CALAMARI: Contact-Aware and Language conditioned spatial Action MApping for contact-RIch manipulation

Anonymous Author(s)
Affiliation
Address
email

Abstract: Making contact with purpose is a central part of robot manipulation and remains essential for many household tasks – from sweeping dust into a dustpan, to wiping tables; from erasing whiteboards, to applying paint. In this work, we investigate learning language-conditioned, vision-based manipulation policies wherein the action representation is in fact, *contact itself* – predicting contact formations at which tools grasped by the robot should meet an observable surface. Our approach, Contact-Aware and Language conditioned spatial Action MApping for contact-RIch manipulation (CALAMARI), exhibits several advantages including (i) benefiting from existing visual-language models for pretrained spatial features, grounding instructions to behaviors, and for sim2real transfer; and (ii) factorizing perception and control over a natural boundary (i.e., contact) into two modules that synergize with each other, whereby action predictions can be aligned per pixel with image observations, and low-level controllers can optimize motion trajectories that maintain contact while avoiding penetration. Experiments show that CALAMARI outperforms existing state-of-the-art model architectures for a broad range of contact-rich tasks, and pushes new ground on embodiment-agnostic generalization to unseen objects with varying elasticity, geometry, and colors in both simulated and real-world settings.

Keywords: Contact-rich Manipulation, Visual-language guided policies

1 Introduction

Contact-rich manipulation is ubiquitous in our day-to-day lives, encompassing a broad range of tasks including sweeping dust into a dustpan, wiping tables, erasing a whiteboard, and applying paint with a brush. A key challenge in performing these tasks lies in controlling the interactions between tools and their environments. For instance, when sweeping, it is crucial to ensure continuous contact between the bristles and the surface while directing the collected dust towards the dustpan.

Language-conditioned representations and policies are a promising approach to addressing the challenges of contact-rich manipulation, particularly for domestic applications. For one, language is a powerful tool for creating abstractions that enable generalization for a wide variety of tasks and environments. Secondly, language will be among the most common methods to command robots such as when performing tasks in the home. Recent work has demonstrated how large pretrained visual-language models (VLMs), such as CLIP [1] and PaLM-E [2], enable zero-shot transfer of visual-semantic reasoning based on language prompts and well-structured visual and language embedding spaces [3, 4, 5, 6, 7, 8, 9, 10]. However, previous efforts have predominantly focused on rearrangement-based tasks and have not adequately addressed the reasoning involved in contact-rich manipulations, thereby limiting their applicability to tasks like wiping, sweeping, or scooping.

In this paper, we introduce a novel language-conditioned and contact-aware spatial action map representation that predicts *contact affordances* – contact formations at which tools grasped by the robot

Submitted to the 7th Conference on Robot Learning (CoRL 2023). Do not distribute.

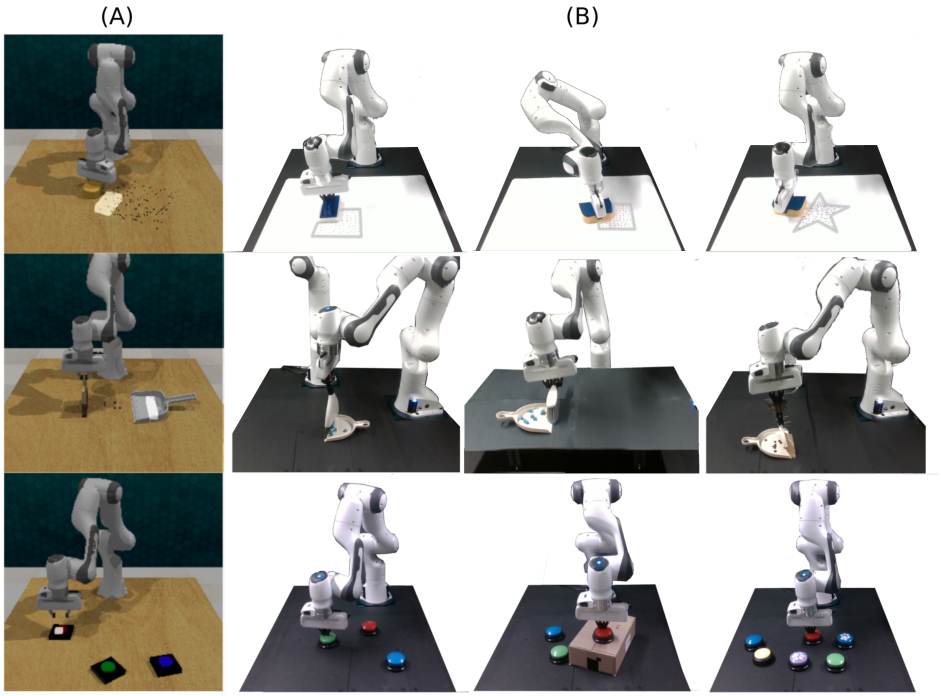

Figure 1: CALAMARI is a contact-aware and language-conditioned spatial-action mapping for contact-rich manipulation. We show that (A) wiping, sweeping, and pushing tasks, trained solely on simulations with a single task and prompt, can be (B) directly transferred to the real world and applied to new environments with unseen tools, robot setups, table elevations, and prompts.

should meet an observable surface in order to perform a task. Our novel multi-modal spatial action maps are specifically for contact-rich manipulation where each pixel represents a binary indication of extrinsic contact between an object and the environment, and the entire map is implicitly linked to the robot configuration. Notably, our extrinsic contact policies remains agnostic to intricacies of specific tools and physical robot platforms, unlocking unprecedented possibilities for generalization to unseen objects with distinct elasticity, geometry, and colors, both in sim and the real world.

