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

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

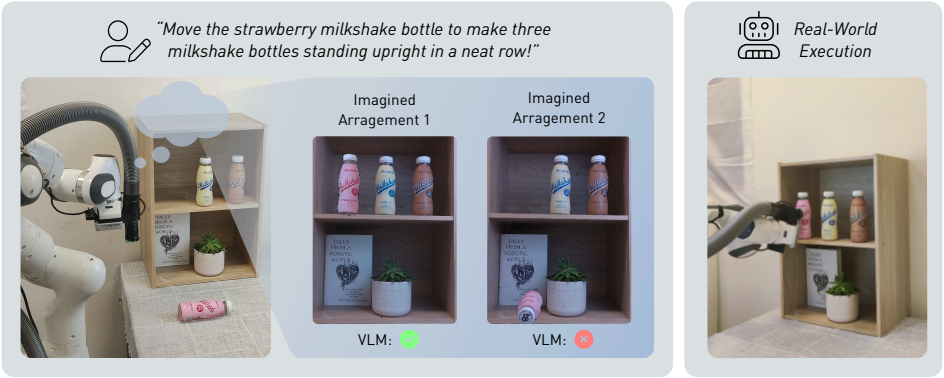



Fig. 1: Dream2Real enables a robot to imagine, and then evaluate, virtual rearrangements of scenes. First, the robot builds an object-centric NeRF of a scene. Then, numerous reconfigurations of the scene are rendered as 2D images. Finally, a VLM evaluates these according to the user instruction, and the best is then physically created using pick-and-place.

***Abstract*— We introduce Dream2Real, a robotics framework which integrates vision-language models (VLMs) trained on 2D data into a 3D object rearrangement pipeline. This is achieved by the robot autonomously constructing a 3D representation of the scene, where objects can be rearranged virtually and an image of the resulting arrangement rendered. These renders are evaluated by a VLM, so that the arrangement which best satisfies the user instruction is selected and recreated in the real world with pick-and-place. This enables language-conditioned rearrangement to be performed zero-shot, without needing to collect a training dataset of example arrangements. Results on a series of real-world tasks show that this framework is robust to distractors, controllable by language, capable of understanding complex multi-object relations, and readily applicable to both tabletop and 6-DoF rearrangement tasks. Videos are available on our webpage at: https://www.robot-learning.uk/dream2real.**

## I. Introduction

Consider being asked to perform a task such as arranging bottles in a row, as in Figure 1. To achieve this, humans might first imagine the goal state that should be created according to the instructions. This imagined arrangement should be physically valid, visually natural (e.g. the bottles are not upside down), and semantically correct for the given task. In this paper, we study how robots can imagine (or *dream*) new configurations of scenes, and then evaluate them to select a suitable goal state. This leads to our language-conditioned 6-DoF object rearrangement framework, Dream2Real.

Recently, vision-language models (VLMs) such as CLIP [1] have enabled robots to connect language instructions with the scene before them [2], [3], enabling generalisation across many objects and tasks. By training on hundreds of millions of captioned images from the Web (including images of object arrangements), CLIP learns to predict how closely an image matches a text description. This is exactly the reasoning a robot requires when evaluating novel scene arrangements it has imagined with respect to a user's language instruction.

Our approach is summarised in Figure 1. We address several difficult technical challenges, including autonomously building a 3D NeRF-based [4] object-level representation of the scene which can be rearranged in imagination, and interfacing this with 2D VLMs to evaluate imagined arrangements. Experiments show that Dream2Real outperforms other recent work on VLMs for tabletop rearrangement [5], and demonstrate that our framework is robust to distractors, can evaluate complex many-object spatial relations, and is readily applicable to 3D scenes.

Integrating a VLM and a NeRF-based representation with editable poses in this novel way yields several strengths. First, Dream2Real is **zero-shot**, as it applies VLMs to object rearrangement without requiring a training dataset of example arrangements to be collected. Second, it achieves full **6-DoF rearrangement**, whereas prior zero-shot work [5] is limited to top-down scenes. Third, we show that the use of VLMs for **visual evaluation** of imagined goal states is better than asking VLMs to predict the goal state directly. Prior work on rearrangement typically requires collecting thousands of example arrangements [6], [7], [8], [9] (for more on related work, see Appendix I). Dream2Real is the first method which performs 6-DoF rearrangement zero-shot by using the web-scale visual reasoning of VLMs.

