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

Ivan Kapelyukh[*1,2], **Yifei Ren**[*1], **Ignacio Alzugaray**[2], **Edward Johns**[1]
[1] The Robot Learning Lab, [2] The Dyson Robotics Lab
Imperial College London

Abstract

We introduce Dream2Real, a robotics framework which integrates 2D vision-language models into a 3D object rearrangement method. The robot autonomously constructs a 3D NeRF-based representation of the scene, where objects can be rendered in novel arrangements. 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 via pick-and-place. Real-world results show that this framework enables zero-shot rearrangement, avoiding the need to collect a dataset of example arrangements. Videos are available at: https://www.robot-learning.uk/dream2real

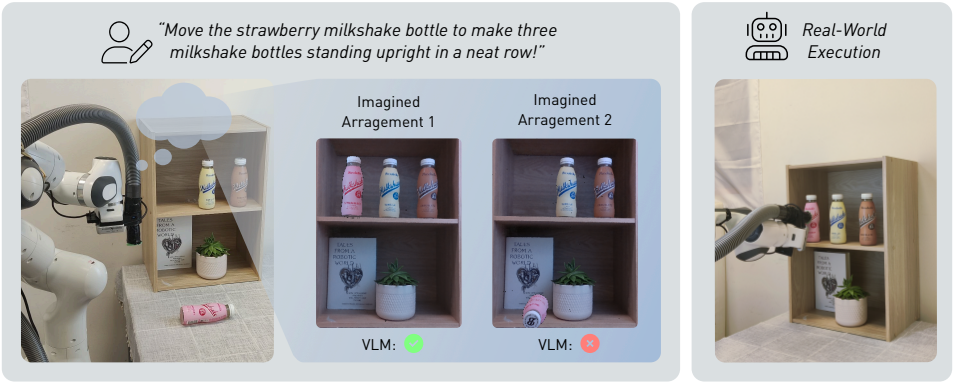

Figure 1: The robot first builds a 3D representation of the scene. Then it imagines new object arrangements, and evaluates them with a VLM to select the arrangement which best matches the user instruction. Finally, the robot uses pick-and-place to recreate the goal arrangement in the real world.

1 Introduction

Consider a task such as arranging bottles in a row (as in Figure 1). Before physically transporting the bottle, a human might first determine where the bottle should go (i.e. a goal pose), so that it is upright and on the top shelf with the others. One approach to determine a goal pose is to sample a pose, imagine (or *dream*) what the scene would look like if the bottle were in that pose, and then evaluate whether that scene arrangement looks correct and fulfills the task. In this paper, we study how robots can use this approach to determine semantically correct goal poses, and show how this leads to a language-conditioned 3D object rearrangement framework, Dream2Real.

[*]Joint first authorship. Corresponding author: ik517@imperial.ac.uk

NeurIPS 2023 Workshop on Robot Learning: Pretraining, Fine-Tuning, and Generalization with Large Scale Models, New Orleans, United States of America

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

Our approach is summarized in Figure 1. We address several difficult technical challenges, including autonomously constructing a 3D NeRF-based [4] representation of the scene which can be rearranged in imagination, and interfacing this with 2D VLMs to make effective use of their web-scale visual-language prior. Experiments show that our Dream2Real approach outperforms other recent work on VLMs for tabletop rearrangement [5], and demonstrates 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 applies VLMs to the task of object rearrangement **zero-shot**, without requiring a training dataset of example arrangements to be collected. Second, it achieves **3D rearrangement**, whereas prior work [5] is limited to top-down scenes. Third, it is **language-conditioned**, allowing it to work on an open set of objects and scenes. Prior work on rearrangement typically requires thousands of example arrangements [6]–[8] (for analysis of related work, see Appendix A). To the best of our knowledge, Dream2Real is the first method which performs 6-DoF rearrangement *zero-shot* by using the web-scale vision-language reasoning of VLMs.