The key contributions of this paper are: 1) multi-modal extrinsic contact policy with novel spatial-action maps that outperforms SOTA language-guided manipulation methods for contact-rich tasks; 2) an MPPI controller algorithm compatible with the predicted contact goal and contact constraint; and 3) generalization to unseen objects with various elasticity, geometry, and colors in both simulation and the real world without requiring fine-tuning of the extrinsic contact policy.

2 Related Works

Language Grounding for Manipulation: The recent advancements in large language models (LLMs) [11, 12, 13] and visual-language models [1, 2] have enabled language-grounded manipulation. Numerous approaches [4, 9, 3, 14, 8, 15, 6, 16], have emerged and demonstrated remarkable multi-task performance, particularly for pick-and-place tasks. These end-to-end methods map directly from RGB or voxel observations to robot configurations. However, one drawback of end-to-end approaches is their reliance on extensive real-world data collection, which can span several weeks or months [15, 6]. To address this real-world data efficiency challenge, [9] achieved significant improvements by predicting only key frames and discretizing input and action spaces. Nevertheless, like end-to-end models, [9] faces the challenge of effectively handling novel tool manipulations, as the tool variations must be captured in the training demonstrations. We show that our approach CALAMARI can efficiently handle contact-rich tasks while reducing the burden on real-world data collection via zero-shot sim2real transfer.

Planning with Extrinsic Contact: Controlling extrinsic contacts for planning and manipulation has been an active area of research, with several notable contributions in recent years[17, 18, 19, 20, 21, 22, 23, 24]. Kim et al. [17] developed a method for simulataneously estimating and controlling extrinsic contacts of rigid objects using tactile signitures, while [18] focused on planning with extrinsic point contact for planar manipulation scenarios. Van der Merwe et al. [21, 22] demonstrated

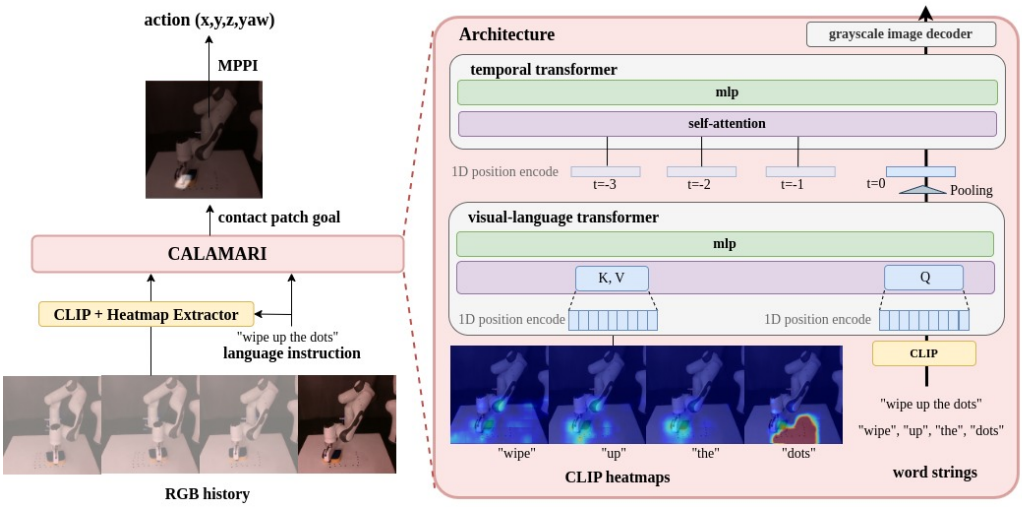

Figure 2: **Overview.** As shown in the left panel, our method utilizes the history of RGB and language instructions as inputs and predicts the contact patch goal as a binary mask from the input image frame. The three yellow blocks (e.g., 'CLIP') represent the pretrained models, which are not updated during training.

extrinsic contact detection for deforming tool manipulation, specifically scraping with spatulas, incorporating predefined contact goals and learned dynamics. Wi et al. [23, 24] presented a technique for predicting dense contact patches for compliant tools using learned tool dynamics. CALAMARI provides a framework for policies with contact goals, allowing for the seamless integration of the contact dynamics models [21, 22, 23] into a broader context of planning and manipulation.

