Mash, Spread, Slice! Learning to Manipulate Object States via Visual 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: Most robot manipulation focuses on changing the *kinematic state* of objects: picking, placing, opening, or rotating them. However, a wide range of real-world manipulation tasks involve a different class of *object state change*—such as mashing, spreading, or slicing—where the object's physical and visual state evolve progressively without necessarily changing its position. We present SPARTA, the first unified framework for the family of object state change manipulation tasks. Our key insight is that these tasks share a common structural pattern: they involve spatially-progressing, object-centric changes that can be represented as regions transitioning from an *actionable* to a *transformed* state. Building on this insight, SPARTA integrates *spatially progressing object change segmentation maps*, a visual skill to perceive actionable vs. transformed regions for specific object state change tasks, to generate a) structured policy observations that strip away appearance variability, and b) dense rewards that capture incremental progress over time. These are leveraged in two SPARTA policy variants: reinforcement learning for fine-grained control without demonstrations or simulation; and greedy control for fast, lightweight deployment. We validate SPARTA on a real robot for three challenging tasks across 10 diverse real-world objects, achieving significant improvements in training time and accuracy over sparse rewards and visual goal-conditioned baselines. Our results highlight progress-aware visual representations as a versatile foundation for the broader family of object state manipulation tasks. More information at https://vision.cs.utexas.edu/projects/sparta-robot

1 Introduction

The dominant paradigm in robotic manipulation centers on tasks involving rigid body motion—such as picking-and-placing [1], opening and closing [2, 3], pushing [4], or rotating [5] objects. While these tasks are foundational, they largely entail changing the kinematic state of objects where progress on the task is readily visible and easily monitored via changes in object pose. However, many real-world scenarios involve a fundamentally different class of manipulations: *object state changes* (OSC) [1] [6, 7, 10]—where an object's physical state and visual appearance is progressively transformed, without necessarily altering its pose (see Fig. 1, top). Everyday examples abound: *mashing* a banana into purée, *spreading* jam on bread, or *slicing* a cucumber. These tasks demand continuous interaction that alters the object's shape, texture, and color, making them both mechanically challenging and visually complex. Such state changes are ubiquitous in everyday activities—from cooking (e.g., grating, peeling, shredding) to household chores (e.g., painting, wiping, ironing)—yet remain largely underexplored in robotics.

What makes OSC manipulation challenging for robotics? Unlike motion-centric tasks, OSC requires continuous reasoning about where transformations have already occurred *within* a possibly non-rigid

[1] Here we adopt the term "object state change" (OSC) from the vision literature [6–8]: 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. Not to be confused with Operational Space Control [9].

object, where they are still needed, and how to act accordingly. Two key obstacles arise. First, at the *representation level*, raw RGB observations entangle appearance with object state, obscuring the signals of progress and hindering generalization across objects. Second, at the *learning level*, obtaining a good reward function is challenging: sparse success rewards provide little guidance for exploration [11], while goal-conditioned reward functions (e.g. LIV [12]) often rely on global scene-level embeddings that miss the *fine-grained, incremental progress* essential for OSC. Together, these limitations make current approaches sample-inefficient and ill-suited to tasks where state changes unfold dynamically within the object.

To tackle these challenges, we propose SPARTA (Spatial Progress-Aware Robotic object TransformAtion)—a robotic system that introduces *structured, progress-aware visual affordances* tailored to OSC manipulation (see Fig. 1, bottom). SPARTA builds on recent computer vision advances in detecting spatially-progressing object state changes (SPOC [8]), integrating them into a fully autonomous robotic system. SPOC segments a transforming object into two regions: *actionable* and *transformed*. For instance, in mashing a potato, unmashed chunks are actionable, while mashed portions are transformed. SPARTA leverages SPOC affordance[2] maps in two crucial ways: (1) as structured visual observations that strip away appearance detail while preserving progress cues, enabling generalization across objects; and (2) as dense, spatially grounded reward signals that quantify incremental progress at each step. By explicitly representing "what has changed" and "what remains," SPARTA equips robots to reason about state progression rather than mere object kinematics.

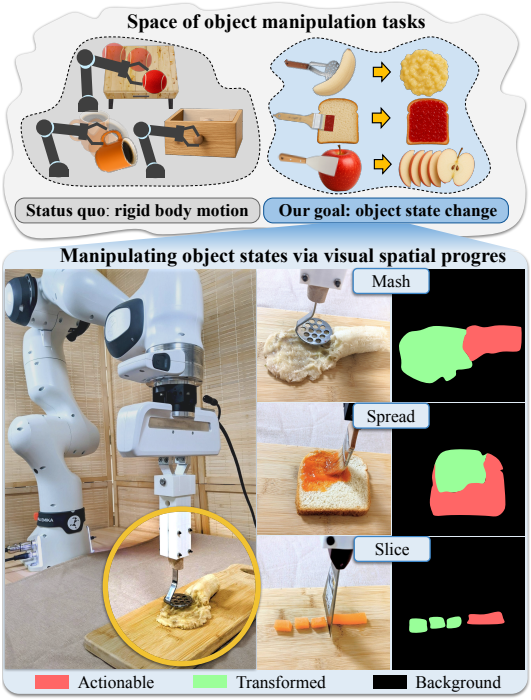


Figure 1: **Top:** While most robotic manipulation focuses on rigid-body motion, many real-world tasks involve *object state changes* such as mashing, spreading, or slicing, where objects are progressively transformed. **Bottom:** SPARTA leverages spatially-progressing affordance maps of *actionable* vs. *transformed* regions, successfully demonstrating how to guide real robot manipulation for this family of tasks.

