BiPreManip: Learning Affordance-Based Bimanual Preparatory Manipulation through Anticipatory Collaboration

Supplementary Material

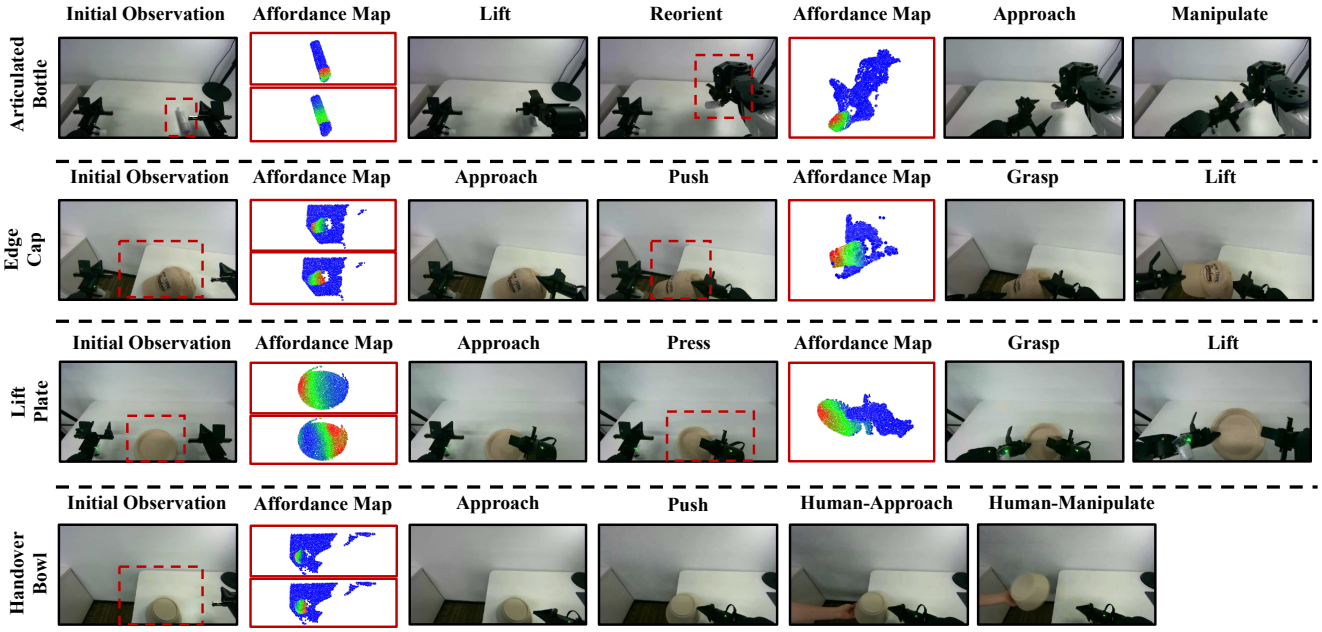

Figure 1. **Real-World experiments.** We present additional qualitative visualizations of the predicted affordance maps and corresponding actions generated by our method. In each example, the second column displays the predicted affordances: the top subfigure depicts the *anticipatory affordance* for the primary arm, while the bottom subfigure shows the *pre-manipulation affordance* for the assistant arm. Manipulation videos are included in the supplementary material.

1. Additional Real-World Results

In real-world experiments, the predicted $\mathrm{SE}(3)$ gripper poses are first validated for inverse kinematics feasibility and controller constraints, and then executed via time-parameterized interpolation with linear translation and SLERP-based rotation under predefined execution tolerances. Actions that fail these feasibility checks are recorded as failures without recovery or resampling (but deployment users may instead resample actions from the affordance and retry execution).

Figure 1 presents additional real-world qualitative results that complement the visualizations shown in Figure 5 of the main paper.

For each example, we provide the predicted affordance maps and the corresponding actions generated by our method. In the second column, the top subfigure illustrates the *anticipatory affordance* for the primary arm, while the bottom subfigure depicts the *pre-manipulation affordance* for the assistant arm. The fourth column displays the *goal affordance* associated with the primary arm's goal-directed action. Together, these visualizations highlight how the model effectively coordinates both arms to achieve stable pre-manipulation behaviors that enable successful task execution.

