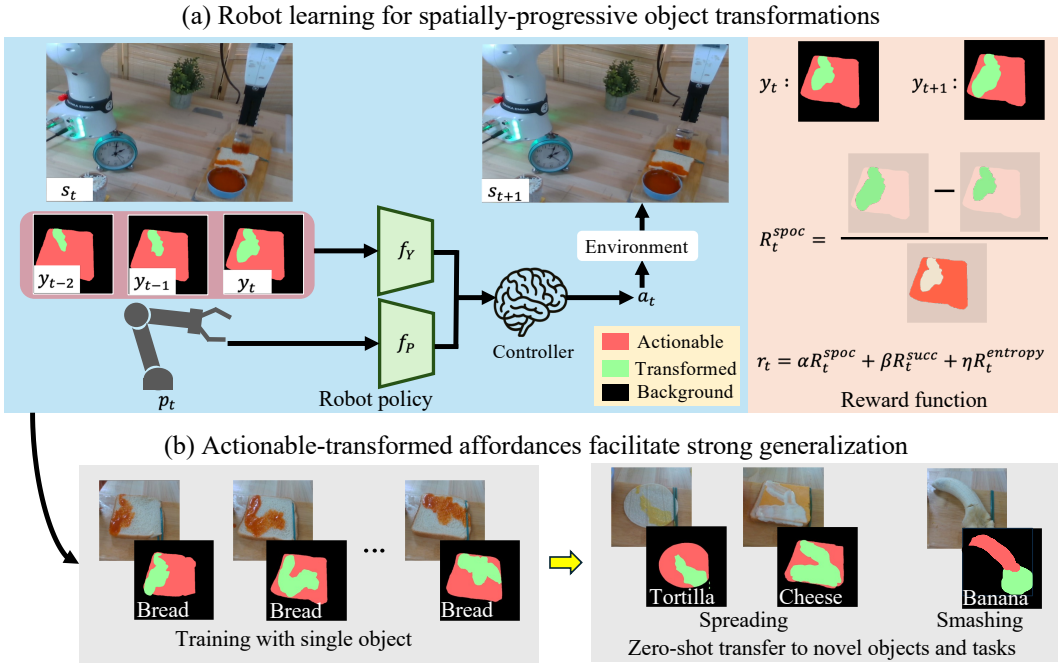


Figure 2: **Overview of SPARTA.** Top: At each episode step, our policy takes the current and past SPOC [9] visual-affordance (segmentation) maps as inputs, along with the robot arm's proprioception data and predicts a displacement action for the arm's end-effector. We train the policy using RL with a novel reward function that incentivizes the robot to keep transforming the actionable object regions as efficiently as possible. Bottom: Using visual-affordance map inputs facilitate zero-shot transfer to novel objects of vastly different shapes, texture and color (e.g., tortilla or cheese vs. bread) and novel tasks (e.g., smashing vs. spreading).

due to its real-world application value, complexity, and wide array of state-changing actions and constituent objects. We now describe the components of our system in detail.

SPOC visual affordances learned from human videos .Our method builds on the SPOC task [9] introduced in computer vision. Given a sequence of video frames $i_1, i_2, \ldots, i_T$, the goal is to generate masks $\mathcal{M}(i_t) = \{m_t^{act}, m_t^{trf}\}$ for each frame, segmenting regions that are yet to be transformed (actionable) and already transformed. See red-green maps in Figure 2. For example, in a chopping task, both whole and large cut avocado chunks are considered actionable until fully chopped. This formulation enables fine-grained, temporally consistent segmentation of evolving object states.

We adapt this idea for robotics by generating SPOC affordance maps directly from real-time visual observations. While prior work [9] leverages Grounded-SAM [44] and CLIP [45], we find that replacing CLIP with a vision-language model (VLM) such as GPT-4o [46] significantly improves segmentation accuracy—particularly in distinguishing intra-object regions. Furthermore, we introduce tracking of the transformed regions to boost real-time throughput for robotic control (see Appendix for details). These affordance maps offer dense, object-centric structure that is crucial for shaping progress-based rewards and guiding spatial-aware policy learning.

We call attention to two key aspects of our design. First, the SPOC masks are learned from large-scale video of *human* action in natural environments, and SPARTA translates that visual know-how into *robot* actions. At the same time, since our video-based learning is object-centric, what SPARTA takes away from the human observations is not the specific motions they use (which can suffer from embodiment mismatch), but rather *what altered visual states* they achieve. econd, by transforming RGB frames into binary actionable/transformed segmentations, our approach allows generalization to novel objects, e.g., learning to spread butter on bread then generalizing to spread paint on a canvas. That is because the binary SPOC maps directly surface to our reinforcement learning model

(detailed below) where robot action has been successful and where it is still needed—no matter the specific visual appearance and shape of the target object. See Figure 2b.

Robot policy learning. We formulate our problem as a Markov Decision Process (MDP) with state space $\mathcal{S}$, action space $\mathcal{A}$, transition dynamics $\mathcal{T}$, initial state distribution ρ_0, reward function r, and discount factor γ. Our goal is to learn a policy π that maximizes expected accumulated discounted rewards: $J(\pi) = \mathbb{E}_\pi[\sum_{t=0}^{T-1} \gamma^t r(s_t, a_t)]$. Building on SERL [47], we use Soft Actor-Critic (SAC) [48] with regularization from RLPD [49] for sample-efficient off-policy learning directly in the real world. Unlike SERL, however, our method does not rely on any human demonstrations. Instead, we show how dense, visually grounded rewards derived from SPOC affordances are sufficient to drive sample-efficient policy learning in the real world, as described next.

Manipulation tasks. We evaluate our approach on two challenging object state-change manipulation tasks: (1) *Spreading*: Spread a coatable substance across the surface of an object (e.g., syrup on a donut, sauce on pizza), resulting in visible changes in color and texture; and (2) *Smashing*: Crush an object into a purée-like consistency (e.g., banana or avocado), altering its texture from solid to semi-fluid. Both tasks involve significant and irreversible appearance and structural transformations, presenting considerable challenges for perception, affordance reasoning, and reward shaping.