Our formulation enables two policy variants within the same framework. In SPARTA-L, we use SPOC-derived rewards to train real-world RL agents from scratch—without demonstrations or simulation—achieving *highly* sample-efficient learning. In parallel, SPARTA-G offers a non-parametric alternative, greedily steering toward nearby actionable regions in the SPOC map. Hence, this unified framework supports both (1) reinforcement learning, for robust, fine-grained control in settings where noise and uncertainty demand adaptive strategies; and (2) greedy control, for fast, lightweight deployment in simpler settings without training. Together, SPARTA's two policy variants demonstrate the versatility of its progress-aware affordances: a single representation can power both quick heuristic controllers and data-driven RL agents, depending on task complexity.

In our experiments, we show that with just 1.5–3 hours of online RL training *directly in the real world* and *no human demonstrations*, SPARTA learns policies that reliably induce object state change. We evaluate across three representative OSC tasks—spreading, mashing, and slicing—on 10 diverse real-world objects, demonstrating both robustness and generality. By contrast, baseline methods fail to learn meaningful behavior, highlighting that dense, interpretable affordances for object state change are key to enabling sample-efficient, generalizable real-world robot learning—charting a path beyond rigid-body manipulation.

[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 [13–16], peeling [17–19], and stir-frying [20]. However, these efforts tackle each task in *isolation*, 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, exploiting a *unified* visual representation that is shared across objects and state-change tasks.

Visual representations for robot learning. To accelerate downstream policy learning, recent works pretrain visual representations on large-scale data [12, 21–23]. More relevant to our novel visual rewards, VIP [23] learns an implicit value function over egocentric videos, while its extension LIV [12] further incorporates language-goal embeddings. There is also growing interest in LLMs [24] & VLMs [25] for robotic reasoning, typically using 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 supporting both efficient planning and 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 [26–31]. Parallel efforts in computer vision predict hand-object interactions [32–34], but they emphasize pick-and-place or grasping 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 achieving non-rigid object interactions on a real robot.

Object state change understanding. OSC is explored in computer vision for video-level classification [6, 7], segmentation [8, 35, 36], and generation [37]. Our work is inspired in part by the *spatially progressing object state change* (SPOC) task [8], which segments state-changing objects into actionable and transformed regions. Trained on large-scale instructional "how-to" videos [38], SPOC exhibits robust spatial reasoning across diverse objects and transformations. However, these models are vision-only: they passively analyze state changes but do not 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 affordances capturing gradual visual progress—difficult to address with tactile sensing [19], force models, or binary state classifiers [2, 39].

Real-world Reinforcement Learning Real-world RL enables autonomous policy learning directly from real-world interactions, avoiding the need for explicit world models or high-fidelity simulators. This makes it particularly promising for contact-rich manipulations, where accurate modeling is notoriously difficult [40, 41]. However, when tasks require progressive object state changes, existing methods struggle on two fronts: first, learning visual representations that capture subtle intra-object changes; and second, defining reward functions that provide dense, informative feedback [11, 42]. These challenges lead to poor sample efficiency and hinder real-world applicability. Our work tackles both issues by leveraging spatially progressing OSC segmentation maps, leading to successful policy learning on challenging tasks.

3 Robotic Object State Change

Our goal is to enable robots to perform *object state change* (OSC) tasks, where the object's morphology, texture, or appearance evolve progressively over time. Unlike traditional robotic manipulation, which focuses on moving rigid objects in space (e.g., pick-and-place or pushing), OSC tasks demand reasoning about transformations *within* the object itself. The challenge is not simply to change an object's kinematic state, but to decide where and how to act on deformable regions so as to drive continuous, irreversible changes in the object's physical state. This fundamentally alters the prob-

lem: the robot must perceive gradual transformations, localize actionable regions, and sequence fine-grained actions that accumulate toward a globally transformed outcome.

Problem Formulation. We formulate OSC task as a Partially Observable Markov Decision Process (POMDP) $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \Omega, r, \rho_0, \gamma)$, where $\mathcal{S}$ are the true environment states, $\mathcal{A}$ are robot actions, Ω are the observations, $\mathcal{T}(s_{t+1} \mid s_t, a_t)$ governs state evolution, $r(s_t, a_t)$ provides feedback, ρ_0 is the distribution over initial states, and γ is the discount factor. The goal is to learn a policy $\pi(a_t \mid \omega_{\leq t})$ that maximizes expected discounted return: $J(\pi) = \mathbb{E}_\pi\left[\sum_{t=0}^{T-1} \gamma^t r(s_t, a_t)\right]$. Here, partial observability arises because the underlying object state (e.g., which region of a banana is mashed) is not directly available—only visual observations and proprioception are accessible. Unlike motion-centric tasks where object poses provide a sufficient state proxy, in OSC we need observation spaces that faithfully approximate these hidden, spatially evolving states.