3 Methodology

3.1 Problem Statement

Our objective is to learn a function F that predicts the next desired contact patch goal, denoted as $\mathbf{C}_t^{goal}$, at time t. This function takes as input a sequence of RGB key frames and language instructions to compute contact goals: $F((\mathbf{I}_{t-w+1}, \boldsymbol{l}_{t-w+1}), \ldots, (\mathbf{I}_{t-1}, \boldsymbol{l}_{t-1}), (\mathbf{I}_t, \boldsymbol{l}_t)) = \mathbf{C}_t^{goal}$, where $\mathbf{I}_t$ is an RGB image, $\boldsymbol{l}_t$ is a language instruction, and w dictates the observation time window considered for making predictions. The output is the contact patch goal $\mathbf{C}_t^{goal} \in \mathbb{R}^{w \times h}$, a 2D binary mask in $\mathbf{I}_t$'s camera frame and is particularly well-suited to for contact-rich planar tasks [5, 25]. The contact patch $\mathbf{C}_t^{goal}$ can be de-projected to a point cloud by overlaying the predicted mask onto a depth map that excludes objects/tools ($\mathbf{D}_{nominal}$). The point cloud conversion enables the utilization of a model predictive controller to achieve the desired contact formation.

In this work, data is provided in the form of demonstrations consisting of variable-length T key frame trajectories, denoted by $\boldsymbol{\tau} = ((\boldsymbol{l}_0, \mathbf{I}_0, \mathbf{C}_0), (\boldsymbol{l}_1, \mathbf{I}_1, \mathbf{C}_1), \ldots, (\boldsymbol{l}_T, \mathbf{I}_T, \mathbf{C}_T))$. Here, $\mathbf{C}_t \in \mathbb{R}^{w \times h}$ is the contact patch between the object (e.g., grasped tool) and environment represented by a binary mask from the camera perspective. The contact patch from the demonstrations serves as the ground truth contact goal for the function F.

3.2 Behavior Cloning Tool-Environment Interactions

Vision-Language Pre-Processing: CALAMARI has two key vision-language pre-processing steps to convert raw observations into CLIP features. Firstly, we generate word-wise heatmaps that highlight the spatial locations in the RGB images corresponding to specific words in the language instruction. The Heatmaps, denoted as $\mathbf{H}_t = f(\mathbf{I}, \boldsymbol{l})$, are grayscale images with dimensions $w \times h$, obtained through the image-language relevancy extraction introduced in [26]. Using heatmaps instead of raw RGB input is particularly advantageous for sim2real and generalization to novel objects and environments, as long as similar heatmap distributions are present. This is because heatmaps provide abstract representation that is less sensitive to variations in visual appearance caused by factors like object colors and lighting conditions. The heatmap is subjected to further abstraction through a pretrained heatmap encoder [27], resulting in word-wise heatmap features. These features are organized based on the word sequence and serve as the query input for the Visual-language trans-

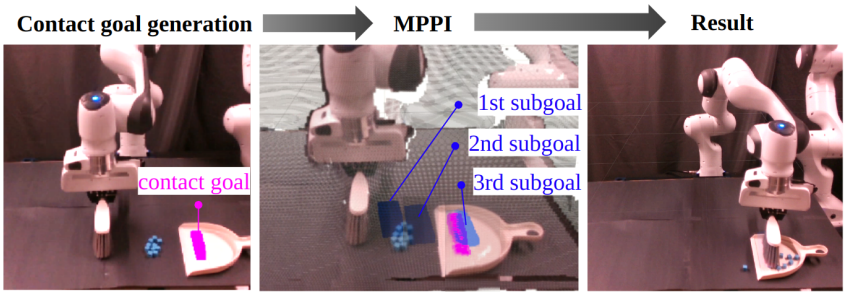

Figure 3: We generate the key contact goal via CALAMARI and generate subgoals to reach the contact goal via MPPI that are linked with corresponding low-level actions. Each subgoal, or the immediate target in 3D space, is generated for each time step until the object reaches the contact goal. Once we have reached it, we generate a new contact goal until the task terminates.

former, which will be discussed in detail in the subsequent section. Additional information about the selection of heatmap encoders can be found in Appendix-B.1.

Similarly, we convert the language prompts into embeddings using CLIP's language encoder. In this case, we combine both sentence and word embeddings to capture the contextual information conveyed by the language prompt while focusing on individual words. These embeddings are also arranged in accordance with the word order and serve as inputs for the subsequent transformer networks in the following section.

Architecture: CALAMARI architecture (Fig. 2) consists of of two types of transformers, visual-language (v-l) transformer and temporal transformer. Drawing inspiration from the LAVA structure [14], our visual-language (v-l) transformer is responsible for encoding inputs into multi-modal features and our temporal transformer fuses latent observation over time to generate spatial-actions. In contrast to [14], the language query is comprised of a set of sentence and word embeddings, denoted as $\mathbf{Q} \in \mathbb{R}^{(l+1)\times d_{ft}}$. Here, l represents the sentence length and d_{ft} is the feature dimension. The keys and values, denoted as $\mathbf{K}$ and $\mathbf{V} \in \mathbb{R}^{l\times d_{ft}}$ respectively, are word-wise heatmap embeddings. The temporal transformer takes into account the history of latent observations from the v-l transformer, considering the w most recent time stamps. Using the history of observation is important in contact-rich manipulation as it often involves occlusion caused by tools and the robot itself. The temporal transformer utilizes self-attention multi-modal features stacked with time, represented as $\mathbf{Q} = \mathbf{K} = \mathbf{V} \in \mathbb{R}^{w\times d_{ft}}$. The outputs of the temporal transformer are decoded using a grayscale image decoder, similar to the decoder architecture of the UNet model proposed by [28], without incorporating skipping layers.