2 Method

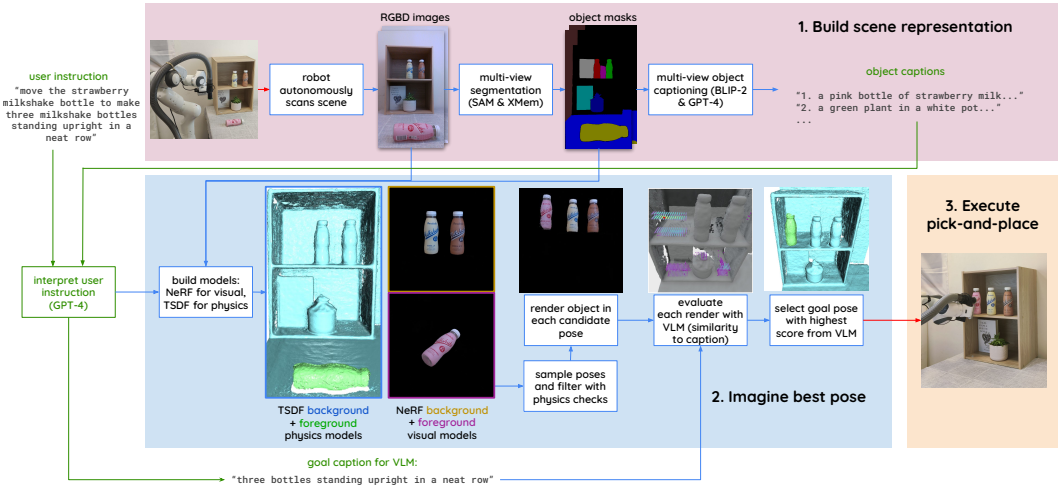

Figure 2: The Dream2Real pipeline. The robot first autonomously builds a model of the scene. Then the user instruction is used to determine which object should be moved. This allows the robot to create a separate physics model and visual model for that object, and also for the task background. The robot can then imagine new configurations of the scene and score them using a VLM. Finally, the highest-scoring pose is used as the goal pose for pick-and-place to execute the rearrangement.

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 the Appendices for further details (Appendix B).

Before the user instruction is received, the robot constructs an object-level scene representation (detailed in Appendix B.1). This includes collecting a set of RGBD images, segmenting the first image into object masks with SAM [9], tracking these object masks across subsequent images with XMem [10], getting a caption for each object from each view using BLIP-2 [11], and aggregating those captions across views into an object description using a language model (GPT-4 [12]).

When the user instruction is received, a language model is used to process this instruction (Appendix B.2) 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

Figure 3: The shopping, pool ball, and shelf scenes used in experiments.

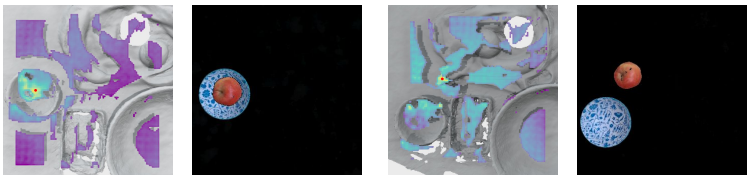

Figure 4: Qualitative results from the shopping scene for the tasks "apple in bowl" (left) and "apple beside bowl" (right), from different initial configurations. In the heatmaps (overlaid on the TSDF of the initial scene), yellow indicates high-scoring poses of the apple, whereas dark blue indicates low scores, and colliding poses are not included. The red dot highlights the highest-scoring position. The best-scoring rendered image (as seen by CLIP) is shown to the right of the corresponding heatmap.

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 B.3). We use NeRF-based Instant-NGP [4], [13] for visual models and TSDF [14] for physics models. Then, these models are used to determine the best pose for the movable object (Appendix B.4). 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 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 B.5).