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

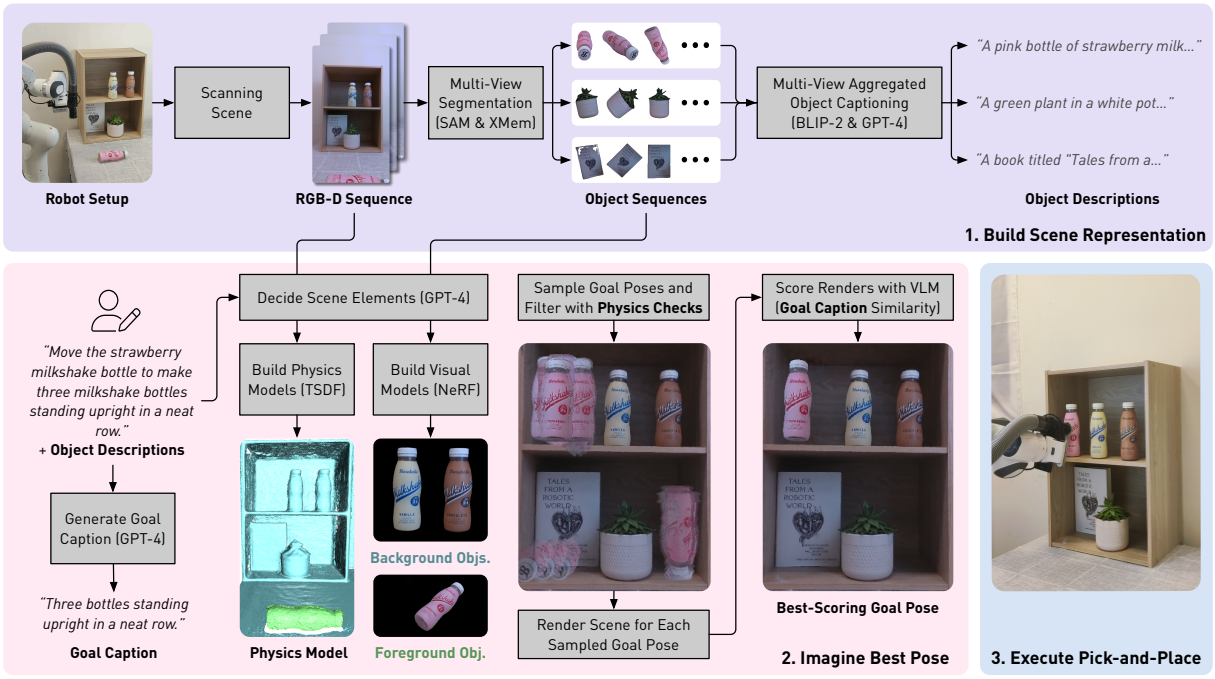



Fig. 2: The Dream2Real pipeline. The robot first autonomously builds a model of the scene. Then the user instruction is used to determine which object should be moved, and so the robot can imagine new configurations of the scene and score them using a VLM. Finally, the highest-scoring pose is used as the goal for pick-and-place to complete the rearrangement.

## II. Method Overview

The core problem we address is determining a goal pose for an object given a language instruction. For this we propose a modular framework shown in Figure 2. In this section we give an overview of how all the components fit together, and we refer to Appendix II for further details.

Before the user instruction is received, the robot constructs an object-level scene representation (detailed in Appendix II-A). This includes collecting a set of RGBD images, segmenting the first image into object masks with SAM [10], tracking these object masks across subsequent images with XMem [11], getting a caption for each object from each view using BLIP-2 [12], and aggregating those captions across views into a coherent object description using a language model (GPT-4 [13]).

When the user instruction is received, a language model is used to process this instruction (Appendix II-B) and determine which object should be moved, as well as which other objects in the scene are relevant to the instruction. This ensures that distractor objects are not shown to the VLM. The language model also outputs the goal caption and normalizing caption, which will be used later by the VLM. Once we have determined the foreground (the object to be moved, which we call the "movable object") and background (other relevant objects) for this task, we can construct visual and physics models for the foreground and background (Appendix II-C). We use NeRF-based Instant-NGP [14], [4] for visual models and TSDF [15] for physics models. Then, these models are used to determine the best pose for the movable object (Appendix II-D). This is done by first sampling many poses and filtering out those that are physically invalid (e.g. if the movable object's pose is in collision or unsupported) using the physics models. The movable object is then moved to each valid pose and the scene is rendered. Next, each render is scored (i.e. evaluated) by the VLM (we use CLIP [1]) by comparing the similarity of the image's embedding with the text embedding of the goal caption. The pose with the highest-scoring render is then used as the goal pose. Finally, the robot moves the object from its initial pose to the goal pose using motion planning with collision avoidance (Appendix II-E).

Fig. 3: The shopping, pool ball, and shelf scenes.

## III. Experiments

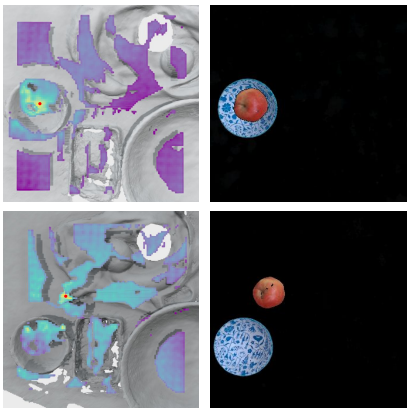
We evaluate on 10 rearrangement tasks across 3 real-world scenes, shown in Figure 3. We compare against DALL-E-Bot [5], since it is also a zero-shot method using VLMs, as well as several ablations of our method. Further experiment details such as baseline descriptions are in Appendix III.

### A. Zero-Shot Multi-Task Rearrangement

First we evaluate on a scene we refer to as the shopping scene, where many tasks are possible. We choose a top-down

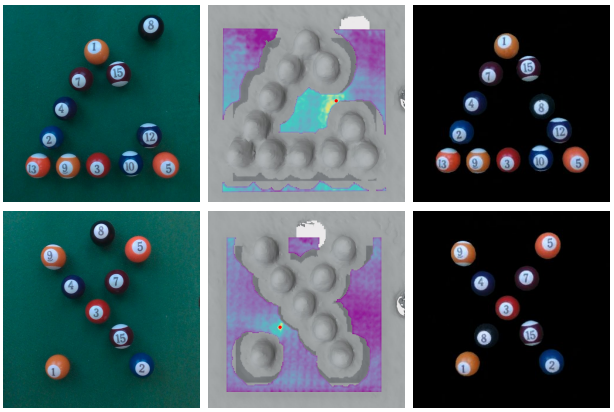
scene to allow a comparison against DALL-E-Bot [5]. The 5 instructions (i.e. 5 tasks) for this scene are: (1) *"put the apple inside the blue and white bowl"*, (2) *"put the apple beside the blue and white bowl"*, (3) *"put the cookies inside the square metal box"*, (4) *"put the orange inside the blue and white bowl"*, and (5) *"put the banana inside the wicker basket"*. We sample object positions but not orientations here. We run 7 repeats for each method-task combination, for a total of 280 goal pose predictions (in imagination). In between repeats, we shuffle object positions and re-scan the scene.