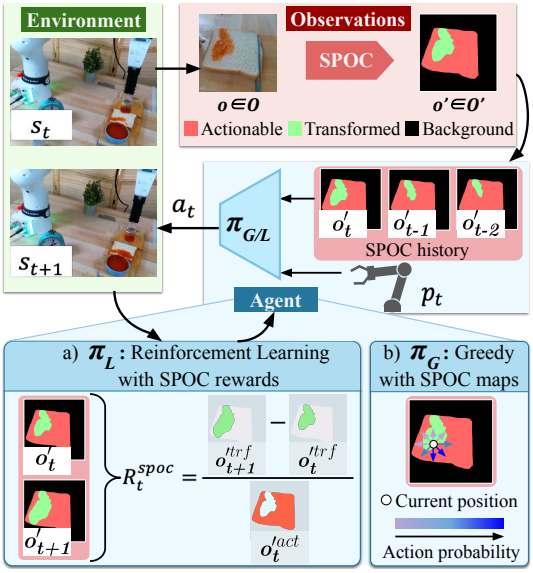


Figure 2: **Overview of SPARTA.** At each episode step, our policy takes the current and past SPOC [8] visual-affordance (segmentation) maps as inputs, along with the robot arm's proprioception data and predicts a displacement action for the arm's end-effector. SPARTA supports two robot policy variants: (a) **SPARTA-L** (Learning): a reinforcement learning agent trained using a dense reward that measures the progressive change of object regions from *actionable* (red) to *transformed* (green). (b) **SPARTA-G** (Greedy): selects among 8 discrete directions based on the local density of actionable pixels, producing a fast, greedy policy guided by visual progress.

Observation Space. The robot operates in a tabletop workspace with a single object presented at random orientations. Each observation $\omega_t \in \Omega$ is represented by visual and proprioceptive components, $\Omega = O \times P$. At time t, the robot receives an RGB observation $o_t \in O$ from a fixed camera and proprioceptive input $p_t \in P$ encoding end-effector position. The key challenge is that raw RGB frames, while visually rich, entangle object-specific appearance with the underlying dynamics of state change. For robots, this makes it difficult to learn sample-efficient, generalizable policies from limited real-world data. What is needed instead are structured visual abstractions that strip away appearance-specific detail while preserving cues that reflect how the object is evolving over time, bringing the observation space closer to the task-relevant state representation. We introduce such a representation in Sec. 4.

Action Space. While classical manipulation often requires planning global object motions, OSC tasks demand acting at *specific intra-object locations* to drive localized transformations (e.g., pressing unmashed potato chunks or brushing uncoated regions of bread). To reflect this, we constrain the action space to a 2D manifold aligned with the object surface, enabling policies to directly reason about *where* to apply tool actions. Concretely, the policy outputs continuous $\Delta x, \Delta y$ displacements, sampled from a Gaussian centered at the mean predicted by π. At the resulting (x, y) location, a task-specific primitive is executed: sweeping motions for spreading, downward pressing for mashing, or slicing strokes. This structured action space both mirrors the spatially progressive nature of OSC tasks and reduces complexity, making it possible to learn sample-efficient policies that generalize across objects.

Next we provide a detailed framework for SPARTA, leveraging visual spatial progress to solve robotic OSC tasks.

4 SPARTA: Robot Policies for OSCs via Visual Spatial Progress

4.1 Integrating SPOC Visual Affordances for Robotics

To provide structured visual abstractions for OSC manipulation, we adapt the *Spatially Progressing Object State Change* (SPOC) task [8], which segments objects into *actionable* and *transformed* regions (e.g., plain vs. coated bread). Given RGB frames $o_1, \ldots, o_T$, SPOC produces binary masks $o'_t = o'^{act}_t, o'^{trf}_t$ that serve as the robot's sole visual input, stripping away appearance variability and supplying interpretable, object-centric progress maps (Fig. 2). For real-time robot learning, we generate SPOC masks online using SAM [43] + GPT-4o [44] with DeAOT [45] propagation for real-time control (details in Appendix Sec. B). Crucially, SPOC affordances capture *what transformations look like* from large-scale human vision data without assuming embodiment, while binary actionable/transformed masks replace raw RGB, enabling generalization across novel objects and materials. SPARTA exploits SPOC affordances through two variants: SPARTA-L (4.2), which uses SPOC rewards for real-world online RL, and SPARTA-G (4.3), which greedily selects actions from SPOC maps. A shared MDP formulation with SPOC-based states and rewards enables both adaptive learning and reactive planning within a unified framework.

4.2 SPARTA-L: Reinforcement Learning with SPOC rewards

Object state change tasks demand *sequential decision-making*: each action transforms only a local region of the object, and the robot must continually decide *where to act next* to optimize long-term task success. Reinforcement learning (RL) naturally fits this setting, as it optimizes long-horizon returns rather than immediate feedback.

A central challenge in real-world RL, however, is reward design. Sparse, binary success signals provide little guidance for sample-efficient training, while dense, automated feedback is rarely available [11]. For instance, coating an additional patch of bread or mashing a new section of banana is meaningful progress toward the goal, but this nuance is lost with simple binary rewards. As we show in Sec. 5, such dense feedback is essential for stabilizing exploration and driving sample-efficient learning in the real world.

