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

Keywords: non-rigid object manipulation, visual representation, real-world RL

1 Introduction

The dominant paradigm in robotic manipulation focuses on rigid body motion tasks—such as picking and placing [1], opening and closing [2, 3], pushing [4], or rotating objects [5]. These tasks are foundational but primarily involve changing an object's kinematic state, with progress easily tracked via pose changes. In contrast, many real-world scenarios require a different class of manipulations: *object state changes* (OSC) [1] [6, 7, 10], where an object's physical state and appearance evolve without necessarily altering its pose (Fig. 1, top). Everyday examples include *mashing* a banana, *spreading* jam, or *slicing* a cucumber. Such tasks demand sustained interaction that progressively alters shape, texture, and color—making them mechanically and visually complex. Despite their ubiquity in daily life—from cooking (e.g., grating, peeling, shredding) to chores (e.g., painting, wiping, ironing)—OSC tasks remain largely underexplored in robotics.

What makes OSC manipulation challenging? Unlike motion-centric tasks, OSC demands continuous reasoning about which parts of a deformable object have already transformed, which have not, and how to act next. Two key obstacles arise. First, at the *representation level*, raw RGB observations conflate appearance with state, obscuring progress signals and hindering generalization.

[1] We adopt the term "object state change" (OSC) from the vision literature [6–8]: an OSC is a transformation that yields a visually distinct post-condition (e.g., chopped apple) following an action (e.g., chopping), often with irreversible changes to morphology, texture, or appearance. Not to be confused with Operational Space Control [9].

Second, at the *learning level*, rewards are difficult to define: sparse success signals give little exploration guidance [11], while goal-conditioned rewards [12] often depend on scene embeddings that miss *fine-grained, incremental progress* central to OSC. As a result, current approaches are sample-inefficient and poorly suited to tasks where state changes evolve dynamically within the object.

To address these challenges, we propose SPARTA (Spatial Progress-Aware Robotic object TransformAtion)—a system that introduces *structured, progress-aware visual affordances* for OSC manipulation (Fig. 1, bottom). SPARTA builds on recent vision advances in detecting OSC (SPOC [8]), which segment an object into *actionable* and *transformed* regions. For example, in mashing a potato, unmashed chunks are actionable while mashed portions are transformed. SPARTA exploits SPOC affordance[2] maps in two ways: (1) as structured visual inputs that filter appearance while preserving progress cues, enabling generalization; and (2) as dense, spatially grounded rewards that capture incremental progress. By explicitly encoding "what has changed" and "what remains," SPARTA enables robots to reason about state progression rather than object kinematics alone.

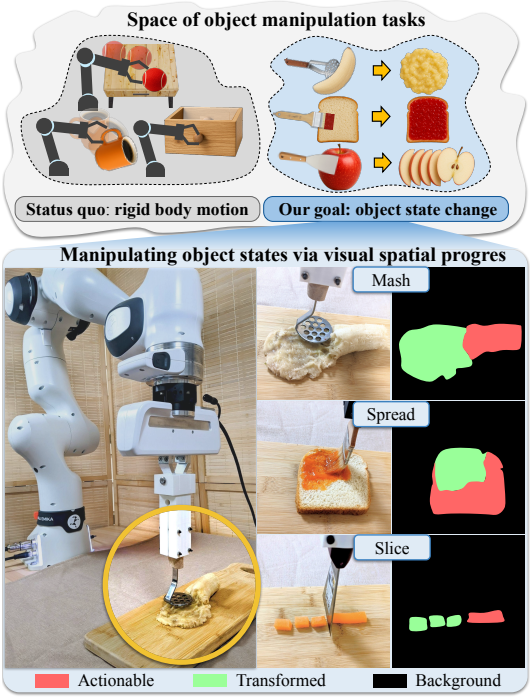

Figure 1: **Top:** While most robotic manipulation focuses on rigid-body motion, many real-world tasks involve *object state changes* such as mashing, spreading, or slicing, where objects are progressively transformed. **Bottom:** SPARTA leverages spatially-progressing affordance maps of *actionable*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