Qualitative results are in Figure 4, showing a heatmap of CLIP scores next to the best-scoring render. Table I shows quantitative results. Our method significantly outperforms *DALL-E-Bot* [5] (83% vs 34% mean success rate). This is due to a key difference in how our approaches use VLMs: DALL-E-Bot is *predictive*, i.e. it generates a goal image and attempts to match those objects to the real world. However, DALL-E-Bot very often generates images with a different number of objects to the real world, and so (despite its filtering techniques) it matches the real object to a generated object in the wrong place. Dream2Real is *evaluative*, using a VLM to score sampled arrangements of the real objects, thus avoiding this difficult matching problem. DALL-E-Bot is also affected by distractors, whereas our method automatically hides them from the VLM. *D2R-Vis-Prior*'s lower performance suggests that conditioning the visual prior on language is important. Our method also doubles the success rate of *D2R-No-Norm*, showing that normalising captions are effective for these tasks in forcing CLIP to focus on the spatial relations in the instruction. *D2R-No-Smooth*'s high performance shows that CLIP works well here even without outlier filtering. The *D2R-Distract* ablation shows that our technique of only showing relevant objects to the VLM is crucial for performance on cluttered scenes. This experiment shows that Dream2Real can succeed at everyday rearrangement tasks zero-shot.



(a) CLIP score heatmap (b) Max score render

Fig. 4: Qualitative results from the shopping scene for the tasks "apple in bowl" (top row) and "apple beside bowl" (bottom row). Figure 3 shows the full shopping scene. In the heatmaps (overlaid on the TSDF of the scene), yellow indicates high-scoring positions of the apple, whereas dark blue indicates low-scoring regions, and colliding poses are not included. The red dot highlights the highest-scoring position. The highest-scoring render is shown on the right.



(a) Initial scene (b) Heatmap (c) Max score render

Fig. 5: Qualitative results from the pool ball scene for the tasks *"in triangle"* (top row) and *"in X shape"* (bottom row). The red dot is used to highlight the high-scoring area.

### B. Multi-Object Geometric Relations

In this scene, we test our method on geometric relations involving many objects: the method must form a triangle out of 12 pool balls, and an X shape out of 9, by placing the final black ball in the correct position (in imagination). Positions are sampled at a 1mm resolution. In between the 5 roll-outs per method-task combination, we randomly take out a ball from the shape, and the method must complete the shape by placing the black ball. Results are shown in Fig 5. The heatmap shows a high-scoring mode near the optimal pose for each task, suggesting that CLIP can understand geometric relations involving many objects. Success rates are in Table II. DALL-E-Bot often fails due to the matching problem as before, which our evaluative approach avoids. *Physics-Only*'s low success rate shows that using CLIP for semantic guidance is useful. Interestingly, *D2R-Distract* performs well because there are no distractors here, and the green pool table background seems to provide helpful context to CLIP.

### C. 6-DoF Rearrangement in a 3D Scene

Here we test our method on a 3D shelf scene (see Figure 3). Our method must perform 6-DoF rearrangement (in imagination) to pick up the bottle lying on the table and position it upright on the shelf. There are 3 tasks: making the bottles into a row, placing the bottle in front of the book, and placing the bottle near the plant. In this scene, we sample 24 orientations at each position (i.e. discretise coarsely into $\pi/2$ orientations around each of the coordinate axes). In between each of the 10 roll-outs per method-task combination, we move and rotate the bottle around the table and shuffle some of the objects on the shelf. The heatmaps for each task are in Figure 6. We compare several interesting variations

TABLE I: Success rates for the shopping scene (%).

| Method | *apple in bowl* | *apple beside bowl* | *orange in bowl* | *cookies in box* | *banana in basket* | *mean* |
|---|---|---|---|---|---|---|
| Physics-Only | 0 | 57 | 14 | 0 | 14 | 17 |
| D2R-Distract | 0 | 71 | 14 | 0 | 0 | 17 |
| D2R-Vis-Prior | 0 | 71 | 14 | 0 | 0 | 17 |
| DALL-E-Bot [5] | 14 | 29 | 0 | **43** | 86 | 34 |
| D2R-No-Norm | 29 | 71 | 71 | 0 | 29 | 40 |
| D2R-One-View | 71 | 14 | 57 | 29 | **100** | 54 |
| Dream2Real | **100** | 71 | **100** | **43** | **100** | 83 |
| D2R-No-Smooth | **100** | **86** | **100** | **43** | **100** | **86** |

TABLE II: Success rates for the pool ball scene (%).

| Method | *in X shape* | *in triangle* | *mean* |
|---|---|---|---|
| D2R-Vis-Prior | 0 | 0 | 0 |
| Physics-Only | 20 | 0 | 10 |
| DALL-E-Bot [5] | 0 | 60 | 30 |
| D2R-No-Norm | 80 | 40 | 60 |
| D2R-One-View | 20 | **100** | 60 |
| D2R-No-Smooth | 80 | 80 | 80 |
| Dream2Real | **100** | 80 | 90 |
| D2R-Distract | **100** | **100** | **100** |

of our method in Figure 7. *Physics-Only* rarely guesses the semantically correct upright orientation, showing that this is a challenging problem which our method addresses. Interestingly, the *D2R-One-View* baseline often fails due to incorrect object identification, whereas our approach which integrates captions across views is more robust. This shows that our multi-view approach is better suited to 6-DoF scenes.