To address this, we design a dense, spatially grounded reward function that reflects incremental progress and enables *real-time, demonstration-free* learning:

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

Here, R_t^{succ} is a sparse terminal reward for task completion (when $>95\%$ of the object is transformed), and $R_t^{entropy}$ promotes action diversity and exploration. The key novel component, R_t^{spoc} (see Fig. 2-a), provides dense feedback at every step by quantifying *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 transformed and actionable areas of SPOC segmentation maps o'^{trf}_t and o'^{act}_t. Unlike sparse success rewards, this formulation rewards incremental transformation of new regions, focusing the agent on actionable areas while avoiding redundancy.

The result is an object-centric, task-agnostic reward that captures fine-grained spatial progress from vision alone. It eliminates the need for simulation, privileged state, or human demonstrations, and—as shown in Sec. 5—yields smooth, monotonic learning curves for real-world OSC tasks. For policy optimization, we build on SERL [42], using Soft Actor-Critic (SAC) [46] with regularization from RLPD [47] for sample-efficient off-policy learning directly in the real world. Unlike SERL, however, SPARTA does not rely on any human demonstrations. Instead, visually grounded rewards derived from SPOC affordances are sufficient to drive sample-efficient real world RL.

4.3 SPARTA-G: Greedy Policy with SPOC Maps

While RL provides a general framework for policy learning under noisy perception, some OSC tasks admit simpler solutions. With large, symmetric tools (e.g., a masher), each action covers broad areas, making control less sensitive to perceptual errors or misalignment. In such cases, a greedy policy that moves toward untransformed regions can suffice. By contrast, tasks with thin, asymmetric tools (e.g., spreading with a brush) demand precise, noise-robust control, where RL offers a clear advantage.

To capture the easier regime, we introduce SPARTA-G, a non-parametric greedy controller that exploits spatial priors in SPOC maps. At timestep t, given a segmentation $o't \in \mathcal{O}'$ labeling pixels as *actionable*, *transformed*, or *background*, and the end-effector position p_t, the controller evaluates 8 candidate motions $a_t^{(i)} i = 1^8$ in the xy-plane (see Fig. 2-b). For each, it computes the density of actionable pixels in a neighborhood $\mathcal{N}(p_t + a_t^{(i)})$ and selects the best direction:

$$a_t = \arg\max_{a_t^{(i)}} \sum_{x \in \mathcal{N}(p_t + a_t^{(i)})} \mathbb{I}[o'(x) = \texttt{actionable}], \tag{3}$$

where $\mathbb{I}[\cdot]$ is the indicator. This steers the tool toward regions most likely to yield progress.

Though it requires no training, SPARTA-G still defines a policy: a deterministic mapping from SPOC-derived state features to actions. It is lightweight, fast to deploy, and effective for coarse transformations. As shown in Sec. 5, however, SPARTA-L achieves superior performance in tasks needing fine directional control. Together, the two variants demonstrate how spatial progress maps support both greedy control and reinforcement learning within one framework.

5 Experimental Evaluation

Manipulation tasks & objects. We evaluate three cooking-related OSC tasks—*spreading*, *mashing*, and *slicing*—each involving irreversible structural and appearance changes that challenge perception, affordance reasoning, and reward design. Experiments span 10 diverse objects with varied shapes, textures, and colors (Table 4a), testing both visual robustness and policy generalization.

Training and implementation details Experiments use a 7-DoF Franka Emika Panda with torque sensing, a 3-finger gripper, and a front camera providing RGB input. Tasks are structured into short episodes (10 steps for spreading, 5 for mashing/slicing) starting from a fixed workspace corner. SPARTA-L trains efficiently under real-world budgets—$\sim$3 hrs for spreading and $\sim$1.5 hrs for mashing/slicing—using clay proxies and a few greedy rollouts to simplify resets and stabilize exploration. Policies are learned with asynchronous SAC from SERL [42], and visual and proprioceptive inputs are encoded with lightweight networks. The same protocol is applied across baselines for fair comparison. Details in Appendix Sec. A.

Comparisons. We benchmark against three baselines:

(**1**) RANDOM: a control baseline with actions sampled uniformly randomly within the constrained action space, reflecting unstructured exploration with no task guidance.

(**2**) SPARSE: a sparse reward baseline using only a binary task completion reward, queried from GPT-4o via task-specific prompts on the final image (e.g., "Is the bread fully coated with ketchup?"). A "yes" ends the episode with +1 reward; otherwise, no reward is given. This mirrors our use of VLMs for SPOC mask generation.

(**3**) LIV [12]: a state-of-the-art goal-conditioned representation learning method trained on human activity videos [10, 48]. 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 bread with ketchup").