Our formulation supports two policy variants. SPARTA-L uses SPOC-derived rewards to train RL agents directly in the real world—without demonstrations or simulation—achieving *highly* sample-efficient learning. In contrast, SPARTA-G provides a non-parametric alternative, greedily acting on nearby actionable regions in the SPOC map. This unified framework thus accommodates both: (1) reinforcement learning, for robust, adaptive control under noise and uncertainty; and (2) greedy control, for fast, lightweight deployment without training. Together, these variants highlight the versatility of SPARTA's progress-aware affordances: a single representation can drive both 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 Robotic Object State Change

Our goal is to enable robots to perform *object state change* (OSC) tasks, where an object's morphology, texture, or appearance evolve over time. Unlike traditional manipulation of rigid bodies (e.g., pick-and-place, pushing), OSC requires reasoning about transformations *within* the object. The challenge is not merely altering pose, but deciding where and how to act on deformable regions to drive continuous, often irreversible changes. This reframes the problem: the robot must perceive gradual

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

transformations, localize actionable regions, and sequence fine-grained actions that accumulate into 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: $J(\pi) = \mathbb{E}_\pi\left[\sum_{t=0}^{T-1} \gamma^t r(s_t, a_t)\right]$. Partial observability arises as the object's true state (e.g., which region of a banana is mashed) is hidden—only visual observations and proprioception are available. Unlike motion-centric tasks where object pose suffices, OSC requires observation spaces that approximate these evolving, spatially localized states.

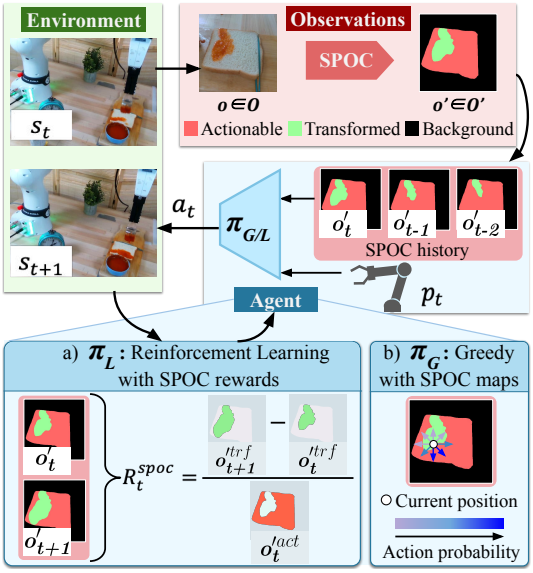

Figure 2: **Overview of SPARTA.** At each episode step, our policy takes the current and past SPOC [8] visual-affordance (segmentation) maps as inputs, along with the robot arm's proprioception data and predicts a displacement action for the arm's end-effector. SPARTA supports two robot policy variants: (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 on a tabletop with a single object placed. Each observation $\omega_t \in \Omega$ has visual and proprioceptive components, $\Omega = O \times P$: an RGB frame $o_t \in O$ from a fixed camera and proprioceptive input $p_t \in P$ encoding end-effector position. Raw RGB frames, though visually rich, conflate object appearance with state dynamics, making it hard to learn sample-efficient, generalizable policies from limited data. What is needed are structured visual abstractions that discard appearance detail while preserving cues of state evolution—bringing observations closer to the task-relevant state.

Action Space. Classical manipulation often plans global object motions, whereas OSC tasks require actions at *specific intra-object locations* to drive local transformations (e.g., pressing unmashed potato chunks or brushing uncoated bread). We therefore constrain the action space to a 2D manifold aligned with the object surface, allowing policies to reason directly about *where* to act. The policy outputs continuous $\Delta x, \Delta y$ displacements, sampled from a Gaussian around the predicted mean. At the resulting (x, y), a task-specific primitive is executed—sweeping for spreading, pressing for mashing, or slicing strokes. This structured action space captures the spatially progressive nature of OSC while reducing complexity, enabling sample-efficient policies that generalize across objects.