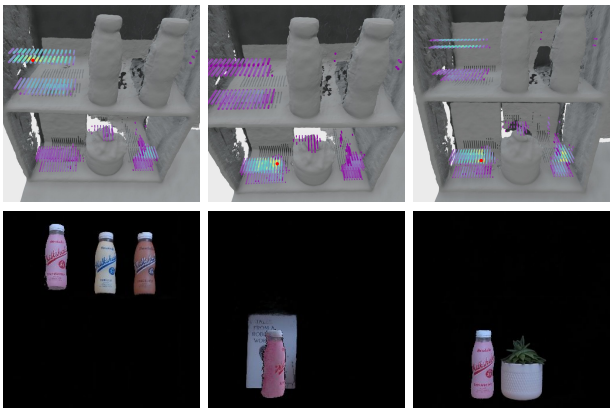
### D. Demonstrating Physical Execution

Although the main contribution of our paper is predicting the goal pose, here we also demonstrate how a robot can pick and place objects into those goal poses. Robot videos for all scenes are available on our website: https://www.robot-learning.uk/dream2real. In Figure 8, we compare our multi-view method with *D2R-One-View* on the 6-DoF tasks in the shelf scene with robotic execution. Since we now automatically eliminate unreachable poses, results will differ from Figure 7. We find that the single-view baseline fails more often due to incomplete reconstruction, which impacts both goal pose prediction and collision-free motion planning. This shows that our multi-view Dream2Real framework can be used to perform 6-DoF rearrangement on a real robot.

## IV. Discussion

**Limitations**. Please see Appendix IV for in-depth analysis on limitations, failure modes, and future work.

**Conclusions**. We show for the first time how 2D VLMs can be used to perform language-conditioned 3D object rearrangement zero-shot, without needing to collect any example arrangements. Encouraging results show that our method Dream2Real can complete everyday rearrangement tasks, understands complex multi-object relations, and is robust to distractors. While prior work [5] uses VLMs to *generate* goal images, our approach uses VLMs to *evaluate* candidate images. This improves performance in tabletop experiments and enables the use of 2D VLMs for 3D tasks.



(a) *"bottles in a row"* (b) *"in front of book"* (c) *"near plant"*

Fig. 6: Results for the three tasks on the shelf scene, with heatmaps (top row) and the highest-scoring renders (bottom).

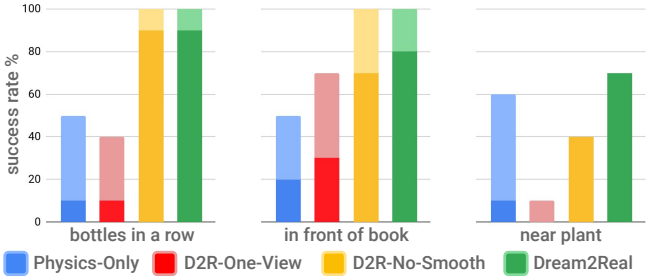



Fig. 7: Success rates for the shelf scene. A darker bar shows the success rate for predicting the full 6-DoF pose, and a lighter bar on top indicates roll-outs where the method correctly predicted the position but not the orientation.


Fig. 8: Success rates for robotic execution. A darker bar shows the success rate for placing the object, and a lighter bar on top indicates roll-outs where the method correctly predicted the 6-DoF pose but did not execute successfully.

## ACKNOWLEDGEMENTS

We thank our colleagues from the Robot Learning Lab and the Dyson Robotics Lab for helpful discussions. In particular, we would like to thank Andrew Davison, Eric Dexheimer, Xin Kong, Hide Matsuki, Marwan Taher, Vitalis Vosylius, and Kentaro Wada. In addition, we are grateful to Shikun Liu for lending his expertise in diagram design. This research was supported by: Dyson Technology Ltd, EPSRC Prosperity Partnerships (EP/S036636/1), and the Royal Academy of Engineering under the Research Fellowship Scheme. For the purpose of open access, the authors have applied a Creative Commons Attribution (CC BY) license to any Accepted Manuscript version arising.

## REFERENCES

[1] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning, ICML*, 2021.

[2] M. Shridhar, L. Manuelli, and D. Fox, "CLIPort: What and where pathways for robotic manipulation," in *Conference on Robot Learning (CoRL)*, 2021.

[3] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, P. Florence, C. Fu, M. G. Arenas, K. Gopalakrishnan, K. Han, K. Hausman, A. Herzog, J. Hsu, B. Ichter, A. Irpan, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, L. Lee, T.-W. E. Lee, S. Levine, Y. Lu, H. Michalewski, I. Mordatch, K. Pertsch, K. Rao, K. Reymann, M. Ryoo, G. Salazar, P. Sanketi, P. Sermanet, J. Singh, A. Singh, R. Soricut, H. Tran, V. Vanhoucke, Q. Vuong, A. Wahid, S. Welker, P. Wohlhart, J. Wu, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich, "RT-2: Vision-language-action models transfer web knowledge to robotic control," in *arXiv*, 2023.

[4] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics (ToG)*, vol. 41, no. 4, pp. 1–15, 2022.

[5] I. Kapelyukh, V. Vosylius, and E. Johns, "DALL-E-Bot: Introducing web-scale diffusion models to robotics," *IEEE Robotics and Automation Letters (RA-L)*, 2023.