Training Loss: The loss function is a standard supervised behavioral cloning loss similar to the prior works [6, 29, 30]. Specifically, we use L2 regression between predicted contact patch $\mathbf{C}_t^{goal}$ and the ground truth contact patch $\mathbf{C}_t^{gt}$ as the following: $\|\mathbf{C}_t^{goal} - \mathbf{C}_t^{gt}\|$.

3.3 MPPI controller with the contact goal

From the contact goals to controlling objects as in Fig. 3, we propose to use the sampling-based Model Predictive Path Integral (MPPI) algorithm [31], compatible with non-convex and non-differentiable cost functions. We use a 4 DoF action space in the robot end effector task space (x, y, z, yaw). For a sampled trajectory $\boldsymbol{a} = (\boldsymbol{a}_0, \boldsymbol{a}_1, \ldots, \boldsymbol{a}_{w-1})$, we define the following cost.

$$\sum_{i=1}^{w} dist(\mathbf{C}_t^{goal} * \mathbf{D}_{nominal}, \mathbf{P}_{t+i}^{dyn}) + \lambda(1 - IoU(\mathbf{C}_t^{goal}, \mathbf{C}_{t+i}^{dyn}))$$

$\mathbf{P}_{t+i}^{dyn}$ is a contact pointcloud prediction from rolling out rigid dynamics given the current pose $\boldsymbol{p}_t$ and a sequence of actions $\{\boldsymbol{a}_0, \ldots, \boldsymbol{a}_{i-1}\}$. $\mathbf{C}_{t+i}^{dyn}$ is a 2D projection of $\mathbf{P}_{t+i}^{dyn}$ to the camera frame. The first term of our cost function is about minimizing the mean euclidean distance between the center of $\mathbf{P}_{t+i}^{dyn}$ and the contact goal center. To do so, we uses $\mathbf{C}_t^{goal} * \mathbf{D}_{nominal}$ to get the contact goal pointcloud by overlaying contact goal mask with the nominal depth map. We estimate $\mathbf{P}_{t+i}^{dyn}$ by transforming objects with known geometry to the word frame and identifying intersections with environment (Appendix C.2). The second term promotes matching the shape of future contact to

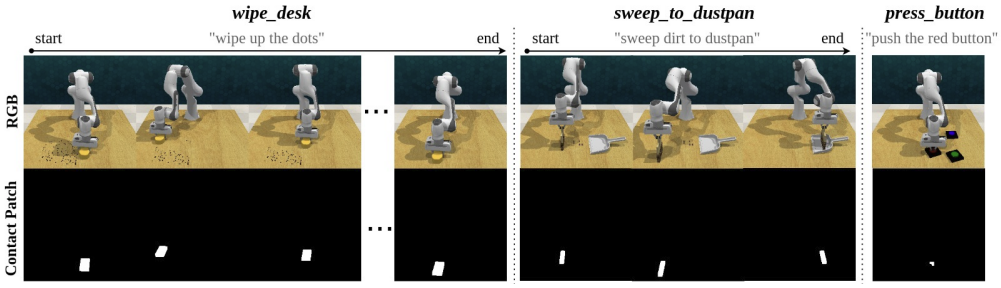

Figure 4: We visualize the dataset by displaying the language prompt alongside the RGB and contact patch sequences extracted from key contact frames in the demonstrations. The wiping task typically includes 7 to 12 key contact frames, sweeping involves 4 key contact frames, and the press button tasks have 1 contact patch.

the goal using Intersection over Union (IoU). We align the center of the prediction and the goal of contacts by subtracting the mean to focus on matching the contact shape (Appendix C.3).

Our MPPI has two contact constraints, implemented via penalty costs. The first constraint is to maintain contact via $\|\mathbf{C}_t^{dyn}\| > 0$. This penalizes any actions that makes no contact. The second constraint is $\max_z(D_{nominal} - P_{t+w}^{dyn}) > \epsilon_p$. This lower bounds the distance in z axis from the environment to the contact patch prediction with epsilon $\epsilon_p > 0$. The entire control algorithm we use is described in Appendix Alg. 1.

