hand contacting the object; (2) the object being lifted to a position near the other hand; (3) the other hand contacting the object; (4) object being transferred to the final goal position. The reward can then be defined based on solely the "contact goals" and "object goals": each contact goal can be specified by penalizing the distance from fingers to desirable contact points or simply the object's center-of-mass position; each object goal can be specified by penalizing the distance from its current state (e.g., current *xyz* position) to its target state (e.g., target *xyz* position). To reduce the difficulty of specifying contact goals, we propose a novel keypoint-based technique as follows: for each simulated asset, we procedurally generate a set of "contact stickers" attaching to the surface of the object, where each sticker represents a potentially desirable contact point. The contact goal, in terms of reward, can then be specified as

$$r_{\text{contact}} = \sum_i \left[\frac{1}{1+\alpha d(\mathbf{X}^L, \mathbf{F}_i^L)} + \frac{1}{1+\beta d(\mathbf{X}^R, \mathbf{F}_i^R)} \right], \tag{1}$$

where $\mathbf{X}^L \in \mathbb{R}^{n\times 3}$ and $\mathbf{X}^R \in \mathbb{R}^{m\times 3}$ are the positions of contact markers specified for left and right hands, $\mathbf{F}^L \in \mathbb{R}^{4\times 3}$ and $\mathbf{F}^R \in \mathbb{R}^{4\times 3}$ are the position of left and right fingertips, α and β are scaling hyperparameters, and d is a distance function defined as

$$d(\mathbf{A}, \mathbf{x}) = \min_i \|\mathbf{A}_i - \mathbf{x}\|_2. \tag{2}$$

We show a visualization of contact markers in Figure 2B, and experimental results on their effectiveness in Section IV.

C. Sample Efficient Policy Learning

Due to the sample complexity and reward sparsity in exploring a high-dimensional space — especially on a humanoid embodiment with multi-fingered hands — policy learning can take a prohibitively long time, even with a well-defined reward function. We propose two techniques that more effectively improve the sample efficiency of policy learning: (1) initializing tasks with task-aware hand poses; (2) dividing a challenging task into easier sub-tasks, then distilling the sub-task specialists into a generalist policy.

Task-aware hand poses for initialization. We reduce the exploration challenge by collecting task-aware hand pose data from humans. This can be done by connecting any teleoperation system for bimanual multi-fingered hands to the simulator of choice. The collected states, including object poses and robot joint positions, are then randomly sampled as task initialization states in simulation. Distinct from prior works that require full demonstration trajectories [3], we find that teleoperators need not accomplish the task and only need to "play around" with the task goal in mind while environmental states are collected. The approach massively reduces the time needed for teleoperation since human operators do not need to spend time "ramping up" to collect high-quality data. In our experiments, each task requires less than 30 seconds for sufficient amount of hand pose data to be collected.

Divide-and-conquer distillation. Previous methods to improve sample efficiency of policy learning mostly focus on exploring the state space more efficiently [7, 30, 39, 52]. However, these methods do not reduce the difficulty of the exploration problem fundamentally: the probability of receiving learning signals from exploring the "right" states remains the same. Following this observation, we reason that an easier way to overcome the exploration problem in sparse reward settings is to break down the explorable state space itself. For example, a multi-object manipulation task can be divided into multiple single-object manipulation tasks. After dividing a complex task into easier sub-tasks, we can train specialized policies for each sub-task and distill them into a generalist policy. Another benefit of this approach is that we can flexibly filter out trajectory data from the sub-task policies based on their optimality and only retain high-quality samples for training. This effectively brings RL closer to learning from demonstrations, where the sub-task policies act as teleoperators for task data collection in the simulation environment, and the generalist policy acts as a centralized model trained from curated data.

D. Vision-Based Sim-to-Real Transfer

Transferring a policy learned in simulation to the real world is challenging because of the sim-to-real gap. In the case of vision-based dexterous manipulation, the sim-to-real gap stems from both dynamics and visual perception — both are challenging open research problems to solve. We outline two key techniques we employ to reduce the gap.

Mixing object representations. Object perception is crucial for dexterous manipulation because the task is inevitably coupled with object interaction. Prior works that show successful sim-to-real transfer of manipulation policies have explored a wide range of object representations, including (in order of increasing dimensionality and complexity) 3D object position [31], 6D object pose [2], depth [35, 42], point cloud [33], and RGB images [17]. There is a delicate trade-off between using these different object representations: while higher-dimensional representations encode richer information about the object, the larger sim-to-real gap in those data modalities makes the learned policy harder to be transferred; on the other hand, it is harder to learn optimal policies with lower-dimensional object representations because of the limited amount of information. We, therefore, propose a combination of both types of object representations to balance the trade-offs: a low-dimensional 3D object position and a high-dimensional depth image. Importantly, the 3D object position is obtained from a third-view camera to ensure the object is also in camera view and its noisy position can be consistently tracked. The depth image complements with information on object geometry. We provide more empirical validation on the effectiveness of this approach in Section IV.

Domain randomization for dynamics and perception. We apply a wide range of domain randomizations to ensure reliable sim-to-real transfer. We list the details in Appendix C.