[6] W. Liu, C. Paxton, T. Hermans, and D. Fox, "StructFormer: Learning spatial structure for language-guided semantic rearrangement of novel objects," *International Conference on Robotics and Automation*, 2022.

[7] W. Liu, Y. Du, T. Hermans, S. Chernova, and C. Paxton, "StructDiffusion: Language-guided creation of physically-valid structures using unseen objects," in *RSS*, 2023.

[8] 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.

[9] A. Simeonov, A. Goyal, L. Manuelli, L. Yen-Chen, A. Sarmiento, A. Rodriguez, P. Agrawal, and D. Fox, "Shelving, stacking, hanging: Relational pose diffusion for multi-modal rearrangement," *arXiv preprint arXiv:2307.04751*, 2023.

[10] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, *et al.*, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.

[11] H. K. Cheng and A. G. Schwing, "Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model," in *European Conference on Computer Vision*. Springer, 2022, pp. 640–658.

[12] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," 2023.

[13] OpenAI, "GPT-4 technical report," 2023.

[14] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[15] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," *arXiv:1801.09847*, 2018.

[16] D. Batra, A. X. Chang, S. Chernova, A. J. Davison, J. Deng, V. Koltun, S. Levine, J. Malik, I. Mordatch, R. Mottaghi, M. Savva, and H. Su, "Rearrangement: A challenge for embodied AI," *arXiv*, 2020.

[17] M. J. Schuster, D. Jain, M. Tenorth, and M. Beetz, "Learning organizational principles in human environments," in *International Conference on Robotics and Automation*, 2012, pp. 3867–3874.

[18] G. Sarch, Z. Fang, A. W. Harley, P. Schydlo, M. J. Tarr, S. Gupta, and K. Fragkiadaki, "TIDEE: Tidying up novel rooms using visuo-semantic commonsense priors," in *European Conference on Computer Vision*, 2022.

[19] K. Ramachandruni, M. Zuo, and S. Chernova, "Consor: A context-aware semantic object rearrangement framework for partially arranged scenes," in *2023 IEEE International Conference on Intelligent Robots and Systems*, 2023.

[20] Y. Zeng, M. Wu, L. Yang, J. Zhang, H. Ding, H. Cheng, and H. Dong, "Distilling functional rearrangement priors from large models," 2023.

[21] Y. Lin, A. S. Wang, E. Undersander, and A. Rai, "Efficient and interpretable robot manipulation with graph neural networks," *IEEE Robotics and Automation Letters*, vol. 7, pp. 2740–2747, 2022.

[22] G. Zhai, X. Cai, D. Huang, Y. Di, F. Manhardt, F. Tombari, N. Navab, and B. Busam, "SG-Bot: Object rearrangement via coarse-to-fine robotic imagination on scene graphs," in *ICRA*, 2024.

[23] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani, and J. Lee, "Transporter networks: Rearranging the visual world for robotic manipulation," *Conference on Robot Learning (CoRL)*, 2020.

[24] C. Paxton, C. Xie, T. Hermans, and D. Fox, "Predicting stable configurations for semantic placement of novel objects," in *Conference on Robot Learning, 8-11 November 2021, London, UK*, ser. Proceedings of Machine Learning Research, vol. 164. PMLR, 2021, pp. 806–815.

[25] N. Gkanatsios, A. Jain, Z. Xian, Y. Zhang, C. G. Atkeson, and K. Fragkiadaki, "Energy-based models are zero-shot planners for compositional scene rearrangement," *Robotics: Science and Systems XIX*, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:258352334

[26] N. Abdo, C. Stachniss, L. Spinello, and W. Burgard, "Robot, organize my shelves! Tidying up objects by predicting user preferences," in *International Conference on Robotics and Automation*, 2015.

[27] I. Kapelyukh and E. Johns, "My house, my rules: Learning tidying preferences with graph neural networks," in *Conference on Robot Learning (CoRL)*, 2021.

[28] V. Jain, Y. Lin, E. Undersander, Y. Bisk, and A. Rai, "Transformers are adaptable task planners," in *Conference on Robot Learning*, 2022.

[29] I. Kapelyukh and E. Johns, "SceneScore: Learning a cost function for object arrangement," in *CoRL 2023 Workshop on Learning Effective Abstractions for Planning (LEAP)*, 2023.

[30] Y. Kant, A. Ramachandran, S. Yenamandra, I. Gilitschenski, D. Batra, A. Szot, and H. Agrawal, "Housekeep: Tidying virtual households using commonsense reasoning," *arXiv*, 2022.

[31] Y. Ding, X. Zhang, C. Paxton, and S. Zhang, "Task and motion planning with large language models for object rearrangement," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2023.

[32] J. Wu, R. Antonova, A. Kan, M. Lepert, A. Zeng, S. Song, J. Bohg, S. Rusinkiewicz, and T. Funkhouser, "Tidybot: Personalized robot assistance with large language models," *Autonomous Robots*, 2023.

[33] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, A. Herzog, D. Ho, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, E. Jang, R. J. Ruano, K. Jeffrey, S. Jesmonth, N. J. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, K.-H. Lee, S. Levine, Y. Lu, L. Luu, C. Parada, P. Pastor, J. Quiambao, K. Rao, J. Rettinghouse, D. Reyes, P. Sermanet, N. Sievers, C. Tan, A. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, S. Xu, M. Yan, and A. Zeng, "Do as I can, not as I say: Grounding language in robotic affordances," *arXiv*, 2022.