4 Experiments and Result

Datasets: Both our model and baseline were trained using 100 demonstrations per task on CoppeliaSim [32, 33]. Each task involved manipulating a *single* object and a *single* language prompt, as described in Fig. 4. In Sec. 4.1, we investigate the generalization performance using various unseen objects and prompts. This paper focuses on three different types of contact-rich tasks: multi-step patch contact '*wipe_desk*', three-step patch contact '*sweep_to_dustpan*', and single-step point contact '*press_button*'. For the wiping task, we improvised our own closed-loop demonstration where the sponge moves towards the center of dust clusters computed from DBScan [34]. The other two tasks use open-loop demonstrations predefined in [33]. Each demonstration consists of a language prompt and a sequence of RGB observations along with the corresponding contact patches obtained from the key contact frames (Fig. 4). The key contact frames were identified when the robot reached the waypoints defined in the demonstrations. However, frames where the contact patch was the same as the previous waypoints were removed to eliminate redundancy. Ground truth contact patches are computed via CoppeliaSim's contact detection algorithm.

4.1 Simulation Results

The task performance of our model across the tasks is presented in Table 1, where the scores are averaged over 25 test episodes. We conducted three different objects/tools for evaluations: one being the training object and the other two being held out objects (Fig. 5). Our objective in directly transferring to these held out objects is twofold: 1) To demonstrate the robustness of our contact goal policy in effectively adapting to shifts in heatmap distribution from variations in object's structural/visual features. 2) To showcase the flexibility of our MPPI controller in accommodating previously unseen contact formations from unseen object geometries. Note that we employed new task-specific language prompts for the *push_button* task's heldout objects to accommodate color change.

Evaluation Metrics: The evaluation metrics are the task success rates ranging from 0% to 100%. We evaluate *wipe_desk* and *sweep_to_dustpan* tasks with the percentage of dust removal. For the *push_button* tasks, binary metrics were used, where 0% indicated failure and 100% represented success without partial credits. ***wipe_desk*:** We evaluated the wiping performance after 20 contact goal generations, which is the point at which task performance plateaus (see Appendix Fig. 11). This task involves clearing one hundred dust particles with size $0.6\text{x}0.6\text{x}0.1\text{cm}$ that are randomly spawned within the bounding box of size $15cm\text{x}15cm$. We show in Tab. 1 that CALAMARI achieves 93% of dust removal with the training object. To assess the performance with heldout tools, we conducted two tests. Test1 examined the ability of our MPPI controller to manipulate objects with 55% decreased contact patch area. Test2 evaluated the robustness of CALAMARI against out-of-distribution broom heatmaps. In test2, the sponge was grasped by the handle, resulting in larger heatmap observations without occlusion by the gripper. As shown in Tab. 1, our method perfor-

Method	*wipe_desk*			*sweep_to_dustpan*			*push_button*		
	train	test1	test2	train	test1	test2	train	test1	test2
Ours	93%	79%	92%	93%	61%	46%	92%	60%	60%
PerAct	48%	70%	82%	70 %	0%	0%	63%	0%	4%

Table 1: RLBench success rate of each tasks in test cases using to train objects denoted as 'train' and heldout objects denoted as 'test1' and 'test2'

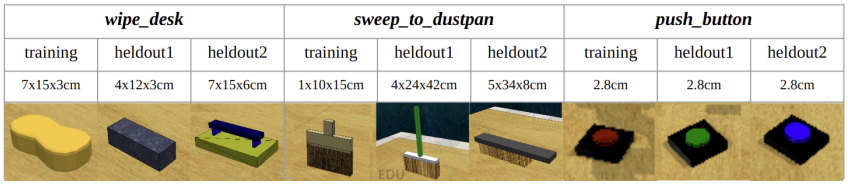

Figure 5: Train and test objects in simulation of four different tasks. The heldout objects exhibit variations not only in color, size, and shape but also in structural attributes, such as the handle locations. The third row indicates object dimensions in centimeter either as $width \times depth \times height$ or as radius.

mance only decreased by 14% for test1 when handling smaller contact patches. Additionally, our goal generation exhibited robustness to heatmaps variations as test2 results are comparable to training object performance.

***sweep_to_dustpan*:** We evaluated sweeping task with 3 contact goal generations as in the demonstration. *sweep_to_dustpan* task involves 5 dust particles with the size of $1\text{x}1\text{x}1\text{cm}$. Using training object, we achieved a 83% success rate in test cases for sweeping to a dustpan. We then transferred the pretrained model to two test objects: one with a longer handle, similar to the original, and another with a handle on the side. We noticed that larger visual discrepancies between the objects resulted in greater performance drops as test 2 shows worse performance than test1.

***push_button*:** We altered the prompt "*push the* $\{\}$ *button*" from the word "red" to "green" and "blue". This change resulted in varying CLIP heatmap intensities and word embedding inputs ,or the queries of our visual-language transformer. Nevertheless, our test results revealed that even with training based on a single word, our model achieved contact goal accuracy of less than 2.5cm in 60% of the two heldout cases. These findings emphasize the robustness and effectiveness of our approach in accurately pushing buttons of different colors.

Baseline: In this section, we compare our methods with PerAct, a state-of-the-art language-conditioned manipulation study that also utilizes CLIP features. PerAct employs voxel inputs from a multiple calibrated RGB-D camera setup, while our method relies solely on a single front camera view. In contrast, our MPPI controller requires additional information about the model's geometry and object pose, whereas the baseline does not require such object-related details. For this experiment, PerAct was trained from scratch with a 512 latent dimension. To align the baseline with our experimental setup, we trained PerAct with a single-task using the same dataset. During testing, we reduced the action space to (x, y, z, yaw) by providing ground truth values for the other actions (pitch, roll, gripper state, and collision prediction) as well as ground truth grasping.