State space. The robot operates in a workspace where a single object is on a tabletop at a random orientation. The state space is defined as $\mathcal{S} = M \times P$, consisting of visual affordance and proprioceptive inputs. At each timestep t, the raw visual observation $i_t \in I$ is an RGB image captured by a fixed front-mounted camera. The affordance input $m_t \in M$ is the SPOC segmentation map inferred from i_t, denoted as $\mathcal{M}(i_t) = \{m_t^{act}, m_t^{trf}\}$, representing the actionable and transformed object regions. The proprioceptive input $p_t \in P$ includes the robot's end-effector position.

Importantly, while SPOC processes RGB images I, the robot policy only sees the SPOC maps M as the visual inputs. SPARTA's abstracted actionable-transformed visual encoding accelerates generalization by decoupling visual affordance prediction from policy learning. A strong vision model pretrained on human videos does the heavy-lifting by generalizing to different objects, while the robot policy inputs are interpretable affordance maps that transfer across objects and tasks.

Action space. We use a 7-DoF joint-controlled Franka Emika robot equipped with a operational-space impedance controller SERL [47]. For all tasks, the appropriate tool (e.g., brush or smasher) is pre-grasped at the start of the episode. Prior work demonstrates that the action space for policy learning has a significant effect on performance [50, 51, 52]. Without loss of generality in our tasks, we align the robot's motion with the structure of state-changing tasks by constraining the action space to a 2D manifold over the object surface. The policy outputs continuous Δx and Δy displacements, sampled from a multivariate Gaussian centered at the predicted mean π, allowing smooth end-effector motion across the object workspace. The vertical (z) interaction is determined based on simple, task-agnostic contact strategies: for spreading tasks, the z-height is fixed based on the object's estimated surface; for smashing tasks, the robot autonomously lowers the end-effector until a preset safety threshold force is detected, initiating interaction. This formulation naturally reflects the spatial and progressive nature of object state-change tasks, while keeping the action space compact to enable fast, sample-efficient learning across diverse objects and manipulation skills.

Reward function. Prior work identifies several challenges hindering RL for policy learning in the real world [53], including the critical need to automate the generation of rewards to avoid costly human annotation and the importance of dense reward signals for sample-efficient training. Our approach brings a fresh perspective to addressing these core challenges in OSC manipulation.

A naive sparse reward for successful task completion would fall short in this context, e.g., a half-coated bread is closer to success than an untouched one, but this nuance is lost with simple binary rewards. Instead, we design a dense, spatially grounded reward function that guides the agent toward actionable regions and enables *real-time, demonstration-free* learning. The reward function combines three components:

$$r_t = \alpha R_t^{spoc} + \beta R_t^{succ} + \eta R_t^{entropy}. \tag{1}$$

Here, R_t^{succ} is a sparse reward given only upon full task completion (when the robot has transformed 95% of the object), and $R_t^{entropy}$ promotes action diversity and exploration. The key novel component, R_t^{spoc}, provides dense feedback at every step by quantifying visual progress in the transformation. To compute R_t^{spoc}, we use SPOC segmentation masks m_t^{act}, m_t^{trf}, which identify the object's actionable and transformed regions. The reward reflects the *newly transformed* area since the previous timestep:

$$R_t^{spoc} = \frac{A_{t+1}^{trf} - A_t^{trf}}{A_t^{act}} \tag{2}$$

where A_t^{trf} and A_t^{act} denote the areas of transformed and actionable regions at time t, respectively. This encourages the agent to consistently make progress by transforming new parts of the object, while focusing attention on regions most amenable to change.

This formulation is object-centric, task-agnostic, and compatible with a variety of manipulators. It enables the robot to learn from vision alone, without access to simulation, privileged state, or human demonstrations. As demonstrated below, our proposed reward design yields smooth, monotonic learning curves and enables real-time training of policies for complex state-changing manipulation tasks from scratch. Altogether our formulation leverages fine-grained visual understanding of object state changes to efficiently train an RL policy for complex manipulation tasks.

Implementation details. For SPOC segmentations, we apply GroundingDino [54] (for object detection) and SAM [55] (for mask proposal generation) on the first frame, using GPT-4o [46] as our vision encoder for determining actionable-transformed labels for object sub-regions. Once a transformed object mask is detected, we track these mask progressions using DeAOT [56] at every time step. For training the robot policy, we follow the real-world training setup of SERL, which uses SAC [48]. While SERL uses RLPD [49], a variant of SAC, to incorporate human demonstrations to bootstrap RL, our work uses no human demonstrations, and solely relies on dense SPOC rewards to boost sample efficiency. The SPOC map affordance inputs are processed by a 10-layer ResNet model, while the robot proprioception inputs are processed using a 2-layer fully-connected encoder of dimension [512,512]. The coefficients in the reward function (Eq. 1) are $\alpha = 1, \beta = 1, \eta = 0.001$.

4 Experiments

Comparisons. We compare our method against three baselines:

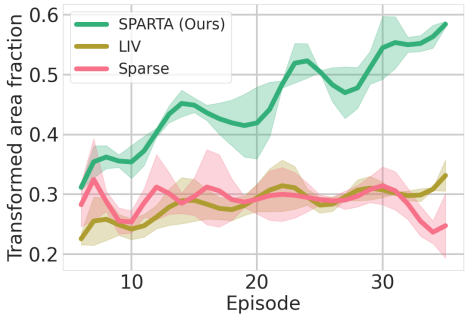
(**1**) RANDOM: a control baseline where actions are sampled uniformly at random within the constrained action space. This reflects unstructured exploration with no task guidance.

(**2**) SPARSE: a sparse reward baseline that uses only a binary task success signal and entropy regularization. Task success is determined by querying GPT-4o with the final image, using task-specific prompts (e.g., "Is the bread fully coated with ketchup?"). If the response is "yes," the episode ends with a reward of +1; otherwise, it continues without reward. The use of a VLM here parallels our use of a VLM to get the SPOC segmentation masks.

