Supplementary Material
Dream2Real: Zero-Shot 3D Object Rearrangement with Vision-Language Models

Ivan Kapelyukh*[1,2], Yifei Ren*[1], Ignacio Alzugaray[2], Edward Johns[1]

I. Experiment Setup

A. List of Instructions

Here we present the list of the 10 user instructions used to evaluate the methods across 10 tasks in our experiments:

1) *"put the apple inside the blue and white bowl"*
2) *"put the apple beside the blue and white bowl"*
3) *"put the orange inside the blue and white bowl"*
4) *"put the cookies inside the square metal box"*
5) *"put the banana inside the wicker basket"*
6) *"move the black 8 pool ball so that there is a triangle made of balls on a pool table"*
7) *"move the black 8 ball so that there are balls in an X shape"*
8) *"move the strawberry milkshake bottle to make three milkshake bottles standing upright in a neat row"*
9) *"put the strawberry milkshake bottle near the plant"*
10) *"place the strawberry milkshake bottle standing upright in front of the white book"*

B. Handling Unreachable Poses

In Section IV-D of the main paper, we demonstrate how the goal pose for the object to be moved may be used by a robot to physically grasp and rearrange the object. To demonstrate this, we use standard grasp prediction and motion planning techniques as described in Section III-E of the main paper. However, it may not be possible for the robot to pick up the object and place it immediately into the predicted goal pose with the predicted goal orientation (e.g. due to kinematic constraints). This issue is explained and addressed in [1], which can be integrated into our framework in future work. For our setting, we find that automatically eliminating unreachable poses during the physics check stage (before rendering) is an efficient and effective solution. The heuristic we apply works as follows. Consider the face of the object which points upwards (away from the table) in the object's initial pose. This is the face that will be grasped by the robot's suction gripper, and so we refer to it as the grasping face. When the robot is imagining new poses for this object, we discard sampled poses where this grasping face is neither facing upwards nor facing towards the robot arm. Intuitively, this discards poses where the robot would need to reorient the object multiple times before being able to place it in its final goal pose (for example, a pose where the grasping face is not pointing upwards but down towards the table would be discarded). Note that this filter is only applied for the physical experiments in Section IV-D, and that for other experiments, we consider the full set of poses and orientations. In future work, we can apply the solution proposed in [1] to address this problem in a more general way, or use other existing rearrangement methods [2], [3], [4] to achieve the goal pose determined by our system.

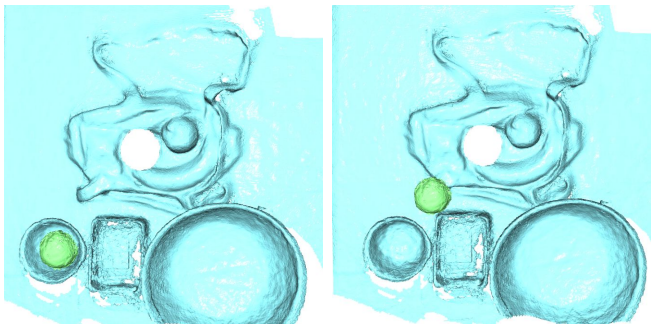

Fig. 1: The predicted goal pose from two different methods on the *"orange in bowl"* task. We visualize this using the foreground TSDF (in green) and the background TSDF (in blue). The left example shows a task success, since the center of mass of the orange is inside the bowl. The right example shows a task failure, since the center of mass of the orange is outside the bowl. However, if the task was to place the orange *beside* the bowl, then the left example would be a failure and the right example a success.

C. Task Success Definitions

To determine whether a predicted goal pose is correct, we visualize the TSDF of the movable object in that goal pose (as can be seen in Figure 1). We then use the task success definitions described below to determine whether this qualifies as a success.

Shopping scene. For the tasks consisting of placing an object inside a container (a bowl, a box or a basket), a predicted goal pose for the movable object constitutes a success if the center of mass of the movable object is within the bounds of the container (or above it). E.g. if the task is to place the apple inside the bowl, and the apple's center of mass is just inside the bowl, then this counts as a success. For the task of placing an object beside a bowl, this counts as a success if the object's center of mass is outside the bowl

* Joint first authorship. [1] The Robot Learning Lab at Imperial College London. [2] The Dyson Robotics Lab at Imperial College London.