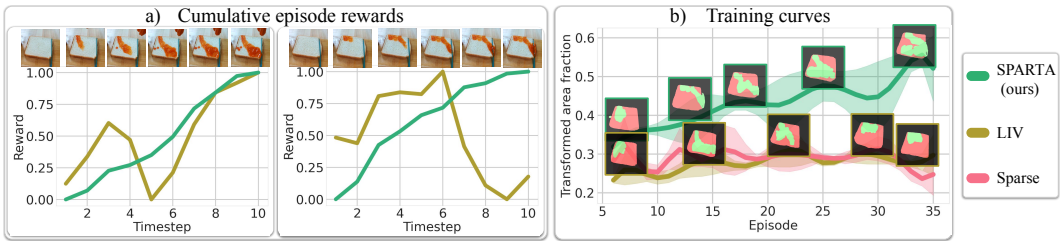


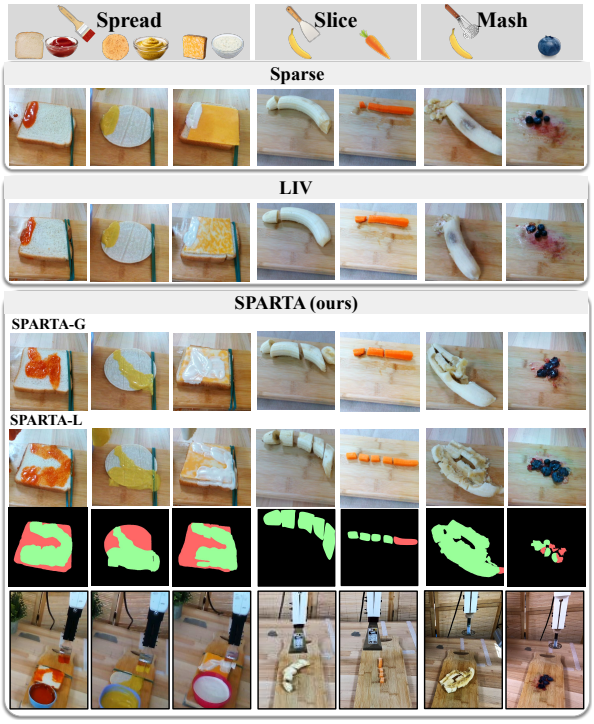
Figure 3: Reward curves for bread-spreading task. **a)** Cumulative episode reward curves: SPARTA produces smooth, incremental rewards aligned with visual progress, while LIV rewards remain unstable throughout the episode, offering poor guidance. **b)** Training curves: stable, dense feedback drives sample-efficient learning, with SPARTA rapidly improving while SPARSE and LIV stagnate.

The baselines represent two dominant approaches: sparse rewards with minimal supervision and pretrained goal-based visual representations. They highlight the limitations of current visual RL methods when applied to fine-grained OSC tasks. We do not include tactile-based or simulation-heavy methods [13, 14], as they require task-specific instrumentation. Further, unlike imitation learning approaches, SPARTA does not require demonstrations. Thus, we focus on general, vision-driven approaches requiring no human demonstrations—hence directly comparable to SPARTA.

Metric. We measure performance using *transformation coverage*: the % of the object's area that has changed state by the end of an episode. Coverage is computed from SPOC segmentations and corrected using [49] with human annotators to ensure reliability. Unlike binary success, this continuous metric captures partial progress, providing a more sensitive evaluation of OSC performance.

Model	Spread					Slice			Mash			
	Seen	*Unseen*				*Seen*	*Unseen*		*Seen*	*Unseen*		
RANDOM	0.24	0.42	0.27	0.29	0.23	0.13	0.15	0.14	0.18	0.14	0.23	0.20
SPARSE	0.14	0.10	0.07	0.11	0.13	0.09	0.08	0.09	0.13	0.08	0.09	0.18
LIV [10]	0.17	0.14	0.12	0.16	0.12	0.10	0.09	0.11	0.13	0.09	0.10	0.08
SPARTA-G	0.44	0.49	0.55	**0.66**	0.39	0.52	0.48	0.51	0.75	0.69	**0.71**	**0.75**
SPARTA-L	**0.61**	**0.55**	**0.58**	0.63	**0.42**	**0.78**	**0.69**	**0.72**	**0.77**	**0.72**	0.62	0.68

(a) SPARTA shows strong training and generalization results for objects with varying textures, colors and shapes. Metric is transformation coverage (%). Results averaged over 3 seeds, 5 rollouts per seed (15 evaluations total).



(b) SPARTA shows strong training and generalization results for objects with varying textures, colors and shapes.

Figure 4: SPARTA results

Experiments and Results:

In our experiments, we aim to answer three key questions:

Q1) How stable and sample-efficient is the learning process? A key determinant of real-world sample efficiency is the stability of dense rewards within each episode [50]. Fig. 3-a shows cumulative reward curves for SPARTA-L and LIV across sample episodes on the bread-spreading task. SPARTA produces *smooth, monotonic* curves that align directly with visual progress, leading to consistent incremental rewards. In contrast, LIV rewards fluctuate erratically, reflecting how goal-conditioned embeddings fail to capture fine-grained transformation dynamics. These unstable signals offer poor guidance, leading to degenerate solutions over time.

This reward stability translates into far more efficient training (Fig. 3-b). SPARTA-L exhibits steep, monotonic learning curves from the very first episodes, often reaching usable policies (>60% coverage) within just 90 minutes of real-world training. By contrast, both SPARSE and LIV remain flat, unable to improve beyond chance due to the absence of dense, progress-aware feedback. Interestingly, the affordance prior also acts as an implicit curriculum: early on, policies focus on small patches of the object, before gradually covering larger regions and learning strategies such as reversing direction near object boundaries.

Together, these results demonstrate how SPARTA's dense rewards provide stable, interpretable feedback that not only reflects spatial progress but also drives sample-efficient policy learning in the real world.