(**3**) LIV [10]: a self-supervised goal-based representation learning method trained on human activity videos [8, 25]. Rewards are computed from state embedding similarities to a language goal. We directly prompt LIV with a natural language description of the OSC task and object (e.g., "coat the bread with ketchup").

The baselines represent two dominant approaches: sparse rewards with minimal supervision and pretrained goal-based representations. They highlight the limitations of current visual RL methods when applied to fine-grained tasks. We do not include tactile-based or simulation-heavy methods [11, 12], as they require task-specific instrumentation. We focus on general, vision-driven approaches requiring no human demonstrations—hence directly comparable to SPARTA.

Metric. We evaluate performance using *transformation coverage*: the percentage of the object that has been visually transformed by the end of the episode, computed using SPOC segmentations then manually verified/fixed, reflecting the robot's degree of progress. As a continuous metric, it captures partial task completion and hence provides a more informative signal than binary success.



Model	SPOC	*Seen*		*Unseen*		
RANDOM	✗	0.24	0.42	0.27	0.29	0.23
SPARSE	✗	0.14	0.10	0.11	0.13	0.13
LIV [10]	✗	0.17	0.14	0.12	0.16	0.12
SPARTA-G	✓	0.44	0.49	0.55	**0.66**	0.39
SPARTA-RL	✓	**0.61**	**0.55**	**0.58**	0.63	**0.42**

Figure 3: **Left:** SPARTA's dense rewards align with the progressive spatial transformation of the object, whereas the baselines are slower to improve their policies. **Right:** SPARTA shows strong training and generalization results for *spreading* with held-out objects with varying textures, colors and shapes. Metric is transformation coverage (%).

Figure 4: **Left:** Progress of SPARTA's reward in a typical training episode; overlayed SPOC [9] affordance maps show how SPARTA continuously transforms actionable regions during an episode. **Right:** Policy trained on spreading ketchup on bread is able to transfer to smashing banana.

Objects. We evaluate our approach on a diverse set of real-world objects spanning different shapes, textures, and colors in order to test both the visual model's robustness and the control policy's generalization. Specifically, for spreading, we use bread, tortilla, and cheese slices, and spread them with ketchup, mustard, and ranch. For smashing, we use bananas.

Training. We begin by training a policy for spreading ketchup onto bread for 40 episodes with an episode length of 10 time steps. At a frequency of 1Hz, each episode is 3.5 minutes long (including ketchup refills and manual bread resets), amounting to ∼3 hours of learning experience. Following this, we test *zero-shot* generalization performance to new objects and substances, e.g. spreading ranch onto cheese. This training protocol is applied consistently across our method and all baselines to ensure a fair comparison.

How well does SPARTA perform complex object transformations? Fig. 3 (right table) reports the spreading task result, for seen objects (left) and zero-shot generalization to unseen objects and coating substances (right). SPARTA achieves an average of 53% coverage across all objects, significantly outperforming all baselines—absolute gain of 17% versus the best baseline, on average—while showing strong generalization to unseen objects. SPARSE yields negligible performance, confirming the absence of dense feedback hinders effective exploration. State-of-the-art LIV [10] improves greatly over SPARSE, but plateaus early due (see Appendix) to its inability to model fine-grained spatial transformations. Interestingly, the exploratory RANDOM action policy is a healthy baseline; its advantage over SPARSE accentuates how in settings with unstable or weak rewards, RL policy training can lead to degenerate solutions over time due to lack of entropy.

How sample-efficient is learning with SPOC affordances? Fig. 3 (left) presents learning curves for all methods on the bread spreading policy. SPARTA exhibits steep, monotonic improvement from the very first few episodes, often achieving usable performance (>60% coverage) within the first 90 minutes of training. This highlights the benefit of dense progress signals provided by our SPOC-based reward. In contrast, both SPARSE and LIV [10] remain flat throughout training seeing

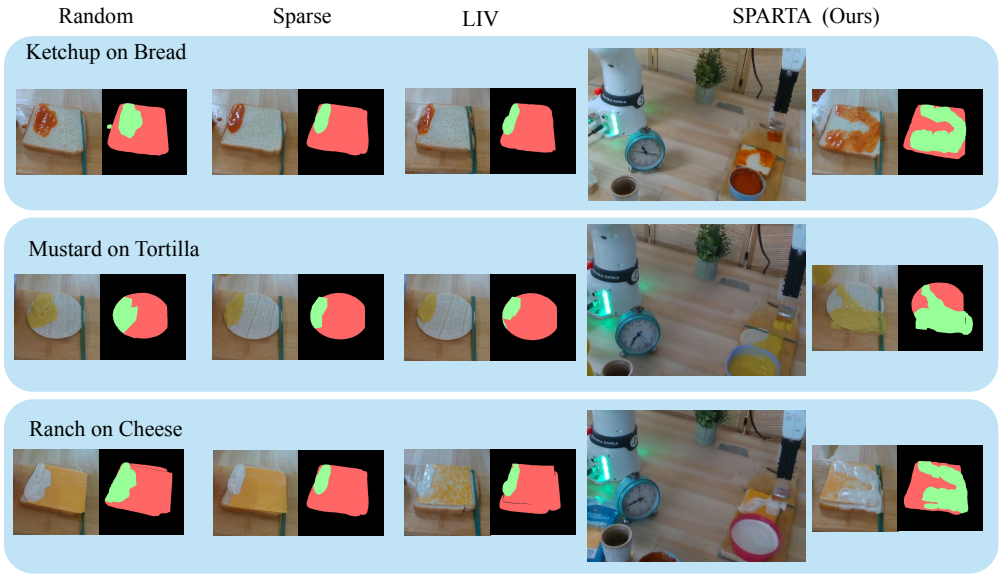


Figure 5: SPARTA's final SPOC [9] affordance maps shown against those of the RANDOM and SPARSE baselines for *spreading*. SPARTA significantly outmatches the baselines vis-a-vis substantially transforming actionable regions across different object-substance combinations.

no actionable feedback for policy improvement. Interestingly, the affordance prior seems to serve as an implicit curriculum: early actions focus on small object patches, while later episodes cover larger surface areas, learning interesting behaviors like reversing direction upon encountering object edges.