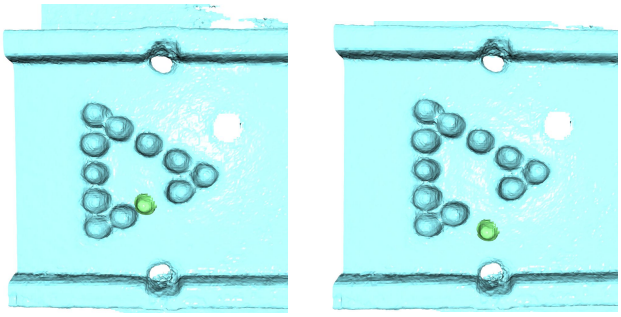

Fig. 2: The predicted goal pose from two different methods on the "pool balls in a triangle" task. The left example shows a task success, since the movable ball is within the bounds created by the neighboring balls. The right example shows a task failure, since the movable ball is outside those bounds.

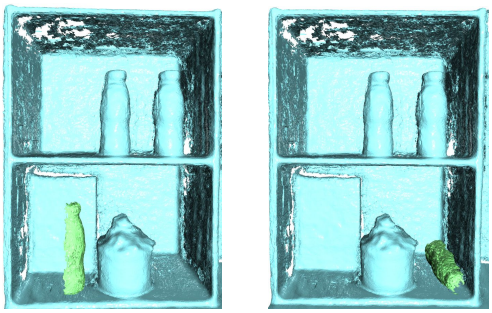

Fig. 3: The predicted goal pose from two different methods on the *"book"* task from the shelf scene. The left example shows a correct prediction of the position, since the bottle is in front of the book. It also shows a correct orientation prediction, since the bottle is standing upright with its cap at the top. Therefore, the predicted 6-DoF goal pose is correct here. In the right example, the predicted position is not correct since the bottle is not in front of the book. Furthermore, the predicted orientation is also not correct, because the bottle is lying on its side rather than standing upright.

but not as far from the bowl as the distance between the bowl and the basket (as that would be too far). Intuitively, the object must be outside the bowl but not too far away from the bowl in order to qualify as beside the bowl. Figure 1 visually illustrates the task success definitions. As always, we shuffle the objects in the scene in between each repeat. Then we run the method again (starting with collecting images of the new initial scene configuration). This allows us to test whether the methods are robust to different starting conditions.

Pool ball scene. For the triangle task, we take out one of the balls from either side of the triangle, and the method must place it back. In between repeats, we also shuffle some of the ball positions, and as always we redo the data collection scanning and run the pipeline again for the new repeat. Note that the missing ball is always the black ball, so that we can evaluate the same user instruction across many starting configurations of the scene. The success region within which the movable ball must be placed to achieve task success is defined by the two neighboring balls (near the optimum position, which is where the ball was originally before being taken out from the side of the shape). Specifically, along the x-axis (i.e. the left-right axis in Figure 2), the center of mass of the movable object must be below the bottom of the ball above it in the x-axis, and above the top of the ball below it. Along the y-axis (the top-bottom axis in the figure), the center of mass of the movable object must be between the centers of mass of those neighboring balls. An example can be seen in Figure 2. This same success region definition (using the neighboring balls) is also used in the *"X shape"* task. Here, we randomly take out one of the balls from near the X (except those at the very ends of the lines), and the methods must place it back to complete the X shape.

Shelf scene. For the task of placing bottles in a row, the movable object must be on the same shelf as the other bottles, and to the left of the leftmost other bottle (forming a row). For the book task, at least part of the bottle must be in front of the book. For the *"near plant"* task, the bottle can be on either side of the plant as long as it is nearby, but its center of mass cannot be in front of the left half of the book (since then it would be too far from the plant). In order for the full 6-DoF pose to qualify as correct for these tasks, the bottle must also be placed upright (i.e. with the bottle cap facing upwards). This is visualized in Figure 3. This means that the methods must perform (in imagination) 6-DoF reorientation, and determine the natural pose for this bottle zero-shot, without requiring any training example arrangements.