Our method consistently outperforms PerAct on all three tasks (Tab. 1), both in training and testing with various objects. In the wiping task, our model achieves nearly double the success rate on training objects. Test objects for the wiping tasks produce a similar relative transformation between the grasped position on the tool and the contact surface. Surprisingly, PerAct showed an even smaller margin (10%) compared to ours on test objects. However, the results of the sweeping task's test1 and test2 suggest that PerAct struggles with larger changes in transformations between the grasped position and the contact surface. Our model is more robust as these changes have less impact on contact planning than on the robot configuration. Furthermore, CALAMARI outperforms PerAct when tested on unseen prompts in the button pressing task. This outcome was expected, as PerAct was not trained to detect buttons of different colors using RGB-D observations. In contrast, CALAMARI is trained on CLIP's heatmap to leverage VLM's generalization ability for interacting with unseen prompts of different colored buttons.

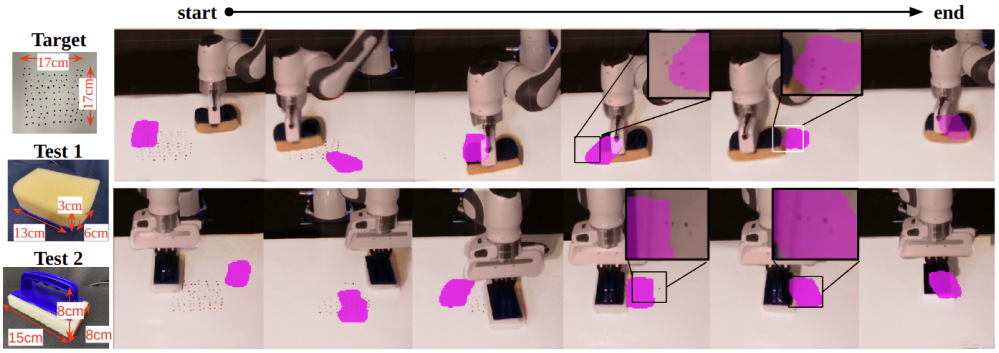

Figure 6: We visualized contact patch goals in magenta from CALAMARI. The first row represents the predictions when using the test1 object, while the second row corresponds to the test2 object. Our model can successfully identify small clusters of dots and can navigate back and forth until all the dots are erased using a closed loop policy. In real-world scenarios where the pressure exerted by a sponge on the board is not evenly distributed, this ability becomes particularly significant as the sponge may fail to erase certain dots even when it passes over them.

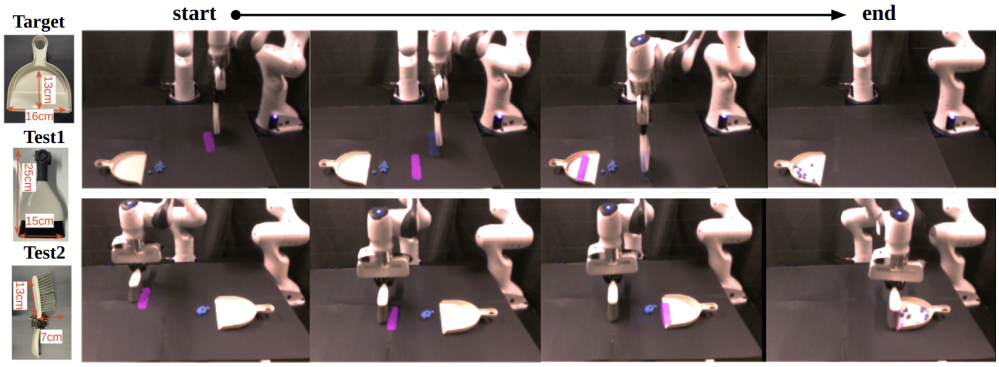

Figure 7: The real-world sweeping results were obtained using two brooms with small width margins (1-3 cm) with the target dustpan. Left panel shows three contact goal generations, visualized in magenta. The results in the last column demonstrate that our policy exhibits excellent zero-shot transfer to the real-world, even with unseen tool geometries. Furthermore, we demonstrate that our method is agnostic to the arrangement of the robot, as the same policy was applied to both the right and left arm.

4.2 Sim2Real

In this section, we directly transferred the pretrained model from simulation to the real world without fine-tuning. The quantitative analysis (Tab. 2) was conducted using 10 runs for each object with consistent resetting across all tasks.

Real-world Set-up: We utilized 2 Franka-emika robots and an Intel Realsense D435 camera for our real-world setup. The distance between the robot and camera between the real-world and simulation environments was [0.18m, 0.02m, -0.21m] in the x, y, and z directions. We found that the spatial-action map allows us to accurately predict contact patches in the camera frame regardless of the differences in camera positioning between simulation and real-world set-ups.