Does SPARTA generalize to new objects? Next we take policies trained on spreading ketchup onto bread and test them on held-out, unseen objects. See Fig. 3 (right side of table) and Fig. 5. SPARTA achieves strong generalization, substantially outperforming all baselines. The SPOC visual affordances encode meaningful spatial priors that transfer across objects with different shapes, textures, and appearances.

What does the reward curve reveal? Fig. 4 (left) displays reward progression over time for SPARTA. SPARTA produces *smooth, monotonic* reward curves, reflecting consistent progress as more of the object is transformed (enabled by our SPOC-based reward), which provides dense, step-wise feedback tied directly to visual state change. This elucidates SPARTA's advantage of interpretable, progress-aware reward signals that drive efficient and consistent learning in visually complex manipulation tasks.

Can SPARTA enable policy transfer across *tasks*? Finally, we take the object transformation challenge a notch higher by testing the *task* transfer ability of the trained robot policy. To this end, we test zero-shot generalization from *spreading* to *smashing*. Initial results (Fig. 4 (right)) show SPARTA's remarkable ability to transfer its learnt spreading behavior on a completely novel object transformation task zero-shot. This is preliminary evidence supporting another key advantage of our approach—once the robot learns effective behaviors for one spatial task, it can transfer its knowledge to other challenging progress-based tasks if the visual representations are shared.

5 Conclusion

We explored robot learning of spatially-progressing object state change manipulations, showing how visual affordances learned from video allow efficient RL policy learning on a real robot—without simulation or human demonstration. Our results offer a promising path towards addressing a full family of manipulation tasks in a holistic manner, by abstracting away their specific appearance to zero in on visual signals of transformed and actionable regions.

6 Limitations and Future Work

While SPARTA demonstrates encouraging performance, several limitations open up directions for future research.

- First, our approach depends on SPOC affordance maps, which can occasionally exhibit noise or tracking inconsistencies—especially during fine-grained transitions. Nonetheless, we do observe some policy robustness to those errors, due to repeated exposure to accurate predictions and the dense reward formulation, which allows learning to proceed even when intermediate frames are noisy, as long as progress is eventually captured. Future work can explore vision segmentation model enhancements.
- Second, we constrain the action space to planar (xy) motion to simplify learning and focus on visual reasoning. Extending to full 3D manipulation, including dynamic control of vertical motion and wrist orientation, would allow more expressive tool use and support tasks requiring vertical slicing, flipping, or angled spreading.
- Third, we fix the tool per task to isolate the challenge of visual transformation reasoning. Generalizing across tool types (e.g., different mashers or brushes) would require incorporating tool geometry into the observation space or enabling tool-conditioned policies.
- Fourth, we observe that while policies transfer well across similarly shaped objects (e.g., from bread to cheese slices), generalization to significantly different geometries (e.g., round tortillas) can result in over-extension beyond the object. Incorporating shape diversity into training, such as through object-aware replay or simulated augmentations, may help improve spatial generalization.
- Finally, to mitigate occlusions during interaction, we currently rely on capturing images when the robot lifts the end-effector between actions. Continuous visual feedback during contact remains challenging due to occlusion and deformation, and developing methods for affordance-aware perception under occlusion is a promising avenue for future work.

Acknowledgments

We would like to thank Luca Macesanu for designing the CAD models for the tool attachments used in our experiments.

References

[1] M. Shridhar, L. Manuelli, and D. Fox. Cliport: What and where pathways for robotic manipulation. In *Proceedings of the 5th Conference on Robot Learning (CoRL)*, 2021.

[2] L. Shao, T. Migimatsu, Q. Zhang, K. Yang, and J. Bohg. Concept2robot: Learning manipulation concepts from instructions and human demonstrations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2020.

[3] A. Bahety, P. Mandikal, B. Abbatematteo, and R. Martín-Martín. Screwmimic: Bimanual imitation from human videos with screw space projection. In *Robotics: Science and Systems (RSS), 2024*, 2024.

[4] L. Pinto and A. Gupta. Learning to push by grasping: Using multiple tasks for effective learning. In *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017.

[5] P. Sharma, L. Mohan, L. Pinto, and A. Gupta. Multiple interactions made easy (mime): Large scale demonstrations data for imitation. In *Conference on robot learning*, pages 906–915. PMLR, 2018.

[6] T. Souček, J.-B. Alayrac, A. Miech, I. Laptev, and J. Sivic. Look for the change: Learning object states and state-modifying actions from untrimmed web videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[7] Z. Xue, K. Ashutosh, and K. Grauman. Learning object state changes in videos: An open-world perspective. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[8] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022.

[9] P. Mandikal, T. Nagarajan, A. Stoken, Z. Xue, and K. Grauman. Spoc: Spatially-progressing object state change segmentation in video. In *ArXiv*, 2025.

[10] Y. J. Ma, W. Liang, V. Som, V. Kumar, A. Zhang, O. Bastani, and D. Jayaraman. Liv: Language-image representations and rewards for robotic control. *arXiv preprint arXiv:2306.00958*, 2023.

[11] E. Heiden, M. Macklin, Y. Narang, D. Fox, A. Garg, and F. Ramos. Disect: A differentiable simulation engine for autonomous robotic cutting. *Robotics: Science and Systems (RSS)*, 2021.

[12] Z. Xu, Z. Xian, X. Lin, C. Chi, Z. Huang, C. Gan, and S. Song. Roboninja: Learning an adaptive cutting policy for multi-material objects. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[13] C. C. Beltran-Hernandez, N. Erbetti, and M. Hamaya. Sliceit!–a dual simulator framework for learning robot food slicing. *International Conference on Robotics and Automation (ICRA)*, 2024.