[34] A. Stone, T. Xiao, Y. Lu, K. Gopalakrishnan, K.-H. Lee, Q. Vuong, P. Wohlhart, S. Kirmani, B. Zitkovich, F. Xia, C. Finn, and K. Hausman, "Open-world object manipulation using pre-trained vision-language model," in *arXiv preprint*, 2023.

[35] T. Yu, T. Xiao, A. Stone, J. Tompson, A. Brohan, S. Wang, J. Singh, C. Tan, D. M, J. Peralta, B. Ichter, K. Hausman, and F. Xia, "Scaling robot learning with semantically imagined experience," *arXiv*, 2023.

[36] Z. Mandi, H. Bharadhwaj, V. Moens, S. Song, A. Rajeswaran, and V. Kumar, "CACTI: A framework for scalable multi-task multi-scene visual imitation learning," *arXiv*, 2022.

[37] Z. Chen, S. Kiami, A. Gupta, and V. Kumar, "GenAug: Retargeting behaviors to unseen situations via generative augmentation," *arXiv*, 2023.

[38] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, "Voxposer: Composable 3d value maps for robotic manipulation with language models," in *Conference on Robot Learning (CoRL)*, 2023.

[39] W. Shen, G. Yang, A. Yu, J. Wong, L. P. Kaelbling, and P. Isola, "Distilled feature fields enable few-shot language-guided manipulation," *arXiv preprint:2308.07931*, 2023.

[40] W. Yu, N. Gileadi, C. Fu, S. Kirmani, K.-H. Lee, M. Gonzalez Arenas, H.-T. Lewis Chiang, T. Erez, L. Hasenclever, J. Humplik, B. Ichter, T. Xiao, P. Xu, A. Zeng, T. Zhang, N. Heess, D. Sadigh, J. Tan, Y. Tassa, and F. Xia, "Language to rewards for robotic skill synthesis," *Arxiv preprint arXiv:2306.08647*, 2023.

[41] Y. J. Ma, W. Liang, V. Som, V. Kumar, A. Zhang, O. Bastani, and D. Jayaraman, "Liv: Language-image representations and rewards for robotic control," *arXiv preprint arXiv:2306.00958*, 2023.

[42] Y. Cui, S. Niekum, A. Gupta, V. Kumar, and A. Rajeswaran, "Can foundation models perform zero-shot task specification for robot manipulation?" in *Proceedings of The 4th Annual Learning for Dynamics and Control Conference*, ser. Proceedings of Machine Learning Research, R. Firoozi, N. Mehr, E. Yel, R. Antonova, J. Bohg, M. Schwager, and M. Kochenderfer, Eds., vol. 168. PMLR, 23–24 Jun 2022, pp. 893–905.

[43] S. Sharma, A. Rashid, C. M. Kim, J. Kerr, L. Y. Chen, A. Kanazawa, and K. Goldberg, "Language embedded radiance fields for zero-shot task-oriented grasping," in *7th Annual Conference on Robot Learning*, 2023.

[44] L. Yen-Chen, P. Florence, A. Zeng, J. T. Barron, Y. Du, W.-C. Ma, A. Simeonov, A. R. Garcia, and P. Isola, "MIRA: Mental imagery for robotic affordances," in *Conference on Robot Learning (CoRL)*, 2022.

[45] D. Driess, Z. Huang, Y. Li, R. Tedrake, and M. Toussaint, "Learning multi-object dynamics with compositional neural radiance fields," in *Proceedings of The 6th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 205. PMLR, 14–18 Dec 2023, pp. 1755–1768.

[46] X. Kong, S. Liu, M. Taher, and A. J. Davison, "vMAP: Vectorised object mapping for neural field slam," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 952–961.

[47] C. Wang, M. Chai, M. He, D. Chen, and J. Liao, "Clip-nerf: Text-and-image driven manipulation of neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3835–3844.

[48] A. Mirzaei, Y. Kant, J. Kelly, and I. Gilitschenski, "Laterf: Label and text driven object radiance fields," in *European Conference on Computer Vision*. Springer, 2022, pp. 20–36.

[49] H. Ha and S. Song, "Semantic abstraction: Open-world 3D scene understanding from 2D vision-language models," in *Proceedings of the 2022 Conference on Robot Learning*, 2022.

[50] K. Jatavallabhula, A. Kuwajerwala, Q. Gu, M. Omama, T. Chen, S. Li, G. Iyer, S. Saryazdi, N. Keetha, A. Tewari, J. Tenenbaum, C. de Melo, M. Krishna, L. Paull, F. Shkurti, and A. Torralba, "Conceptfusion: Open-set multimodal 3d mapping," *Robotics: Science and Systems*, 2023.

[51] N. M. M. Shafiullah, C. Paxton, L. Pinto, S. Chintala, and A. Szlam, "Clip-fields: Weakly supervised semantic fields for robotic memory," in *Robotics: Science and Systems*, 2023.

[52] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996, pp. 303–312.

[53] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," *arXiv:1801.09847*, 2018.

[54] V. Satish, J. Mahler, and K. Goldberg, "On-policy dataset synthesis for learning robot grasping policies using fully convolutional deep networks," *IEEE Robotics and Automation Letters*, 2019.

[55] 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.

[56] 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.

[57] J. Kuffner and S. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 2, 2000, pp. 995–1001 vol.2.

[58] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, "pixelNeRF: Neural radiance fields from one or few images," in *CVPR*, 2021.

[59] L. Melas-Kyriazi, C. Rupprecht, I. Laina, and A. Vedaldi, "Realfusion: 360° reconstruction of any object from a single image," in *Arxiv*, 2023.