***wipe_desk*:** For the reset, we draw one hundred dots within a bounding box measuring $17cm$x$17cm$ with an black marker. We generated 20 contact goals per run and counted the number of erased particles for evaluation. Fig. 6 shows our contact patch goals, where we visualize the first and last three frames when contacts goals are generated. We also conducted experiment to erase different distributions of dots with the test1 object in Appendix Fig. 12.

***sweep_to_dustpan*:** We performed resets by arranging the dustpan in 10 different configurations within a bounding box measuring $9cm$x$14cm$. Each configuration included 10 $1cm^3$ cubes placed in front of the dustpan. For evaluation, we counted the number of cubes

wipe_desk	test1	91%
	test2	98%
sweep_to_dustpan	test1	91%
	test2	92%
	comp.	85%
push_button	red	90%
	green	60%
	blue	40%

Table 2: Direct transfer to real-world using test objects described in Fig. 6, 7, 8. 'comp' utilizes learned compliant tool dynamics as in Sec. 4.3.

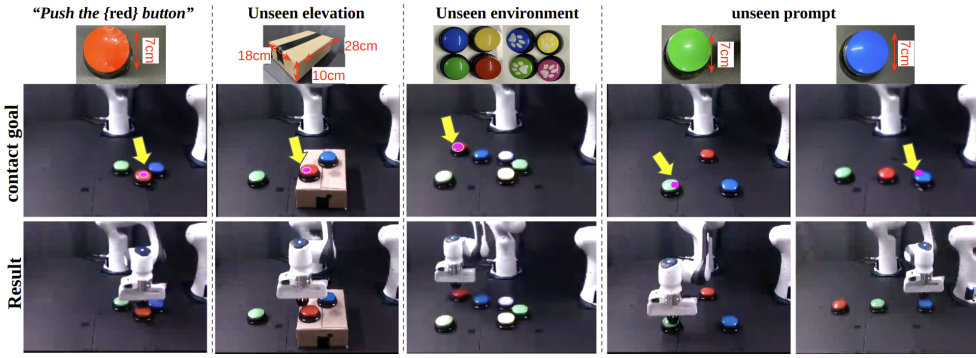

Figure 8: Our model, trained in CoppeliaSim on the prompt "press the red button," can be directly transferred to the real world without fine-tuning (first column). Our spatial action space is not limited to the tabletop (third column) and is robust to visual distribution (second and third column). It also handles unseen prompts like "press the green/blue button" (last two columns). The contact goal is visualized in magenta in the first row, while the second row shows the actual execution results

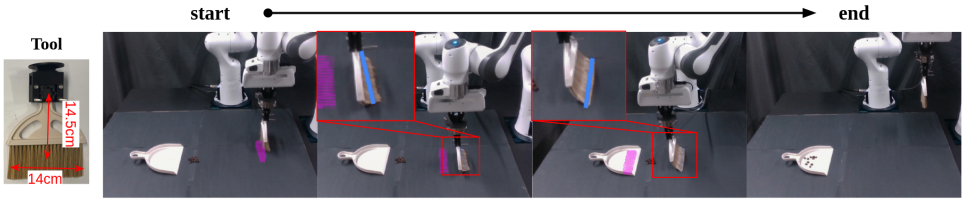

Figure 9: We demonstrate the ability of our model to generate goals for non-rigid tools. We repeat the sweeping task with a compliant tool, using a learned dynamics model to servo the contact of the tool with the tabletop. The predicted contact goal is visualized in magenta while the contact feature predictions from the dynamics model are overlaid in blue.

successfully swept into the dustpan. Fig. 7 visualizes our contact goals, which directed the broom to align with the dustpan and sweep towards it. Interestingly, our real-world results show a dust sweeping rates of 91-92% from test1 and test2 brooms, which surpassed performance in simulation with unseen tools.

***push_button*:** We positioned three buttons (red, green, blue) in 10 predefined line and triangular arrangements on the desk. For evaluation, we assessed whether the end-effector successfully made contact with the target button. The real-world outcomes aligned with the simulation results for the red and green buttons (training and test1), but pressing the blue button exhibited a performance discrepancy of 20%. Fig. 8 demonstrates the adaptability of our pretrained model to diverse, unseen setups involving variations in elevation and the number of buttons in the scene.

4.3 Compliant Tool Manipulation

In this section, we demonstrate our model performance on a real world, compliant tool manipulation task. By decoupling goal generation and dynamics, our method can generate valid contact goals for tasks using compliant tools, so long as dynamics of the tool are available. We execute the *sweep_to_dustpan* task with a compliant brush (compare tool deformation in Fig. 9 to Fig. 7). Note that the goal generation is learned on rigid tools in CoppeliaSim and transferred without finetuning to a deformable tool in the real-world. We replace rigid dynamics with a learned contact feature dynamics to estimate $\mathbf{P}^{dyn}_{t+i}$ [21]. Full details of the contact feature dynamics can be found in Appendix A.4. Quantitative results are shown in Tab. 2 and a qualitative sweep is shown in Fig. 9. Our method can effectively predict contact goals for a deforming tool, yielding 82% performance.