[14] H. Shi, H. Xu, S. Clarke, Y. Li, and J. Wu. Robocook: Long-horizon elasto-plastic object manipulation with diverse tools. *arXiv preprint arXiv:2306.14447*, 2023.

[15] R. Ye, Y. Hu, Y. A. Bian, L. Kulm, and T. Bhattacharjee. Morpheus: a multimodal one-armed robot-assisted peeling system with human users in-the-loop. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[16] T. Chen, E. Cousineau, N. Kuppuswamy, and P. Agrawal. Vegetable peeling: A case study in constrained dexterous manipulation. *arXiv preprint arXiv:2407.07884*, 2024.

[17] C. Dong, L. Yu, M. Takizawa, S. Kudoh, and T. Suehiro. Food peeling method for dual-arm cooking robot. In *IEEE/SICE International Symposium on System Integration (SII)*, 2021.

[18] J. Liu, Y. Chen, Z. Dong, S. Wang, S. Calinon, M. Li, and F. Chen. Robot cooking with stir-fry: Bimanual non-prehensile manipulation of semi-fluid objects. *IEEE Robotics and Automation Letters*, 2022.

[19] N. Kanazawa, K. Kawaharazuka, Y. Obinata, K. Okada, and M. I. and. Real-world cooking robot system from recipes based on food state recognition using foundation models and pddl. *Advanced Robotics*, 2024.

[20] L. Yen-Chen, A. Zeng, S. Song, P. Isola, and T.-Y. Lin. Learning to see before learning to act: Visual pre-training for manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[21] R. Shah and V. Kumar. Rrl: Resnet as representation for reinforcement learning. *International Conference on Machine Learning (ICML)*, 2021.

[22] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3m: A universal visual representation for robot manipulation. *Conference on Robot Learning (CoRL)*, 2022.

[23] I. Radosavovic, T. Xiao, S. James, P. Abbeel, J. Malik, and T. Darrell. Real-world robot learning with masked visual pre-training. *Conference on Robot Learning (CoRL)*, 2022.

[24] Y. J. Ma, S. Sodhani, D. Jayaraman, O. Bastani, V. Kumar, and A. Zhang. VIP: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2022.

[25] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. Scaling egocentric vision: The epic-kitchens dataset. In *European Conference on Computer Vision (ECCV)*, 2018.

[26] Y. J. Ma, W. Liang, G. Wang, D.-A. Huang, O. Bastani, D. Jayaraman, Y. Zhu, L. Fan, and A. Anandkumar. Eureka: Human-level reward design via coding large language models. *ICLR*, 2024.

[27] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, A. Herzog, D. Ho, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, E. Jang, R. J. Ruano, K. Jeffrey, S. Jesmonth, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, K.-H. Lee, S. Levine, Y. Lu, L. Luu, C. Parada, P. Pastor, J. Quiambao, K. Rao, J. Rettinghouse, D. Reyes, P. Sermanet, N. Sievers, C. Tan, A. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, S. Xu, M. Yan, and A. Zeng. Do as i can and not as i say: Grounding language in robotic affordances. In *CoRL*, 2022.

[28] 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. In *CoRL*, 2023.

[29] S. Brahmbhatt, A. Handa, J. Hays, and D. Fox. Contactgrasp: Functional multi-finger grasp synthesis from contact. In *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019.

[30] P. Mandikal and K. Grauman. Dexterous robotic grasping with object-centric visual affordances. In *International Conference on Robotics and Automation (ICRA)*, 2021.

[31] P. Mandikal and K. Grauman. Dexvip: Learning dexterous grasping with human hand pose priors from video. In *Conference on Robot Learning (CoRL)*, 2021.

[32] Y. Wu, J. Wang, and X. Wang. Learning generalizable dexterous manipulation from human grasp affordance. In *Conference on Robot Learning (CoRL)*, 2022.

[33] A. Agarwal, S. Uppal, K. Shaw, and D. Pathak. Dexterous functional grasping. In *Conference on Robot Learning*. PMLR, 2023.

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

[35] Y. Hasson, G. Varol, D. Tzionas, I. Kalevatykh, M. J. Black, I. Laptev, and C. Schmid. Learning joint reconstruction of hands and manipulated objects. In *CVPR*, 2019.

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

[37] Y. Ye, X. Li, A. Gupta, S. D. Mello, S. Birchfield, J. Song, S. Tulsiani, and S. Liu. Affordance diffusion: Synthesizing hand-object interactions. In *CVPR*, 2023.

[38] N. Aboubakr, J. L. Crowley, and R. Ronfard. Recognizing manipulation actions from state-transformations. *arXiv preprint arXiv:1906.05147*, 2019.

[39] S. Liu, S. Tripathi, S. Majumdar, and X. Wang. Joint hand motion and interaction hotspots prediction from egocentric videos. In *CVPR*, 2022.

[40] J. Yu, X. Li, X. Zhao, H. Zhang, and Y.-X. Wang. Video state-changing object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20439–20448, 2023.

[41] P. Tokmakov, J. Li, and A. Gaidon. Breaking the "object" in video object segmentation. In *CVPR*, 2023.

[42] T. Souček, D. Damen, M. Wray, I. Laptev, and J. Sivic. Genhowto: Learning to generate actions and state transformations from instructional videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024.

[43] A. Miech, D. Zhukov, J.-B. Alayrac, M. Tapaswi, I. Laptev, and J. Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2630–2640, 2019.

[44] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, Z. Zeng, H. Zhang, F. Li, J. Yang, H. Li, Q. Jiang, and L. Zhang. Grounded sam: Assembling open-world models for diverse visual tasks, 2024.

[45] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[46] OpenAI. Gpt-4 technical report. Technical report, OpenAI, 2023. Submitted on 15 Mar 2023, last revised 27 Mar 2023.