Q2) How well does SPARTA perform complex object state changes? Table 4a evaluates SPARTA on three tasks with seen and unseen objects, with qualitative examples in Fig. 4b. Both variants—SPARTA-G (greedy) and SPARTA-L (learning)—substantially outperform all baselines, highlighting the value of spatial object-centric affordances. SPARSE offers little guidance, and LIV [12], despite strong representation learning, fails to track fine-grained spatial progress. Even the simple RANDOM policy surpasses these baselines, illustrating how weak or unstable rewards can cause RL policies to collapse without sufficient entropy.

Among our methods, SPARTA-G leverages directional priors to reliably target actionable regions, excelling in mashing where symmetric tools reduce sensitivity to noise. SPARTA-L dominates in spreading and slicing, where asymmetric tools demand precise, noise-robust control and long-horizon optimization enables pixel-accurate transformations.

Overall, spatial OSC segmentation emerges as a versatile representation that supports both fast greedy planning and robust reinforcement learning, depending on task demands. We discuss limitations in Appendix Sec. C.

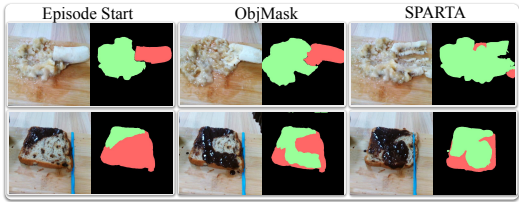


Figure 5: Unlike OBJMASK, which wastes actions on already transformed regions, SPARTA targets only actionable areas for efficient state progression.

Q3) What is the utility of state change segmentations for robot learning over plain object segmentation maps? To isolate the benefit of SPOC affordances over traditional object segmentation, we compare SPARTA-G against a greedy baseline that traverses the entire object segmentation mask without reasoning about state change. We initialize objects in partially transformed states (e.g., a half-mashed banana) and evaluate if the policy can target the remaining untransformed regions (see Fig. 5). The object segmentation baseline, being agnostic to intra-object state change, wastes actions by repeatedly revisiting already transformed regions (e.g., mashing an already mashed banana segment). In contrast, SPARTA-G, exploits SPOC maps to selectively target only the actionable (untransformed) regions, achieving 3× higher coverage efficiency. This validates the efficacy of SPARTA for spatially-progressive manipulation policies that reason over state change dynamics—not just object presence.

6 Conclusion

This work investigates real-world robot learning for a family of spatially-progressing manipulation tasks—such as spreading, mashing, and slicing—by leveraging dense visual affordances that signal object state change. Our method, SPARTA, uses a unified spatial-progress representation to support both greedy planning and reinforcement learning, allowing policy generation for very challenging tasks without simulation or human demonstrations, while being generalizable across diverse objects. Overall, SPARTA demonstrates that progress-aware affordances can unlock a family of object state manipulations essential for everyday tasks, charting a path beyond rigid-body control.

References

[1] Shridhar *et al.*, "Cliport: What and where pathways for robotic manipulation," in *Proceedings of the 5th Conference on Robot Learning (CoRL)*, 2021.

[2] Shao *et al.*, "Concept2robot: Learning manipulation concepts from instructions and human demonstrations," in *Proceedings of Robotics: Science and Systems (RSS)*, 2020.

[3] Bahety *et al.*, "Screwmimic: Bimanual imitation from human videos with screw space projection," in *Robotics: Science and Systems (RSS)*, 2024.

[4] Pinto *et al.*, "Learning to push by grasping: Using multiple tasks for effective learning," in *ICRA*, 2017.

[5] Sharma *et al.*, "Multiple interactions made easy (mime): Large scale demonstrations data for imitation," in *Conference on robot learning*, 2018.

[6] Souček *et al.*, "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] Xue *et al.*, "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] Mandikal *et al.*, "Spoc: Spatially-progressing object state change segmentation in video," in *ArXiv*, 2025.

[9] Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal on Robotics and Automation*, 1987.

[10] Grauman *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*, 2022.

[11] Zhu *et al.*, "The ingredients of real-world robotic reinforcement learning," *ICLR*, 2020.

[12] Ma *et al.*, "Liv: Language-image representations and rewards for robotic control," *arXiv preprint arXiv:2306.00958*, 2023.

[13] Heiden *et al.*, "Disect: A differentiable simulation engine for autonomous robotic cutting," *Robotics: Science and Systems (RSS)*, 2021.

[14] Xu *et al.*, "Roboninja: Learning an adaptive cutting policy for multi-material objects," in *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[15] Beltran-Hernandez *et al.*, "Sliceit!–a dual simulator framework for learning robot food slicing," *International Conference on Robotics and Automation (ICRA)*, 2024.

[16] Shi *et al.*, "Robocook: Long-horizon elasto-plastic object manipulation with diverse tools," *arXiv preprint arXiv:2306.14447*, 2023.

[17] Ye *et al.*, "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.

[18] Chen *et al.*, "Vegetable peeling: A case study in constrained dexterous manipulation," *arXiv preprint arXiv:2407.07884*, 2024.

[19] Dong *et al.*, "Food peeling method for dual-arm cooking robot," in *IEEE/SICE International Symposium on System Integration (SII)*, 2021.