5 Limitation

Our approach offers versatility in manipulating objects on various planar manipulation scenarios, including elevated and potentially inclined planes. However, our ability to predict contact is limited to a 2D binary contact patch, therefore, it is challenging to directly apply our method for more intricate contact-rich manipulation scenarios like screwing bulbs or peg insertions. Additionally, our approach lacks support for discontinuous contact. As a future work, we suggest integrating binary contact prediction with a contact mask to inform the controller to switch between free space motion and in-contact mode, potentially allowing CALAMARI to handle intermittent contacts.

References

[1] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[2] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

[3] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

[4] A. Zeng, M. Attarian, B. Ichter, K. Choromanski, A. Wong, S. Welker, F. Tombari, A. Purohit, M. Ryoo, V. Sindhwani, J. Lee, V. Vanhoucke, and P. Florence. Socratic models: Composing zero-shot multimodal reasoning with language, 2022. URL https://arxiv.org/abs/2204.00598.

[5] M. Shridhar, L. Manuelli, and D. Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on Robot Learning*, pages 894–906. PMLR, 2022.

[6] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pages 991–1002. PMLR, 2022.

[7] W. Liu, C. Paxton, T. Hermans, and D. Fox. Structformer: Learning spatial structure for language-guided semantic rearrangement of novel objects. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6322–6329, 2022. doi:10.1109/ICRA46639.2022.9811931.

[8] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022.

[9] M. Shridhar, L. Manuelli, and D. Fox. Perceiver-actor: A multi-task transformer for robotic manipulation, 2022. URL https://arxiv.org/abs/2209.05451.

[10] P. Sharma, B. Sundaralingam, V. Blukis, C. Paxton, T. Hermans, A. Torralba, J. Andreas, and D. Fox. Correcting robot plans with natural language feedback, 2022.

[11] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[13] I. Solaiman, M. Brundage, J. Clark, A. Askell, A. Herbert-Voss, J. Wu, A. Radford, G. Krueger, J. W. Kim, S. Kreps, et al. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*, 2019.

[14] C. Lynch, A. Wahid, J. Tompson, T. Ding, J. Betker, R. Baruch, T. Armstrong, and P. Florence. Interactive language: Talking to robots in real time. *arXiv preprint arXiv:2210.06407*, 2022.

[15] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

[16] A. Stone, T. Xiao, Y. Lu, K. Gopalakrishnan, K.-H. Lee, Q. Vuong, P. Wohlhart, B. Zitkovich, F. Xia, C. Finn, et al. Open-world object manipulation using pre-trained vision-language models. *arXiv preprint arXiv:2303.00905*, 2023.

[17] S. Kim, D. K. Jha, D. Romeres, P. Patre, and A. Rodriguez. Simultaneous tactile estimation and control of extrinsic contact. *arXiv preprint arXiv:2303.03385*, 2023.

[18] B. Aceituno and A. Rodriguez. A hierarchical framework for long horizon planning of object-contact trajectories. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 189–196. IEEE, 2022.

[19] A. Sipos and N. Fazeli. Multiscope: Disambiguating in-hand object poses with proprioception and tactile feedback, 2023.

[20] M. Wilson and T. Hermans. Learning to manipulate object collections using grounded state representations. In *Conference on Robot Learning*, pages 490–502. PMLR, 2020.

[21] M. Van der Merwe, D. Berenson, and N. Fazeli. Learning the dynamics of compliant tool-environment interaction for visuo-tactile contact servoing. In *Conference on Robot Learning*, pages 2052–2061. PMLR, 2023.

[22] M. V. der Merwe, Y. Wi, D. Berenson, and N. Fazeli. Integrated object deformation and contact patch estimation from visuo-tactile feedback, 2023.

[23] Y. Wi, A. Zeng, P. Florence, and N. Fazeli. Virdo++: Real-world, visuo-tactile dynamics and perception of deformable objects. *arXiv preprint arXiv:2210.03701*, 2022.

[24] Y. Wi, P. Florence, A. Zeng, and N. Fazeli. Virdo: Visio-tactile implicit representations of deformable objects. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 3583–3590. IEEE, 2022.

[25] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani, et al. Transporter networks: Rearranging the visual world for robotic manipulation. In *Conference on Robot Learning*, pages 726–747. PMLR, 2021.

[26] H. Ha and S. Song. Semantic abstraction: Open-world 3d scene understanding from 2d vision-language models. In *Conference on Robot Learning*, 2022.

[27] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[28] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[29] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5628–5635. IEEE, 2018.

[30] C. Lynch and P. Sermanet. Grounding language in play. *arXiv preprint arXiv:2005.07648*, 40: 105, 2020.

[31] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou. Information theoretic mpc for model-based reinforcement learning. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1714–1721. IEEE, 2017.

[32] E. Rohmer, S. P. Singh, and M. Freese. V-rep: A versatile and scalable robot simulation framework. In *2013 IEEE/RSJ international conference on intelligent robots and systems*, pages 1321–1326. IEEE, 2013.

[33] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

[34] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.