[60] R. Liu, R. Wu, B. V. Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick, "Zero-1-to-3: Zero-shot one image to 3D object," in *ICCV*, 2023.

[61] M. Yuksekgonul, F. Bianchi, P. Kalluri, D. Jurafsky, and J. Zou, "When and why vision-language models behave like bags-of-words, and what to do about it?" in *International Conference on Learning Representations*, 2023.

## APPENDIX I
## RELATED WORK

**Predicting goal poses** is a key challenge in object rearrangement [16]. Prior work classifies the correct receptacle in which to place an object [17], [18], [19], or predicts a continuous goal pose [20], [21], [22]. Another approach learns rearrangement from full demonstrations [23], [2]. The prediction of goal poses can be conditioned in several ways. Some methods allow users to specify relational predicates [24], [25]. Others learn a personalised representation of user preferences [26], [27], [28]. User instructions may also be expressed in free-form language. StructFormer [6] trains a language-conditioned transformer on a synthetic dataset of over 100,000 rearrangement sequences. To better avoid collisions, StructDiffusion [7] learns a distribution over desirable poses using a diffusion model. SceneScore [29] uses an energy-based model, thus learning to evaluate whether a given arrangement is desirable. These methods typically require training on thousands of example arrangements [6], [8], [9]. This is effective for specific tasks, but is difficult to scale to unstructured environments such as homes, because of the difficulty of generating realistic training data.

Instead, **large language models** (LLMs) pre-trained on web-scale data can be used to predict arrangements [30], [31], [32]. These are effective for high-level planning, but language models typically lack the visual perception needed to e.g. assess whether an object is oriented correctly. Our framework uses VLMs to *visually* evaluate imagined scenes. The closest work to ours is DALL-E-Bot [5], which achieves visual rearrangement zero-shot by using a web-scale diffusion model to generate a goal image, and then matching that to the real scene. However, this is limited to top-down scenes. Experiments show that our evaluative approach is more robust than predicting a goal state directly.

Beyond rearrangement, **vision-language models** and LLMs have proven to be useful for bringing web-scale semantic understanding to embodied agents [3], [33], [34], [35], [36], [37], [38]. VLMs in particular can also be used to connect user instructions in natural language with the robot's visual perception [2], [39]. Closer to our work, VLMs and LLMs can also act as reward signals to train robot policies [40], [41], [42]. In this work, we show how the web-scale visual prior of VLMs can solve 3D rearrangement tasks zero-shot without any further policy training.

**3D reconstruction** research continues to yield useful techniques for robotics [39], [43], [44], [45]. Implicit neural representations such as NeRF [14] have shown a strong ability to produce photorealistic renders from novel views. Instant-NGP [4] significantly accelerates the training and rendering speed of NeRF using multiresolution hash encoding, enabling real-time rendering. Scenes can also be decomposed with object-level reconstruction, as shown in works like vMAP [46]. Related to our method, some works [47], [48], [49], [50], [51] also combine 3D representations with LLMs, but do not focus on zero-shot 3D robotic rearrangement.

## APPENDIX II
## METHOD DETAILS

### A. Observing the Scene

In our framework, the robot constructs as much of its scene representation as possible before a user instruction is received (i.e. when a robot first observes a scene), thus reducing the instruction execution delay. Therefore, the robot starts by autonomously scanning the scene to collect a set of RGB-D images which will be used later for building the object-level visual and physics models. In our experiments, we use a hemispherical camera trajectory facing the scene centre. We segment the scene into objects by running Segment Anything (SAM) [10] on the first frame from our trajectory where all objects are assumed to be at least partially visible. Those objects are tracked across the other views using XMem [11], handling the data association problem across frames. Given the tracked objects, we extract image crops from each of the views in which they are visible and apply BLIP-2 [12] to retrieve a per-crop caption. Since captions for the same object may differ across views, we use an LLM (GPT-4) [13] to aggregate these into one coherent object description. We find that this multi-view aggregation allows objects to be identified more reliably. An example object description produced by the language model is: *"A pink bottle of strawberry milk or juice with a red label, white cap, and barcode on it, sitting on a table or white surface."*

### B. Interpreting User Instructions

Once the user instruction is received, the robot must process it to understand the task. We automatically extract four key items from the user instruction using a language model (GPT-4): the *movable object*, *relevant objects*, the *goal caption* and the *normalising caption*. E.g. suppose that the user instruction is *"put the apple inside the bowl"*. The movable object here is the apple, since it is the one which should be moved to fulfil the instruction. Relevant objects are those which the VLM should observe to evaluate whether the user instruction is fulfilled (apple and bowl). This technique avoids showing distractor (irrelevant) objects to CLIP, which we show is crucial for performance. The goal caption is a description of the desired final state after the instruction has been fulfilled. In this example, it would be: *"an apple inside a bowl"*. Lastly, the normalising caption is a description of the scene that remains neutral to the pose of the object being moved. Typically, GPT-4 simply returns a list of objects within the scene, e.g. *"an apple and a bowl"*. This will be used later for normalising CLIP scores (Section II-D).

### C. Building Task-Specific Visual & Physics Models

We now describe how to construct the physics models which we use to check imagined arrangements for collisions, and the visual models that we use for rendering those arrangements. We separate the scene into the foreground (the movable object) and the background (relevant objects excluding the movable object), then build two separate visual models accordingly. We use NeRF (specifically Instant-NGP [4]) because of its high visual realism and speed for both

training and rendering. In detail, for both foreground and background objects, using masks from XMem, we assign pixels outside of the corresponding masks 0 alpha value. During NeRF training, this encourages the space around the object to be represented as empty, which will later allow us to freely move this object around the scene and render it from novel poses. Since we move the entire foreground NeRF, this empty space supervision is important to allow the two NeRFs to be rendered together correctly. To build the physics models, we combine depth images from across views to create a separate foreground and background Truncated Signed Distance Function (TSDF) [52], [53], which we find achieves more accurate geometry than extracting a mesh from Instant-NGP.