[20] Liu *et al.*, "Robot cooking with stir-fry: Bimanual non-prehensile manipulation of semi-fluid objects," *IEEE Robotics and Automation Letters*, 2022.

[21] Nair *et al.*, "R3m: A universal visual representation for robot manipulation," *Conference on Robot Learning (CoRL)*, 2022.

[22] Radosavovic *et al.*, "Real-world robot learning with masked visual pre-training," *Conference on Robot Learning (CoRL)*, 2022.

[23] Ma *et al.*, "VIP: Towards universal visual reward and representation via value-implicit pre-training," *arXiv preprint arXiv:2210.00030*, 2022.

[24] Ma *et al.*, "Eureka: Human-level reward design via coding large language models," *ICLR*, 2024.

[25] Brohan *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in *CoRL*, 2023.

[26] Brahmbhatt *et al.*, "Contactgrasp: Functional multi-finger grasp synthesis from contact," in *International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[27] Mandikal *et al.*, "Dexterous robotic grasping with object-centric visual affordances," in *International Conference on Robotics and Automation (ICRA)*, 2021.

[28] Mandikal *et al.*, "Dexvip: Learning dexterous grasping with human hand pose priors from video," in *Conference on Robot Learning (CoRL)*, 2021.

[29] Wu *et al.*, "Learning generalizable dexterous manipulation from human grasp affordance," in *Conference on Robot Learning (CoRL)*, 2022.

[30] Agarwal *et al.*, "Dexterous functional grasping," in *Conference on Robot Learning*, 2023.

[31] Bahl *et al.*, "Affordances from human videos as a versatile representation for robotics," *CVPR*, 2023.

[32] Hasson *et al.*, "Learning joint reconstruction of hands and manipulated objects," in *CVPR*, 2019.

[33] Ye *et al.*, "Affordance diffusion: Synthesizing hand-object interactions," in *CVPR*, 2023.

[34] Liu *et al.*, "Joint hand motion and interaction hotspots prediction from egocentric videos," in *CVPR*, 2022.

[35] Yu *et al.*, "Video state-changing object segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[36] Tokmakov *et al.*, "Breaking the "object" in video object segmentation," in *CVPR*, 2023.

[37] Souček *et al.*, "Genhowto: Learning to generate actions and state transformations from instructional videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[38] Miech *et al.*, "Howto100m: Learning a text-video embedding by watching hundred million narrated video clips," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019.

[39] Kanazawa *et al.*, "Real-world cooking robot system from recipes based on food state recognition using foundation models and pddl," *Advanced Robotics*, 2024.

[40] Hu *et al.*, "Slac: Simulation-pretrained latent action space for whole-body real-world rl," *CoRL*, 2025.

[41] Zhang *et al.*, "Rewind: Language-guided rewards teach robot policies without new demonstrations," *arXiv preprint arXiv:2505.10911*, 2025.

[42] Luo *et al.*, *Serl: A software suite for sample-efficient robotic reinforcement learning*, 2024.

[43] Ravi *et al.*, "Sam 2: Segment anything in images and videos," *arXiv preprint arXiv:2408.00714*, 2024.

[44] OpenAI, "Gpt-4 technical report," OpenAI, Tech. Rep., 2023.

[45] Yang *et al.*, "Decoupling features in hierarchical propagation for video object segmentation," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[46] Haarnoja *et al.*, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *ICML*, 2018.

[47] Ball *et al.*, "Efficient online reinforcement learning with offline data," in *ICML*, 2023.

[48] Damen *et al.*, "Scaling egocentric vision: The epic-kitchens dataset," in *European Conference on Computer Vision (ECCV)*, 2018.

[49] Kar *et al.*, *Toronto annotation suite*, https://aidemos.cs.toronto.edu/toras, 2021.

[50] Gupta *et al.*, "Unpacking reward shaping: Understanding the benefits of reward engineering on sample complexity," *Advances in Neural Information Processing Systems*, 2022.

[51] Bahl *et al.*, "Affordances from human videos as a versatile representation for robotics," *CVPR*, 2023.

[52] Nasiriany *et al.*, "Augmenting reinforcement learning with behavior primitives for diverse manipulation tasks," in *International Conference on Robotics and Automation (ICRA)*, 2022.

[53] Ren *et al.*, *Grounded sam: Assembling open-world models for diverse visual tasks*, 2024. arXiv: 2401.14159 [cs.CV].

[54] Radford *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, 2021.

A Implementation Details

Tool-use primitives. To translate high-level policy outputs into physical interactions, we define simple task-specific motion primitives at the predicted 2D location on the object, following prior work using motion primitives for point-based control [51, 52]. For *spreading*, the robot executes in-plane brush strokes, automatically "refilling" the brush every two steps to keep it coated; z-height is fixed relative to the estimated object surface. For *mashing* and *slicing*, it performs a lateral motion followed by a downward press until a preset force threshold is reached. In all tasks, the tool is lifted after each action to avoid visual occlusion before capturing the next observation.

Robot platform. All experiments are conducted on a Franka Emika Panda robot, a 7-DoF collaborative manipulator equipped with torque sensing in each joint and a 3-finger parallel gripper. Its precise joint control and compliant torque feedback make it well-suited for fine manipulation tasks such as spreading, mashing, and slicing. A front-facing camera provides RGB observations for the vision model.