[47] J. Luo, Z. Hu, C. Xu, Y. L. Tan, J. Berg, A. Sharma, S. Schaal, C. Finn, A. Gupta, and S. Levine. Serl: A software suite for sample-efficient robotic reinforcement learning, 2024.

[48] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning (ICML)*, 2018.

[49] P. J. Ball, L. Smith, I. Kostrikov, and S. Levine. Efficient online reinforcement learning with offline data. In *International Conference on Machine Learning (ICML)*, 2023.

[50] R. Martín-Martín, M. A. Lee, R. Gardner, S. Savarese, J. Bohg, and A. Garg. Variable impedance control in end-effector space: An action space for reinforcement learning in contact-rich tasks. In *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 1010–1017. IEEE, 2019.

[51] A. Allshire, R. Martín-Martín, C. Lin, S. Manuel, S. Savarese, and A. Garg. Laser: Learning a latent action space for efficient reinforcement learning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6650–6656. IEEE, 2021.

[52] M. Bogdanovic, M. Khadiv, and L. Righetti. Learning variable impedance control for contact sensitive tasks. *IEEE Robotics and Automation Letters*, 5(4):6129–6136, 2020.

[53] H. Zhu, J. Yu, A. Gupta, D. Shah, K. Hartikainen, A. Singh, V. Kumar, and S. Levine. The ingredients of real-world robotic reinforcement learning. *ICLR*, 2020.

[54] 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. *European Conference on Computer Vision (ECCV)*, 2024.

[55] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. Segment anything. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[56] Z. Yang and Y. Yang. Decoupling features in hierarchical propagation for video object segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

A Cross-task generalization to smashing

Model	*Seen* Clay Avocado	*Unseen* Banana
RANDOM	0.62	0.54
SPARSE	0.42	0.27
LIV [10]	0.48	0.42
SPARTA (Ours)	**0.79**	**0.67**

Table 1: **Cross-task generalization results for the smashing task.** We report transformation coverage on the proxy clay avocado (seen during fine-tuning) and a real banana (unseen). SPARTA achieves the highest performance across all settings, showing strong generalization from spreading to smashing. Fine-tuned policies transfer effectively to real objects, outperforming both sparse and goal-conditioned baselines. Metric is transformation coverage (%).

We evaluate SPARTA on a challenging cross-task generalization setting, transferring a spreading policy to a visually and mechanically distinct smashing task. Given that the masher tool covers a larger area in a single motion, we reduce the episode length to 5 time steps.

To facilitate transfer, we initialize the smashing policy by fine-tuning a pretrained spreading policy for 10 episodes on a proxy object—molded clay shaped like an avocado. This finetuned policy is then evaluated zero-shot on a real banana. As shown in Table 1, SPARTA significantly outperforms all baselines, achieving up to a 25% gain in transformation coverage. Qualitative results are shown in Fig. 6.

We also conduct an ablation comparing this finetuned policy to the zero-shot spreading policy (i.e., directly applying the spreading policy to smashing without adaptation). Sample rollouts are shown in Fig. 7. We observe that zero-shot behavior often fails to transform the object effectively. In particular, the spreading policy tends to move the end-effector directly to the object center, where resistance is highest. In contrast, effective smashing requires initiating contact from the object's edge and progressively advancing inward to ensure full deformation. During fine-tuning, the robot learns this behavior, adapting its motion strategy to maximize transformation coverage—demonstrating the benefit of short, task-specific adaptation for policy transfer across manipulation modes.

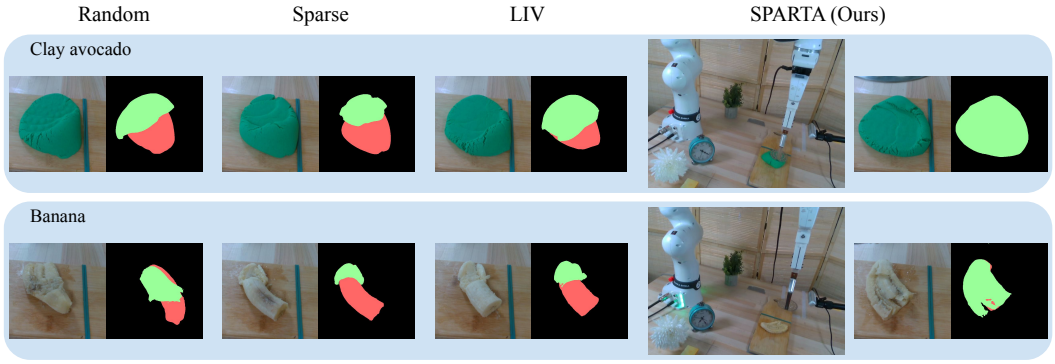


Figure 6: **Smashing results.** We finetune the spreading policy for 10 episodes on molding clay shaped as an avocado. This is then tested zero-shot on a real banana. SPARTA is able to effectively transform larger areas of the object in comparison to the baselines. Moreover, the smashed regions are also more visually distinct in texture indicating a successful state change transformation. Full rollouts are provided in the supplementary video.

B Reward curve analysis for SPARTA and LIV

In Fig.8, we present additional reward curves comparing SPARTA to LIV[10], a state-of-the-art goal-conditioned method that produces dense visual rewards. While Fig. 4 in the main paper illus-