Beyond the articulated manipulation, edge-pushing, and plate-lifting tasks, we further evaluate a robot–human handover scenario. This experiment demonstrates that our approach is not only effective for bimanual collaborative preparatory manipulation but also readily extends to robot–human interaction. In this new setting, the model anticipates the human's intended goal, analogous to how it predicts the primary arm's objective in the bimanual manipulation scenario, and acts as an assistant by performing preparatory actions that deliver the object in a configuration suitable for comfortable and efficient human use. This

scenario further illustrates the versatility and practical applicability of our proposed method.

Manipulation videos are provided in the supplementary material.

2. Additional Visualizations in Simulation

We present additional visualization results from simulation in Figure 2, covering different task types and object categories, to augment Figure 4 in the main paper.

3. Data Statistics

We provide detailed statistics of the datasets used in our experiments in Table 1. This table lists the number of object instances in the training and unseen test sets for each object category across all task types. As described in Section 4.1 of the main paper, objects are randomly divided into training and unseen (novel) sets with a 3:1 ratio. After training the model on the training objects, we construct two evaluation sets: one containing seen (training) objects with randomly varied initial poses, and another containing unseen (novel) objects with distinct shapes and poses.

We further note that, due to the object-level random split, some unseen (novel) objects may exhibit simpler geometries than those in the training set, leading to consistently higher accuracy across all methods for certain categories.

Table 1. Number of object instances per category in the training and unseen test sets.

Task Type	Category	Train	Unseen	Total
Edge-Pushing Tasks	Bowl	140	46	186
	Cap	42	14	56
	Keyboard	49	16	65
	Laptop	42	13	55
	Phone	14	4	18
	Remote	37	12	49
	Scissors	36	11	47
	Switch	53	17	70
	Window	44	14	58
	Subtotal	**457**	**147**	**604**
Articulated Manipulation Tasks	Bottle	43	14	57
	Dispenser	43	14	57
	Lighter	12	3	15
	Pen	36	12	48
	Pliers	19	6	25
	Stapler	18	5	23
	USB	23	7	30
	Subtotal	**194**	**61**	**255**
Plate-Lifting Tasks	Plate	17	5	22
	Subtotal	**17**	**5**	**22**
All Tasks	**Overall Total**	**668**	**213**	**882**

4. Dataset Collection and Heuristic Baseline

This section details the data collection procedures for the three simulated task sets: (1) Articulated Manipulation, (2) Edge-Pushing, and (3) Plate-Lifting. To enable efficient large-scale data generation, we incorporate a set of handcrafted heuristic strategies that guide action selection and substantially reduce failed rollouts.

The Heuristic Baseline reported in the main paper is implemented using the same set of heuristics.

4.1. Articulated Manipulation Tasks

For each trial, we randomly sample an articulated object from the PartNet-Mobility dataset [5, 7]. The object is placed near the table center with a randomized 6D pose, and an overhead depth camera provides RGB-D observations throughout the episode.

4.1.1. Pre-Grasp Action

The first step is to generate a pre-grasp pose for the assistant arm. Using the depth map and link masks rendered from simulation, we sample a pixel on a non-articulated (fixed) link of the object. This bias toward the non-articulated part helps reduce the likelihood of the assistant arm interacting with articulated parts, thereby mitigating potential inter-arm interference during the subsequent goal-directed manipulation. The selected pixel is then back-projected to obtain a 3D grasp point.

The gripper's orientation is determined using simple geometric rules. The gripper's forward direction (x-axis) is oriented approximately downward toward the object, with a small (10°) randomized deviation for diversity. The left direction (y-axis) is defined according to a category-specific canonical direction in the object's local coordinate frame (e.g., for elongated objects, the left direction is set perpendicular to the object's principal axis). This direction is then transformed from the object's local coordinate frame to the world frame, based on the object's sampled pose, promoting consistency of the gripper orientation relative to the object across different placements. A small random angular perturbation is added for further diversity.

Together, the forward (x), left (y), and derived up (z) axes define a complete 6-DoF pre-grasp pose, which is used for the assistant arm to grasp. The assistant arm then lifts the object by elevating the gripper.