3 Experiments & Discussion

We evaluate on 10 rearrangement tasks across 3 real-world scenes, as shown in Figure 3 (see Appendix C for details). We compare against DALL-E-Bot [5], since it is also a zero-shot method using VLMs, and we also study several ablations and variants of our method. Baseline details are in Appendix C.3.

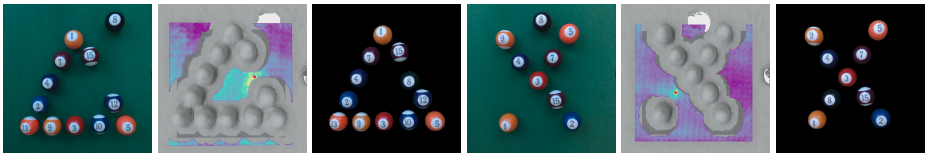

Figure 5: Qualitative results from the pool ball scene for the tasks *"in triangle"* (left half) and *"in X shape"* (right half). The initial scene is shown first. The red dot highlights the high-scoring area.

Zero-shot multi-task rearrangement. First we evaluate on the "shopping scene". The robot must select an object from the shopping bag and place it in one of the containers based on the user command. We evaluate success rate based on the predicted goal pose (see Appendix C.2). Qualitative results are in Figure 4. Table 1 shows quantitative results. Our method significantly outperforms DALL-E-Bot [5]. DALL-E-Bot generates a goal image and then attempts to match generated objects

Table 1: 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**

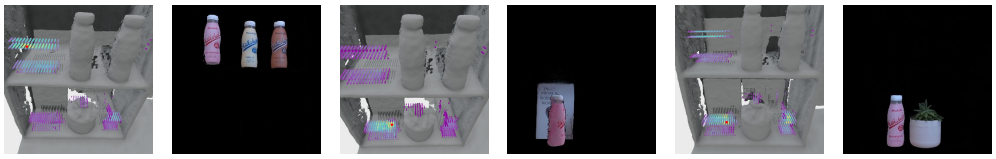

Figure 6: Qualitative results for the tasks *"bottles in a row"* (leftmost heatmap & render), *"in front of book"* (middle) and *"near plant"* (rightmost heatmap & render) from the shelf scene.

to the real objects, whereas our method avoids this difficult matching problem by evaluating imagined arrangements of the real objects. Further experiment details and results discussion (including ablation analysis) are in Appendix C.4. This also shows that our method is robust to distractors and can succeed at everyday rearrangement tasks zero-shot, using the VLM's web-scale visual-language prior.

Multi-object geometric relations. In this scene, the robot must complete a partial arrangement of pool balls to create a geometric shape, e.g. a triangle or an X shape. Further experiment details and a table of quantitative results (Table 2) are in Appendix C.5. Qualitative results are in Figure 5. Our method outperforms DALL-E-Bot [5] and demonstrates an understanding of multi-object relations.

6-DoF rearrangement in a 3D scene. In the shelf scene (see Figure 3), the method must perform 6-DoF rearrangement to pick up the bottle lying on the table and place it upright on the shelf. There are 3 tasks: making the bottles into a row, placing the bottle in front of the book, or placing it near the plant. Visualizations are in Figure 6, and quantitative comparisons in Figure 7 (left). See Appendix C.6 for details. This shows how Dream2Real makes it possible to apply 2D VLMs to 3D scenes.

Demonstrating physical execution. Although the main contribution is predicting the goal pose, we also demonstrate how a robot can physically perform the rearrangement. Results are in Figure 7 (right), showing the benefits of multi-view observations (see Appendix C.7 for analysis). Robot videos for all scenes are available on our website: https://www.robot-learning.uk/dream2real.

Conclusions. We show for the first time how 2D VLMs can be used for language-conditioned 3D object rearrangement *zero-shot*, without needing to collect any example arrangements. We analyze the limitations of this approach in Appendix D. Direct comparisons show that our method Dream2Real out-performs prior work on tabletop rearrangement by using VLMs in an evaluative manner.