3 SPARTA: Robot Policies for OSCs via Visual Spatial Progress

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 [13] + GPT-4o [14] with DeAOT [15] propagation for real-time control (details in Appendix Sec. C). 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, which uses SPOC rewards for real-world online RL, and SPARTA-G, 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.

SPARTA-L: Reinforcement Learning with SPOC rewards. OSC tasks require sequential decision-making, as each action transforms only a local region and the robot must decide *where to act next*. RL is well-suited for this setting, but sparse success signals hinder exploration while dense feedback is rarely available [11]. To address this, SPARTA-L introduces a dense, spatially grounded reward that combines a sparse terminal success term R_t^{succ}, an entropy bonus $R_t^{entropy}$, and a novel SPOC-based progress reward $R^{spoc}t = \frac{At+1^{trf} - A_t^{trf}}{A_t^{act}}$ capturing newly transformed area between timesteps (see Fig. 2a). This formulation rewards incremental, non-redundant progress in actionable regions, yielding an object-centric, task-agnostic signal derived directly from vision. Crucially, it enables *real-time, demonstration-free* training: policies are optimized with SERL [16] using SAC [17] and RLPD regularization [18], but unlike SERL, no demonstrations are needed. SPOC-derived rewards alone are sufficient for stable, sample-efficient real-world learning (Sec. 4).

SPARTA-G: Greedy Policy with SPOC Maps. While RL provides robustness under noisy perception, some OSC tasks can be solved with simpler controllers. For example, with large, symmetric tools (e.g., a masher), each action covers a broad area, so a greedy strategy that steers toward untransformed regions suffices, unlike tasks with thin, directional tools (e.g., spreading) where RL excels. To capture this easier regime, we introduce SPARTA-G, a non-parametric greedy controller that uses SPOC maps to guide actions. At each step, the agent samples eight candidate motions in the xy-plane and selects the direction leading to the highest density of actionable pixels around the predicted endpoint, effectively driving the tool toward regions most likely to yield progress (see Fig. 2b). Though training-free, SPARTA-G still fits within the MDP framework as a deterministic mapping from SPOC states to actions, making it lightweight and fast to deploy for coarse transformations, while SPARTA-L remains superior for fine-grained control.

4 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 4), testing both visual robustness and policy generalization.

Comparisons. We benchmark against three baselines: (1) RANDOM, uniform exploration within the action space; (2) SPARSE, using only a binary GPT-4o–queried success reward from the final image (e.g., "Is the bread fully coated?"); and (3) LIV [12], a state-of-the-art goal-conditioned method that computes rewards from video-trained embeddings prompted with natural language (e.g., "coat bread with ketchup"). These baselines represent the two dominant strategies in visual RL—sparse rewards and pretrained goal representations—highlighting their limitations for fine-grained OSC. We exclude tactile- or simulation-heavy approaches [19, 20] that require specialized setups, and unlike imitation-based methods, SPARTA needs no demonstrations, making these vision-driven comparisons most directly relevant. We evaluate using *transformation coverage*, the percentage of object area changed state (from SPOC segmentations corrected with human annotations [21]), which captures partial progress beyond binary success.

How stable and sample-efficient is the learning process? Dense, stable rewards are crucial for real-world efficiency [22]. As shown in Fig. 3a, SPARTA-L yields smooth, monotonic reward curves aligned with visual progress, while LIV produces noisy signals that fail to capture fine-grained transformations. This stability drives steep, monotonic learning curves (Fig. 3b), with SPARTA-L reaching usable policies (>60% coverage) in just 90 minutes, whereas SPARSE and LIV stagnate. Moreover, affordance priors provide an implicit curriculum, guiding policies from lo-