4.1.2. Reorientation Action

After grasping, the assistant arm reorients the object to expose its articulated component to the primary arm. The desired object pose is defined using category-dependent rules. For example, in the case of a bottle, the lid is rotated to face the primary arm and positioned within the shared reachable

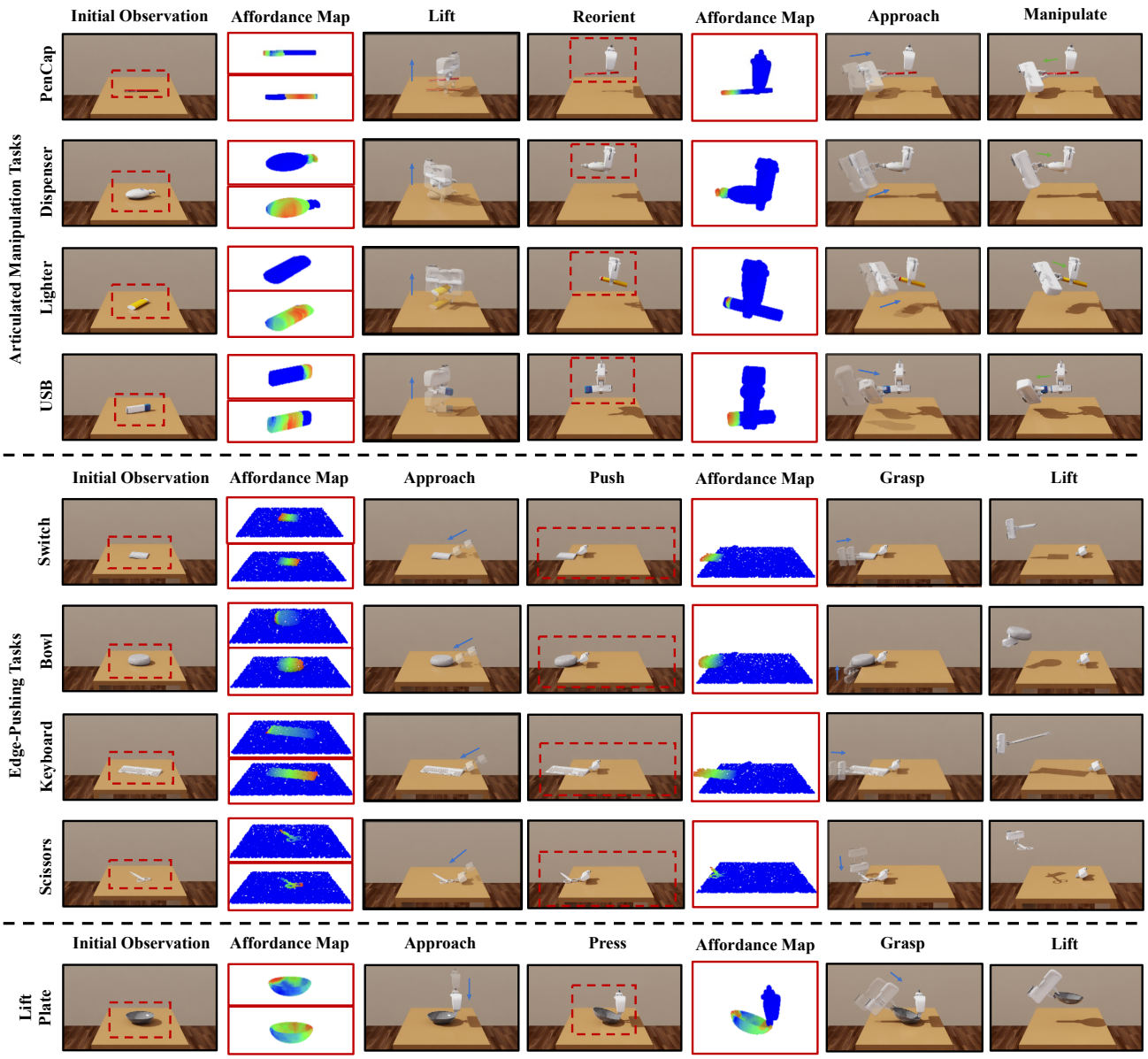

Figure 2. **Simulation experiments.** We present additional qualitative visualizations of the predicted affordance maps and corresponding actions produced by our method. In each example, the second column displays the predicted affordances: the top subfigure depicts the *anticipatory affordance* for the primary arm, while the bottom subfigure shows the *pre-manipulation affordance* for the assistant arm.

workspace of both arms.

Let $T_{\text{obj}}^{\text{grasped}} \in SE(3)$ and $T_{\text{grp}}^{\text{grasped}} \in SE(3)$ denote the object and gripper poses after grasping. The desired object pose, $T_{\text{obj}}^{\text{reorient}}$, defines the target gripper pose $T_{\text{grp}}^{\text{reorient}}$ by preserving the relative transform between the gripper and the object:

$$T_{\text{grp}}^{\text{reorient}} = T_{\text{obj}}^{\text{reorient}} \left(T_{\text{obj}}^{\text{grasped}}\right)^{-1} T_{\text{grp}}^{\text{grasped}}. \quad (1)$$

The assistant arm then executes this reorientation action, positioning the object into the desired configuration to enable reliable goal-directed manipulation by the primary arm.