### D. Dreaming the Best Pose

Now that we have a separate, movable model for the foreground object, we can sample many different poses for it and evaluate each of these "imaginary" arrangements, to find a desirable pose. We find experimentally that a straightforward sampling strategy where we sample positions in a dense, regular 3D grid covering the scene (and sample orientations from discretised bins) works well. We move (virtually) the movable object's physics model to each of the sampled poses in turn and check for physical validity, i.e. the object must not be in collision with the scene or unsupported in free space. Thus we avoid rendering and evaluating invalid poses. Then, for each valid pose, we render the foreground NeRF as if the object were in that pose, and combine this with the background NeRF render (using a similar approach to [46]). We render the NeRFs from a fixed camera pose from our scanning trajectory facing the centre of the scene.

We now have a rendered RGB image for each sampled goal state, which can be evaluated with a web-scale VLM. We batch-compute the CLIP similarity [1] between the image of each arrangement and the goal caption. We also divide this similarity score by the similarity of the image with the normalising caption. Intuitively, we want the overall similarity score to focus only on whether the spatial relation requested by the user is satisfied or not in the image. We show experimentally that this is important for performance. We also implement *spatial smoothing*: a Gaussian smoothing filter is applied on the 3D grid of scores to reduce the score of outlier poses, which have a high score but are surrounded by many low-scoring poses. Finally, we select the highest-scoring sampled pose as the goal pose for the movable object.

### E. Robot Execution

Once the goal pose has been determined, the robot executes the rearrangement using pick-and-place. For our grasping module we use the FC-GQCNN from DexNet 4.0 [54], but any off-the-shelf grasping method can easily be applied [55], [56]. We then use inverse kinematics and a motion planner (RRT-Connect [57]) to find a path between the pick and place poses which avoids collisions, using the object collision meshes that the robot constructed previously.

## Appendix III
## Experiment Details

### A. Hardware Setup

For real-world evaluation, we instantiate our framework on a 7-DoF Franka Panda with a wrist-mounted Intel RealSense D435i RGB-D camera and a compliant suction gripper for physically performing the evaluation.

### B. Evaluation Metric

As the primary contribution of this paper is a method for determining a goal pose, our evaluation focuses on whether the predicted goal pose is correct. We measure task success using success regions. This allows us to efficiently and fairly evaluate many variations of our method, by controlling for noise that would arise from physical execution. Physical execution is evaluated as part of the whole pipeline in Section III-D. Further detail (e.g. on success regions) is provided in the supplementary material document, available on our webpage: https://www.robot-learning.uk/dream2real

### C. Baseline Descriptions

Here we list the 7 baselines that we compare with our main method (Dream2Real) in the experiments. (1) Since our method is zero-shot, it cannot be compared fairly against methods which require thousands of example arrangements to be collected [6]. Therefore we compare against *DALL-E-Bot* [5], a method for zero-shot rearrangement with VLMs. It uses a diffusion model to generate a goal image, and is restricted to 2D top-down scenes. (2) We also compare with a variant of our method, *D2R-One-View*, which uses only the first camera view throughout the whole pipeline (including object captioning), avoiding the need for data collection. Instead of a NeRF, a colour point cloud is rendered as the visual model. (3) Next, the *D2R-Distract* ablation does not use GPT-4 to filter out irrelevant objects (distractors). (4) The *Physics-Only* baseline does not use CLIP to evaluate poses: instead, it uses a random physically valid pose. (5) *D2R-No-Norm* does not use normalising captions. (6) *D2R-Vis-Prior* investigates the visual prior of CLIP: it does not use normalising captions for normalisation. Instead, it uses them as goal captions. E.g. if the goal caption was previously "an apple inside a bowl", then it now becomes "an apple and a bowl". This tests whether CLIP knows a natural pose for the apple without being told it should go in the bowl. (7) Finally, *D2R-No-Smooth* ablates the spatial smoothing technique.

## Appendix IV
## Limitations Analysis

### A. Limitations and future work

In this section, we analyse the limitations of the current implementation of Dream2Real, and identify interesting directions for future work.

**Low-tolerance tasks** like insertion would require sampling poses more densely, which is computationally expensive if poses are sampled in a grid.

**Running time** is a limitation of the current implementation of our framework. Scanning the scene (e.g. when the

robot first observes a new room) takes 3-5 minutes. This can be reduced in future work using sparse NeRFs [58] or generative image-to-3D methods [59], [60]. The current implementation also requires significant computation time, although everything can be run with only 1 desktop GPU (we used an RTX 4090). After the user instruction is received, it currently takes approximately 6 minutes to render, check and score all the poses. In future work this can be sped up with an iterative coarse-to-fine approach: sampling poses sparsely at first, and then more densely in the higher-scoring regions.

As shown in prior research [61], CLIP can exhibit **bag-of-words behaviour**, i.e. CLIP performs poorly on goal captions where the order of words matters. E.g. *"a fork to the left of a knife"* often places the knife to the left of the fork instead. However, as shown empirically, CLIP performs well on several useful tasks and even complex spatial relations. As VLMs improve in the future, this limitation will be less significant. To further improve the robustness of CLIP scoring, in future work arrangements could be rendered from multiple views, and the CLIP scores aggregated from different perspectives.

### B. 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.