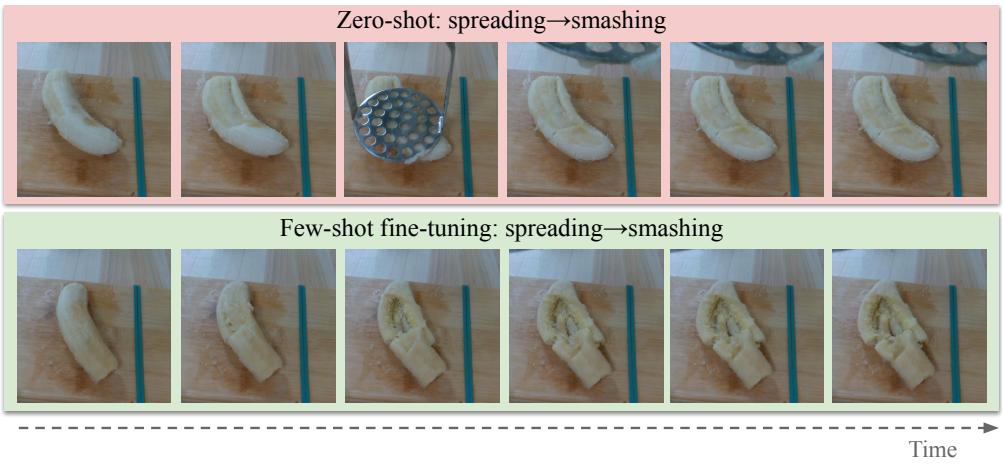


Figure 7: **Effectiveness of few-shot fine-tuning for cross-task transfer.** *Top:* Applying the spreading policy zero-shot to the smashing task results in sub-optimal behavior—often triggering premature force thresholds and getting stuck mid-motion (center frame), while failing to fully transform the object. *Bottom:* Fine-tuning the spreading policy for just 10 episodes ($\sim$ 20 minutes) yields a robust smashing policy that produces clear visual and textural changes in the object, indicating successful transformation.

trates a single rollout, here we show multiple representative examples to analyze how each method guides learning throughout an episode.

SPARTA consistently produces *smooth and monotonic* reward curves, reflecting steady object transformation over time. This behavior emerges from our SPOC-based reward, which offers dense, spatially grounded feedback directly tied to visual state change—encouraging consistent policy improvement from the very first step.

In contrast, LIV exhibits flat or erratic reward signals during much of the episode, despite visible progress in the task. Since LIV relies on goal-conditioned language embeddings, it often fails to reflect intermediate object states, producing rewards only when the object is nearly fully transformed. This results in weak guidance during early and mid-stage actions, hindering learning. These results highlight a key limitation of goal-conditioned embeddings like LIV: they struggle to capture fine-grained, spatially distributed progress in visually complex tasks. In contrast, SPARTA's progress-aware rewards drive more consistent, sample-efficient learning—especially in manipulation tasks that unfold gradually over space and time.

C SPOC Map Generation

While prior work [9] leverages Grounded-SAM [44] and CLIP [45] for generating object state change affordance maps, we find that replacing CLIP with a vision-language model (VLM) such as GPT-4o [46] significantly improves segmentation accuracy—particularly for distinguishing intra-object regions undergoing transformation. To enable real-time robot training, we further introduce an efficient mask-tracking strategy that propagates SPOC segmentations across frames, reducing VLM overhead while preserving spatial consistency.

In the original SPOC formulation [9], actionable and transformed masks are obtained by first detecting object instances with Grounded-SAM and then assigning a binary label (e.g.,"coated" vs. "uncoated") based on CLIP similarity scores with text prompts. While this approach works when multiple objects in different states are present (e.g., four bread slices with varying coating), it fails in the common case where transformation occurs *within* a single object. For example, if only part of a bread slice is coated, Grounded-SAM outputs a single object mask that spans both transformed and

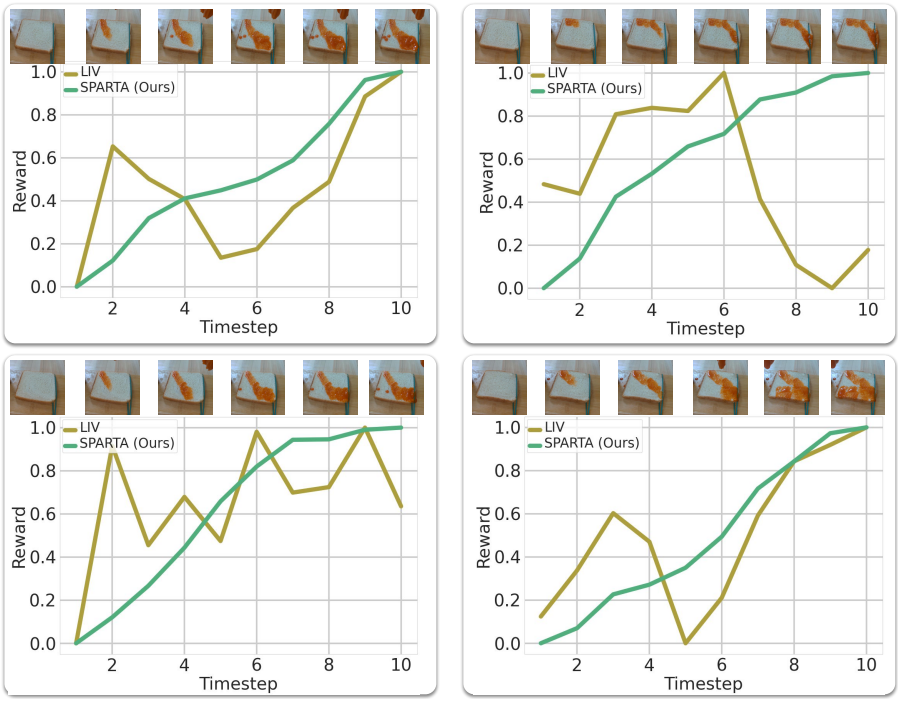


Figure 8: **Cumulative reward curve analysis for SPARTA and LIV.** Comparison of cumulative dense rewards during sample rollouts of a spreading task (ketchup on bread). SPARTA produces smooth, incremental rewards aligned with visual progress. In contrast, LIV rewards remain unstable throughout the episode, offering poor guidance.

untransformed regions. Since CLIP assigns a single label to this entire mask, intra-object progress is lost—undermining the dense, spatially evolving feedback needed for effective RL.