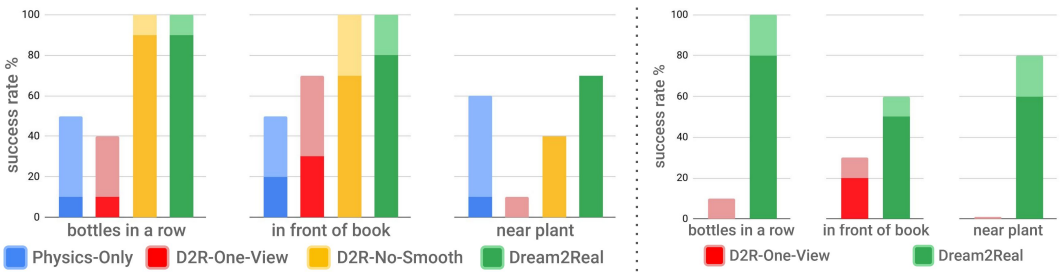

Figure 7: **Left:** Success rates for the shelf scene. A darker bar shows the success rate for predicting the full 6-DoF pose, and a lighter bar on top indicates roll-outs where the method correctly predicted the position but not the orientation. **Right:** Success rates for robotic execution. A darker 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 (e.g. due to shelf collision).

Acknowledgments

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, Xin Kong, Hide Matsuki, Marwan Taher, Vitalis Vosylius, and Kentaro Wada. We are also grateful to Shikun Liu for diagram design. This work was supported by Dyson Technology Ltd, and the Royal Academy of Engineering under the Research Fellowship Scheme.

References

[1] A. Radford, J. W. Kim, C. Hallacy, *et al.*, "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, *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in *arXiv*, 2023.

[4] 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.

[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] 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.

[8] A. Simeonov, A. Goyal, L. Manuelli, *et al.*, "Shelving, stacking, hanging: Relational pose diffusion for multi-modal rearrangement," *arXiv preprint arXiv:2307.04751*, 2023.

[9] A. Kirillov, E. Mintun, N. Ravi, *et al.*, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.

[10] 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.

[11] 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. arXiv: 2301.12597 [cs.CV].

[12] OpenAI, *Gpt-4 technical report*, 2023. arXiv: 2303.08774 [cs.CL].

[13] 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.

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

[15] D. Batra, A. X. Chang, S. Chernova, *et al.*, "Rearrangement: A challenge for embodied AI," *arXiv*, 2020.

[16] 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.

[17] G. Sarch, Z. Fang, A. W. Harley, *et al.*, "TIDEE: Tidying up novel rooms using visuo-semantic commonsense priors," in *European Conference on Computer Vision*, 2022.

[18] 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.

[19] 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.

[20] A. Zeng, P. Florence, J. Tompson, *et al.*, "Transporter networks: Rearranging the visual world for robotic manipulation," *Conference on Robot Learning (CoRL)*, 2020.

[21] 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.

[22] 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`.

[23] 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.

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

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

[26] 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*, 2023.

[27] 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.

[28] Y. Kant, A. Ramachandran, S. Yenamandra, *et al.*, "Housekeep: Tidying virtual households using commonsense reasoning," *arXiv*, 2022.

[29] 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.

[30] J. Wu, R. Antonova, A. Kan, *et al.*, "Tidybot: Personalized robot assistance with large language models," *Autonomous Robots*, 2023.

[31] M. Ahn, A. Brohan, N. Brown, *et al.*, "Do as I can, not as I say: Grounding language in robotic affordances," *arXiv*, 2022.

[32] A. Stone, T. Xiao, Y. Lu, *et al.*, "Open-world object manipulation using pre-trained vision-language model," in *arXiv preprint*, 2023.

[33] T. Yu, T. Xiao, A. Stone, *et al.*, "Scaling robot learning with semantically imagined experience," *arXiv*, 2023.

[34] 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.

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

[36] 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.

[37] 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.