4.1.3. Goal-Directed Action

A new depth image is rendered after reorientation. We then sample a pixel on the articulated link and back-project it to obtain the target 3D contact point. The gripper's forward (x) axis is aligned with the articulation's functional direction (e.g., opening or pushing), with a small probability. The

left (y) and up (z) axes are randomly assigned to introduce variability.

4.2. Edge-Pushing Tasks

For each trial, an object is randomly sampled from the ShapeNet [1] or PartNet-Mobility [5, 7] datasets. The object is then placed on the table with a randomized pose.

The assistant arm begins with preparation manipulation. A random pixel is sampled from the object in the depth image and back-projected to determine the 3D contact point. The gripper's forward (x) axis is oriented roughly downward toward the object, with a random deviation for variability. The gripper's left (y) axis is roughly aligned with the direction of the table edge toward which the object is intended to be pushed, biasing the configuration so that the gripper's fingers, rather than the gripper body, are more likely to contact the object. Together, this orientation and the selected contact point define the gripper's initial approach pose. The robot then pushes the object toward the table edge, while maintaining the orientation and translating the gripper in the pushing direction. Random deviations are applied to introduce variability in the poses.

After the preparatory motion, if the object remains on the table, the primary arm attempts to grasp it at the edge. A new depth image is captured, and a pixel is sampled from the part of the object extending beyond the table, from which a new 3D contact point is computed. The gripper's orientation is then determined based on the object's geometry. For thin objects, such as a keyboard or phone, the gripper's forward (x) axis is aligned horizontally and perpendicular to the exposed object edge, while the left (y) axis is aligned vertically relative to the table surface. For objects like an inverted bowl, the gripper's forward (x) axis is oriented from underneath to grasp the bowl's rim, with the left (y) axis assigned randomly. All generated axes are perturbed with small random deviations to increase pose diversity. Finally, after the primary arm grasps the object, it lifts the object to a higher position. The grasp is considered successful if the object remains stable and does not fall, indicating that the primary arm has maintained a secure grip.

4.3. Plate-Lifting Tasks

This task is adapted from the PerAct2 [3] benchmark. Following their setup, the assistant arm first presses down one side of the plate. To do this, we randomly select a point along the plate's outer edge as the pressing point. The gripper's forward (x) axis is oriented vertically downward, and its left (y) axis is aligned with the tangent direction of the plate's boundary at the selected point. Small random perturbations are applied to each axis to introduce orientation variability. Once the plate is pressed down, the primary arm selects a grasping point from the top 10% of points on the plate with the highest z-coordinates. For the chosen point, the gripper's forward (x) axis is aligned normal to the plate edge at that location, while the up (z) axis is aligned parallel to the corresponding tangent direction, promoting a stable grasp. After closing the gripper, the primary arm lifts the plate to a higher position. The grasp is considered successful if the plate remains stable and does not fall.

5. Training Details and Computational Costs

As described in Section 3.6.3 of the main paper, all modules in our framework are jointly optimized using a weighted sum of their respective loss functions. Gradients are propagated through shared feature encoders, promoting consistent and coherent feature representations across the network.

Training the BiPreManip framework on a single NVIDIA V100 GPU requires approximately 24 hours to converge.

During inference, the framework consumes 1,166 MB of GPU memory. The average model inference time (excluding physics simulation) for a complete rollout is 0.27 seconds. Specifically, the pre-grasping manipulation stage takes approximately 0.12 seconds, the reorientation stage requires 0.08 seconds, and the final goal-directed action stage takes 0.07 seconds.

6. Failure Case Analysis

Figure 3 illustrates several common failure modes of our model. Below, we discuss the underlying causes of these errors and outline potential directions for improvement.

(a) Challenges posed by small components. In this case, the assistant arm attempts to grasp the non-movable body of a USB drive while avoiding its movable cap, as disturbing the cap would impede the subsequent grasp by the primary arm. However, due to the small size of the non-movable body, the assistant gripper tends to approach very close to the bottom edge. This often results in grasps that are too close to the boundary, yielding an unstable hold.