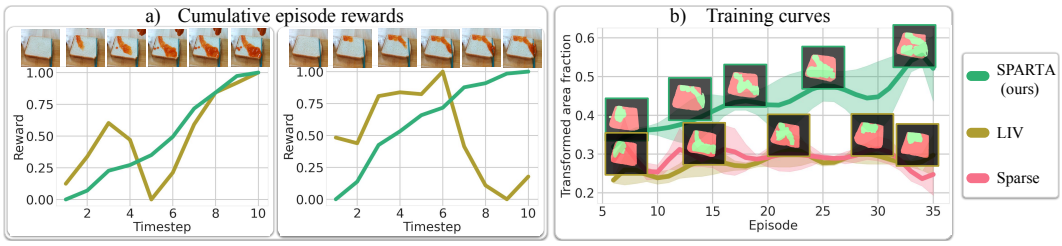

Figure 3: Reward curves for bread-spreading. **a)** SPARTA yields smooth, progress-aligned rewards, while LIV remains unstable. **b)** Dense feedback enables rapid learning, whereas SPARSE and LIV stagnate.

cal patches to full-object strategies. Thus, SPOC-based rewards deliver interpretable feedback that enables stable, sample-efficient real-world learning.

How well does SPARTA perform complex object state changes? As shown in Fig. 4 (qualitative results in Fig. 5), both SPARTA variants far outperform all baselines, highlighting the strength of spatial affordances for OSC. SPARSE and LIV [12] fail to capture fine-grained progress, with RANDOM even surpassing them due to weak reward signals. Among our methods, SPARTA-G excels in mashing, where symmetric tools mitigate perceptual noise, while SPARTA-L dominates spreading and slicing, where precise, noise-robust control is critical. Overall, SPOC affordances provide a versatile visual representation, supporting both greedy planning and reinforcement learning depending on task demands.

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

Figure 4: 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).

What is the utility of state change segmentations over plain object masks? We compare SPARTA-G to a greedy baseline using only object masks, initialized with partially transformed states (e.g., half-mashed banana). The baseline, blind to intra-object changes, repeatedly revisits transformed regions, while SPARTA-G targets only actionable areas, achieving 3× higher coverage efficiency. This shows that reasoning over state change dynamics—not just object presence—is critical for spatially progressive manipulation.

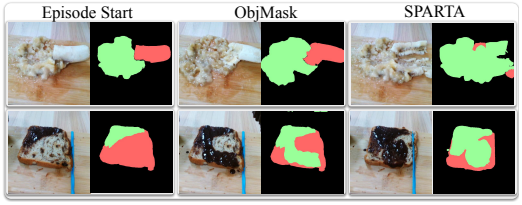

Figure 6: Unlike OBJMASK, which wastes actions on already transformed regions, SPARTA targets only actionable areas for efficient state progression.

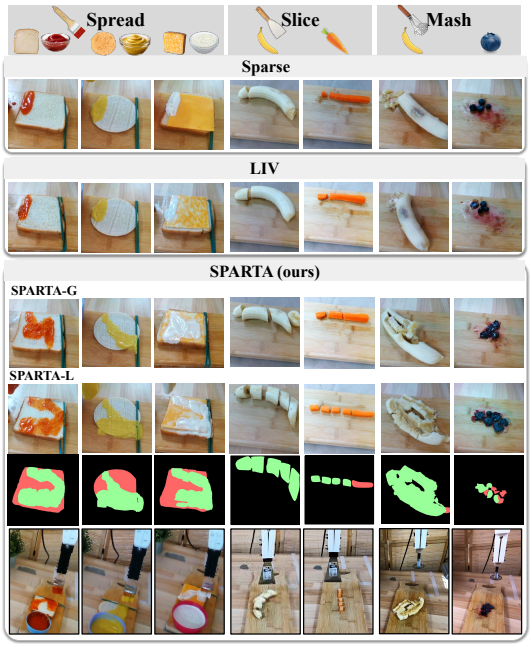

Figure 5: SPARTA outperforms baselines by transforming actionable regions across diverse objects with varying colors, shapes, and textures.

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

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

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

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

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

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

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

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

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

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

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

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

[25] 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.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[45] 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.

[46] 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.

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

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

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

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

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

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

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