Physical execution. In Section IV-D of the main paper, we demonstrate how a robot might use the object goal pose determined by our method to perform pick-and-place. The success definition for predicting a 6-DoF goal pose is the same as the one described above for the shelf scene. The success definition for physically placing the bottle is that when the robot has placed and released it, it must be upright and in the correct position. Failures can occur e.g. when the bottle collides with the shelf while the robot is moving it and falls out of the gripper, so that it does not end up being placed successfully. We report the success rates for both goal pose prediction and physical execution in Section IV-D.

II. Failure Modes

In this section, we provide deeper insights into some failure modes of the current Dream2Real implementation which we observed in the experiments.

Heavily occluded containers. In the shopping scene, all methods and baselines achieved a lower success rate on the *"cookies in box"* task. By inspecting the score heatmaps, we also find that the higher-scoring poses of the cookie packet are often on the edge of the box. One possible explanation for this behavior is as follows: when the cookie packet is in the middle of the box, almost all of the box is occluded from the top-down view by the cookie packet. This makes it difficult for CLIP to identify the box in the rendered images,

and therefore to reliably evaluate whether the goal caption is satisfied. This difficulty is also exacerbated by the reflective surface of the metal box. It may be possible to mitigate this difficulty by rendering the scene from multiple views and aggregating CLIP scores across them, because then the box would be less occluded from other views, and so CLIP would be able to evaluate the arrangement more accurately.

Nearly symmetric geometry. Consider the pink bottle from the shelf scene. When flipped upside down, it has a similar geometry, due to its almost cylindrical shape. We find that CLIP sometimes gives high scores to arrangements where the three bottles are in a row but the pink bottle is upside down. This could be explained by the fact that the three bottles still form a symmetric row of three cylinders. Given this, CLIP is not able to distinguish very well whether the pink bottle is upside down or not, since it is nearly symmetric along this axis due to its cylindrical shape. We also note that the CLIP model we use has a 336x336 input resolution. Higher resolution models may be able to discern this detail more easily. Preliminary exploration with the GPT-4 vision encoder suggests that it may be promising for discriminating whether a bottle's orientation is correct in such cases.

Collisions with unobserved faces. When executing the rearrangement from the initial pose to the goal pose, the robot performs motion planning with collision avoidance, using the physics models constructed previously. We also check for collisions between the object being grasped and the environment during this motion planning. However, we do not have a complete physics model of the object, as the underside of the object cannot be observed. For example, in the shelf scene, the robot cannot easily see the part of the bottle which makes contact with the table. This means that during the motion planning, collisions between this side of the object and the environment may not be detected, and during execution this may lead to a collision. In future, this issue can be mitigated through shape completion methods.

Stability checking too permissive. When filtering out poses using physics checks, we use checks which are faster to run than fully-fledged dynamics simulation, but are less accurate: when in doubt, we err on the side of letting the VLM decide. However, this means that sometimes the VLM is shown physically strange arrangements which are very out-of-distribution (e.g. a bottle lying sideways on top of a plant), and so it might give erroneous outputs. This can be fixed using more thorough physics checking and also a stronger VLM.

REFERENCES

[1] K. Wada, S. James, and A. J. Davison, "ReorientBot: Learning object reorientation for specific-posed placement," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

[2] A. Goyal, A. Mousavian, C. Paxton, Y.-W. Chao, B. Okorn, J. Deng, and D. Fox, "IFOR: Iterative flow minimization for robotic object rearrangement," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[3] A. Murali, A. Mousavian, C. Eppner, A. Fishman, and D. Fox, "CabiNet: Scaling neural collision detection for object rearrangement with procedural scene generation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2023.

[4] J. Urain, N. Funk, J. Peters, and G. Chalvatzaki, "SE(3)-DiffusionFields: Learning smooth cost functions for joint grasp and motion optimization through diffusion," *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.