(b) Constraints of one-shot pushing. Here, the assistant arm attempts to push the object toward the table edge, but the object becomes unstable during the motion. The primary reason is that our task setting allows only a single pushing attempt. Achieving a stable final pose with a single push is inherently difficult, especially for objects that shift unpredictably. A reasonable mitigation strategy is to allow multiple small corrective pushes, enabling the robot to iteratively adjust its contact points and orientations in a closed-loop manner. However, such iterative pushing strategies fall outside the central focus of this work, which emphasizes inter-arm collaboration. We leave multi-step closed-loop pushing strategies to future work.

(c) Action ambiguity caused by occlusion. After the primary arm grasps and lifts the lighter, the gripper unin-

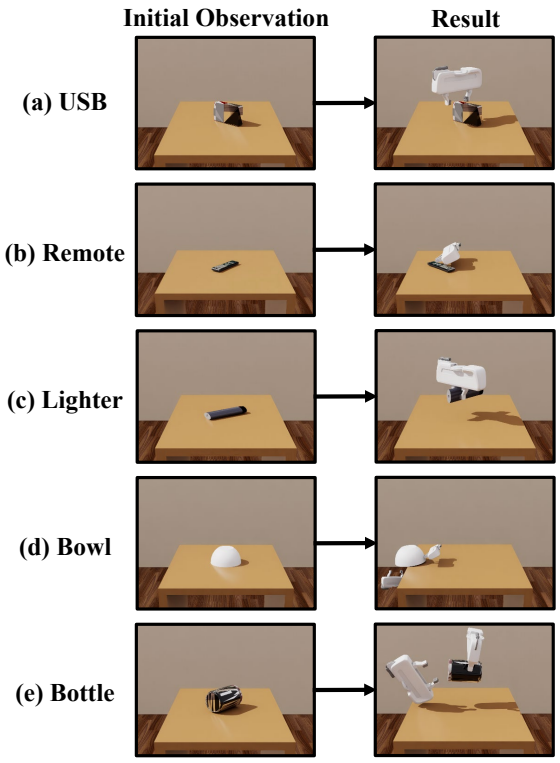

Figure 3. **Representative failure cases.** These examples illustrate scenarios in which the robot struggles to determine appropriate actions, reflecting the inherent difficulty of the tasks. The first column shows the input observations, and the second column presents the resulting failures.

tentionally occludes the lighter's ignition button from the camera's viewpoint. This occlusion introduces ambiguity when the model predicts the subsequent reorientation action, sometimes resulting in incorrect decisions. This limitation arises because our setup uses only a single camera. A multi-camera setup could substantially reduce such ambiguities. Another potential direction is to teach the policy to reorient objects proactively to reveal occluded regions, similar to strategies used in object-reconstruction tasks [2, 4, 6].

(d) Geometric constraints near the table surface. In this case, the primary arm attempts to grasp the rim of a small-diameter bowl but collides with the table surface. We observe a high failure rate for such bowls because their geometry significantly restricts feasible grasp poses, increasing the likelihood of table contact.

(e) Sensitivity of interaction poses. Objects such as bottles with wide, shallow lids require highly precise grasp poses. Even minor deviations in the gripper's orientation or position can result in failure. Including several such objects in our benchmark allows for a more comprehensive evaluation of the model's manipulation precision.

References

[1] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 4

[2] Saptarshi Dasgupta, Akshat Gupta, Shreshth Tuli, and Rohan Paul. Uncertainty-aware active learning of nerf-based object models for robot manipulators using visual and re-orientation actions. *arXiv preprint arXiv:2404.01812*, 2024. 5

[3] Markus Grotz, Mohit Shridhar, Yu-Wei Chao, Tamim Asfour, and Dieter Fox. Peract2: Benchmarking and learning for robotic bimanual manipulation tasks. In *CoRL 2024 Workshop on Whole-body Control and Bimanual Manipulation: Applications in Humanoids and Beyond*, 2024. 4

[4] Zhizhou Jia, Shaohui Zhang, and Qun Hao. An efficient projection-based next-best-view planning framework for reconstruction of unknown objects. *arXiv preprint arXiv:2409.12096*, 2024. 5

[5] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 909–918, 2019. 2, 4

[6] Nicholas Pfaff, Evelyn Fu, Jeremy Binagia, Phillip Isola, and Russ Tedrake. Scalable real2sim: Physics-aware asset generation via robotic pick-and-place setups. *arXiv preprint arXiv:2503.00370*, 2025. 5

[7] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, Li Yi, Angel X. Chang, Leonidas J. Guibas, and Hao Su. SAPIEN: A simulated part-based interactive environment. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 4