[38] W. Yu, N. Gileadi, C. Fu, *et al.*, "Language to rewards for robotic skill synthesis," *Arxiv preprint arXiv:2306.08647*, 2023.

[39] Y. J. Ma, W. Liang, V. Som, *et al.*, "Liv: Language-image representations and rewards for robotic control," *arXiv preprint arXiv:2306.00958*, 2023.

[40] 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*, R. Firoozi, N. Mehr, E. Yel, *et al.*, Eds., ser. Proceedings of Machine Learning Research, vol. 168, PMLR, 23–24 Jun 2022, pp. 893–905.

[41] S. Sharma, A. Rashid, C. M. Kim, *et al.*, "Language embedded radiance fields for zero-shot task-oriented grasping," in *7th Annual Conference on Robot Learning*, 2023.

[42] L. Yen-Chen, P. Florence, A. Zeng, *et al.*, "MIRA: Mental imagery for robotic affordances," in *Conference on Robot Learning (CoRL)*, 2022.

[43] 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.

[44] 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.

[45] 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.

[46] 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.

[47] 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.

[48] K. Jatavallabhula, A. Kuwajerwala, Q. Gu, *et al.*, "Conceptfusion: Open-set multimodal 3d mapping," *Robotics: Science and Systems*, 2023.

[49] 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.

[50] 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, 995–1001 vol.2.

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

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

[53] 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*, 2023. arXiv: 2303.11328 [cs.CV].

[54] 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.

A Related Work

Predicting goal poses is a key challenge in object rearrangement [15]. Prior work classifies the correct receptacle in which to place an object [16]–[18], or chooses from a dense set of possible poses [19]. Another approach learns rearrangement from full demonstrations [2], [20]. The prediction of goal poses can be conditioned in several ways. Some methods allow users to specify relational predicates [21], [22]. Others learn a personalized representation of user preferences [23]–[25]. 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 [26] learns a language-conditioned distribution over desirable poses using a diffusion model. SceneScore [27] uses an energy-based model, thus learning to evaluate whether a given arrangement is desirable. These rearrangement methods typically require a training dataset of thousands of example arrangements [6]–[8]. This is effective for specific rearrangement tasks, but is difficult to scale to unstructured environments such as homes, because of the difficulty of generating realistic training data.

Instead, some methods use **large language models** (LLM) pre-trained on web-scale data to predict arrangements [28]–[30]. These are effective for high-level planning, but language models on do not have the visual perception needed to e.g. assess whether an object is oriented correctly. Our framework integrates vision-language models to achieve this. 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. Additionally, direct experimental comparisons (Sections C.4 & C.5) 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], [31]–[36]. VLMs in particular can also be used to connect user instructions in natural language with the robot's visual perception [2], [37]. Closer to our work, VLMs and LLMs can also be used as reward signals to train robot policies [38]–[40]. 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 [37], [41]–[43]. Implicit neural representations such as NeRF [4] have shown a strong ability to produce photorealistic renders from novel views. Instant-NGP [13] significantly accelerates the training and rendering speed of NeRF using multiresolution hash encoding, making it possible to render in real-time. Works like vMAP [44] also demonstrate that scenes can be decomposed with object-level reconstruction. Related to our method, some works [45]–[48] also combine 3D representations with LLMs, but do not focus on zero-shot 3D robotic rearrangement.

B Method Details

B.1 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). This design reduces the time between a user issuing an instruction and the robot completing it, since the robot has already built a task-agnostic scene representation beforehand. Therefore, the robot starts by autonomously scanning the scene, and collecting a set of RGBD images which will be used later for NeRF training, and for building physics models that will later be used for collision checking. In our experiments, we use a hemispherical camera trajectory facing the scene center. We segment the scene into objects by running Segment Anything (SAM) [9] 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 [10] which effectively handles 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 [11] to retrieve a per-crop caption. Since individual captions for a specific object may be different across views, we use an LLM (GPT-4) [12] to aggregate these into one coherent object description. 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.2 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 *normalizing 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 fulfill the instruction. Relevant objects are those which the VLM should observe to evaluate whether the user instruction is fulfilled (apple and bowl in this instance). This technique avoids showing distractor (irrelevant) objects to CLIP, which we show experimentally is important for performance. The goal caption is a description of the desired final state after the instruction has been completed. In this example, it would be: *"an apple inside a bowl"*; Lastly, the normalizing 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 normalizing CLIP scores (Section B.4).