To address this, we refine the pipeline for generating SPOC affordance maps, shown in Fig. 9. First, we obtain the full object mask using Grounded-SAM (Fig.9a). We then apply farthest-point sampling to generate a grid of points over the object surface and prompt SAM with each point to produce intra-object mask proposals. Each mask is then visually overlaid with a distinct boundary color and passed to GPT-4o along with a prompt asking whether the highlighted region reflects the object's initial state (e.g., "uncoated bread") or transformed state (e.g., "coated bread"). This results in fine-grained actionable and transformed sub-regions (Fig. 9b), even within a single object.

Since querying GPT-4o takes roughly 5 seconds per frame, applying it to every frame would bottleneck training. To overcome this, we incorporate a fast mask propagation strategy using DeAOT [56] to track labeled regions across subsequent frames (Fig. 9c). This reduces runtime to 0.2 seconds per frame and maintains temporal consistency with high accuracy.

This pipeline allows us to generate high-quality, real-time SPOC affordance maps, enabling dense reward signals and sample-efficient learning in the real world.

D Implementation details

In addition to the implementation details provided in the main paper (L204–213, L240–245), we include further specifics on RL policy training.

We use Soft Actor-Critic (SAC) as our RL algorithm, chosen for its strong sample efficiency in real-world online RL settings [47, 49]. Our implementation builds on the publicly available codebase from SERL [47], in which the actor and critic are trained asynchronously in parallel processes. As the robot interacts with the environment and populates the replay buffer, the learner continuously

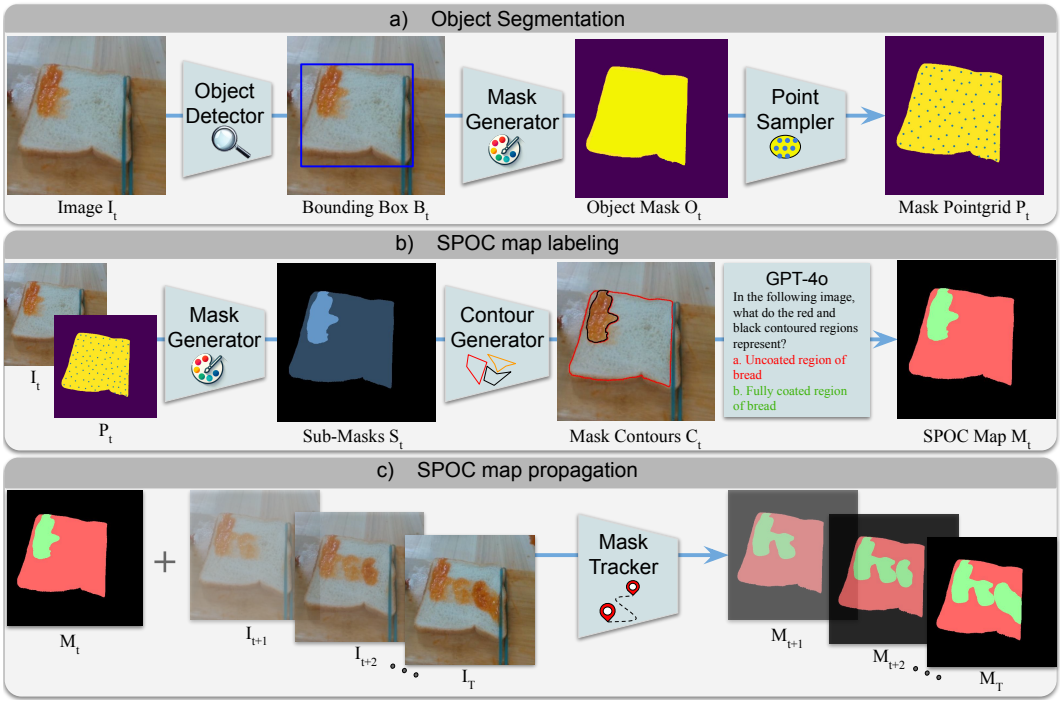


Figure 9: **SPOC affordance map generation pipeline. (a)** Grounded-SAM is used to extract a full-object mask from the initial frame. **(b)** Farthest-point sampling generates intra-object regions, which are classified into *actionable* or *transformed* by prompting GPT-4o using color-coded overlays. **(c)** Once classified, transformed regions are tracked across subsequent frames using DeAOT to maintain temporal consistency with minimal computation.

updates both the policy and critic networks. We found the actor-to-critic update ratio to be a key factor in policy performance. Empirically, an update ratio of 1:10 yielded the best results, striking a balance between policy improvement and stable value estimation. Both the actor and critic networks use a learning rate of 3e-4, with a linear warmup over the first 2000 training steps. We use a discount factor $\gamma = 0.95$. To improve generalization and robustness, we apply random image cropping as a data augmentation step prior to visual encoding. The vision encoder is a 10-layer ResNet (ResNet-10), initialized with ImageNet-pretrained weights. All training and policy execution is performed in real time on a single NVIDIA RTX 3090 Ti GPU.

D.1 Task-specific choices

We adopt a few task-specific design choices to ensure stable and effective policy learning, while keeping the core action space abstract and general across tasks.

Episode Length. We set episode lengths to match the expected duration required to complete each task. For spreading, we use a length of 10 steps, while for smashing, we use 5 steps—reflecting the larger area covered per action due to the wider base of the masher compared to the brush. In all cases, episodes begin from a fixed corner position in the workspace for consistency.

Action Execution. The robot predicts continuous Δx, Δy displacements of the end-effector, which allows generalization across tasks. However, the execution strategy differs slightly by task:

For *spreading*, the robot moves along the object surface in-plane from a fixed corner.

For *smashing*, the robot starts elevated, moves laterally in the air, then lowers to press into the object with a downward motion before lifting back up.

To avoid occlusion during visual observation, the robot lifts above the workspace after each action in both tasks before capturing the next input.

Brush Refilling. In spreading, we include an automatic routine to simulate refilling the brush with coating every 2 steps. This ensures that the brush remains "loaded" throughout the episode. An interesting future extension would be to make refilling a learnable action, allowing the robot to decide when to replenish material based on task progress.