Training details. To keep training grounded in real-world constraints, we set episode lengths to match the natural granularity of each task. For spreading, episodes last 10 steps to reflect the smaller coverage per action, while for mashing and slicing, 5 steps suffice due to the broader area transformed by each action. All episodes begin from a fixed corner of the workspace for consistency. For SPARTA-L, we train policies with short real-world budgets: spreading is trained for 40 episodes ($\sim$3 hours at 1 Hz, including brush refills and resets), while mashing and slicing converge within $\sim$1.5 hours thanks to shorter episodes. To simplify resets, we use clay proxies for mashing and slicing, and bootstrap exploration with a handful ($\sim$5) greedy rollouts, which stabilize early training. For policy learning, we adopt asynchronous SAC from SERL [42], finding that an actor-to-critic update ratio of 1:10 yields the best balance between policy improvement and stable value estimation. Other hyperparameters follow standard practice (learning rate 3e-4 with warmup, $\gamma = 0.95$, reward weights $\alpha = 1, \beta = 1, \eta = 0.001$). Visual inputs are encoded via a ResNet-10 backbone, and proprioceptive inputs through a two-layer MLP. The same training protocol is applied across our method and baselines to ensure fair comparison.

B Integrating SPOC for Robotics

We adapt SPOC for robotics by generating SPOC affordance maps directly from real-time visual observations. While prior work [8] leverages Grounded-SAM [53] and CLIP [54], we find that replacing CLIP with a stronger vision-language model (VLM) such as GPT-4o [44] significantly improves segmentation accuracy—particularly in distinguishing intra-object regions (e.g., partially mashed banana). Instead of assigning a single label to the entire object mask, we sample multiple intra-object regions using farthest-point prompts with SAM, and classify each via GPT-4o into actionable or transformed states. Since per-frame GPT queries are slow ($\sim$5s), we introduce a fast mask propagation strategy using DeAOT [45] tracking ($\sim$0.2s/frame) to boost real-time throughput for robotic control. See Fig. 6 for the full pipeline. These affordance maps offer dense, object-centric structure that is crucial for shaping progress-based rewards and guiding spatial-aware policy learning.

C Limitations and Future Work

While effective, SPARTA also reveals open challenges that suggest avenues 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, although policies generalize to new objects, performance degrades

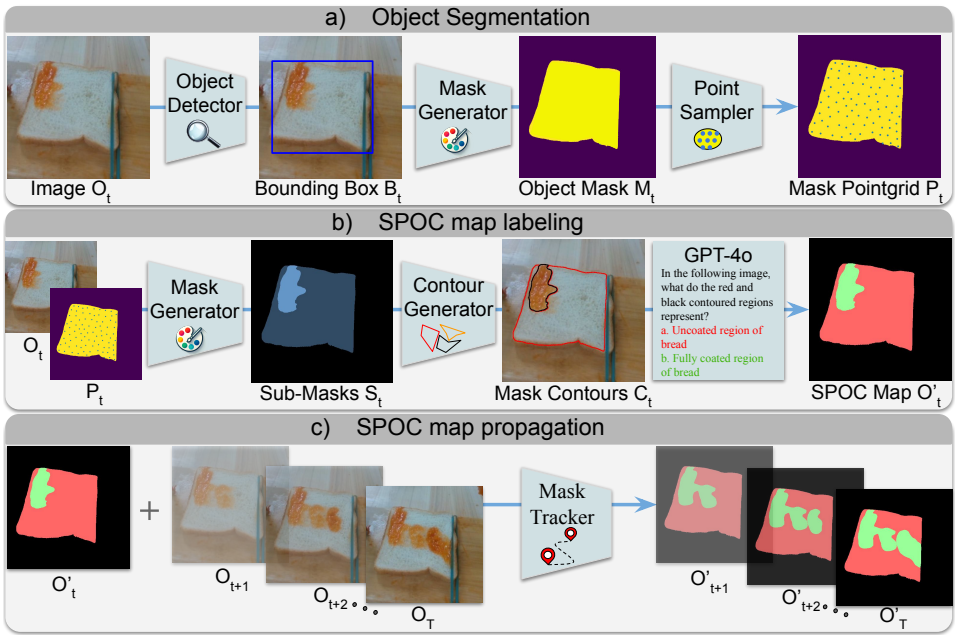


Figure 6: **Our SPOC affordance map generation pipeline.** **(a)** Grounded-SAM [53] is used to extract an object mask from the initial frame. **(b)** Farthest-point sampling generates intra-object regions, classified into *actionable* or *transformed* by prompting GPT-4o [44] using color-coded overlays. **(c)** Once classified, transformed regions are tracked across subsequent frames using DeAOT [45] to maintain temporal consistency with minimal computation.

on unseen geometries—for example, a policy trained on rectangular slices may struggle with circular tortillas. Addressing this gap calls for shape-aware training or augmented experience. Third, our current pipeline avoids occlusion by only capturing visual inputs when the end-effector lifts between actions, precluding continuous perception during contact. Developing occlusion-resilient, contact-aware visual reasoning remains an open challenge. Overall, SPARTA demonstrates that progress-aware affordances can unlock a family of object state manipulations essential for everyday tasks, charting a path beyond rigid-body control.