B.3 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 [13]) 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 TSDF and background TSDF, which we find achieves more accurate geometry than extracting a mesh from Instant-NGP.

B.4 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 discretized bins) works well. We move (virtually) the movable object's physics model to each of the sampled poses in turn and checked for collisions, to avoid rendering and evaluating invalid poses. For each valid pose after filtering, 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 [44]). We render the NeRFs from a fixed camera pose from our scanning trajectory pointing at the center 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 normalizing 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 pose as the goal pose for the movable object.

B.5 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 [49], but any off-the-shelf grasping method can easily be applied. We then use inverse kinematics and a motion planner (RRT-Connect [50]) to find a valid path between the pick and place poses which avoids collisions, using the object collision meshes that the robot constructed previously.

C Experiment Details

C.1 Hardware Setup

We instantiate our framework on a 7-DoF Franka Emika Panda robot, equipped with a wrist-mounted Intel RealSense D435i RGBD camera and a compliant suction gripper for physically performing the rearrangement.

C.2 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 by examining whether the predicted goal pose is within the success region for the task. Task success definitions are detailed in the supplementary material on our project webpage. 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 Figure 7 (right).

C.3 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 color 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 normalizing captions. (6) *D2R-Vis-Prior* investigates the visual prior of CLIP: it does not use normalizing captions for normalization. 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. We want to investigate whether spatial smoothing is necessary, or whether using the pure, unsmoothed CLIP scores is still robust.

C.4 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 rescan the scene.

Qualitative results are in Figure 4, showing a heatmap of CLIP scores next to the best-scoring render. Table 1 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 normalizing captions are effective for these tasks in forcing CLIP to focus on the spatial relations in the instruction. *D2R-No-Smooth* performs slightly better than our method on this scene, showing that in this case the CLIP heatmap is not so

affected by outliers as to require smoothing. 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.

C.5 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 2. 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. *D2R-Distract* performs well because there are no distractors here, and the green pool table background may provide helpful context to CLIP.

Table 2: 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**

C.6 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. discretize 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 of our method in Figure 7 (left). *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 fails occasionally because the single-view captions are incorrect, 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.

C.7 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, along with supplementary material with additional experiment details: https://www.robot-learning.uk/dream2real. Quantitatively, we evaluate on the 6-DoF shelf scene. We compare our multi-view method with *D2R-One-View*. Results are in Figure 7 (right). We conduct 10 roll-outs for each method-task combination. Note that we now also automatically exclude unreachable goal poses, so the results for each task will differ from those in the previous experiment (Appendix C.6). We find that the single-view baseline fails more often due to incomplete reconstructions, which impacts both goal pose prediction and collision-free motion planning. This shows that our multi-view Dream2Real method can physically perform 6-DoF rearrangement using the predicted goal poses.

D Limitations

Low-tolerance tasks. Our method is not well-suited for low-tolerance tasks like insertion, since sampling high-resolution poses is computationally expensive.

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 [51] or generative image-to-3D methods [52], [53]. After the user instruction is received, it currently takes approximately 6 minutes to render, check and score all the poses. In future work, sampling object poses iteratively rather than from a grid would make this more time-efficient, as will better parallelizing the computations.

Limitations of CLIP. As shown in prior research [54], CLIP can exhibit bag-of-words behavior, 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 our experiments show, CLIP performs well on several useful everyday tasks and even complex spatial relations. As VLMs improve in the future, this limitation will be less significant.