2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 979-8-3503-7770-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/IROS58592.2024.10802849

Uncertainty-Aware Deployment of Pre-trained Language-Conditioned Imitation Learning Policies

Bo Wu[†], Bruce D. Lee[†], Kostas Daniilidis[†], Bernadette Bucher[†,‡], Nikolai Matni[†]

***Abstract*— Large-scale robotic policies trained on data from diverse tasks and robotic platforms hold great promise for enabling general-purpose robots; however, reliable generalization to new environment conditions remains a major challenge. Toward addressing this challenge, we propose a novel approach for uncertainty-aware deployment of pre-trained language-conditioned imitation learning agents. Specifically, we use temperature scaling to calibrate these models and exploit the calibrated model to make uncertainty-aware decisions by aggregating the local information of candidate actions. We implement our approach in simulation using three such pre-trained models, and showcase its potential to significantly enhance task completion rates. The accompanying code is accessible at the link: https://github.com/BobWu1998/uncertainty_quant_all.git**

I. Introduction

Inspired by advancements in general-purpose natural language processing and computer vision models, the robotics community has invested considerable effort in developing generalist decision making agents that operate across various robotic platforms and perform a broad spectrum of tasks [1]. These so-called "foundation models" are typically trained once on a large and diverse dataset, then deployed in a specific setting with minimal task-specific fine-tuning. Such pre-trained models have been shown to generalize in many ways such as to new tasks, new robot platforms, and new environments [2–5].

However, the extent of these agents' generalization abilities remains poorly understood. A significant barrier hindering the generalization abilities and our understanding of them is the lack of well-calibrated, task-specific uncertainty estimates. This gap precludes the possibility of making uncertainty-aware decisions. Our work aims to close this gap by proposing a calibration enabled uncertainty-aware decision making protocol to boost the success rate for language-conditioned imitation learning (IL) policies.

A. Contributions

We propose a straightforward yet effective modification to enhance the generalization capabilities of pre-trained language-conditioned imitation learning agents for robotic manipulation. The modification is composed of two components, outlined in Figure 1:

- A calibration[1] step, tailored to the task of interest, that refines the model's outputs to generate confidence scores that approximate the correctness likelihood for imitating the expert.
- An uncertainty-aware action selection technique using the probability distribution output by the calibrated model (See Algorithm 1).

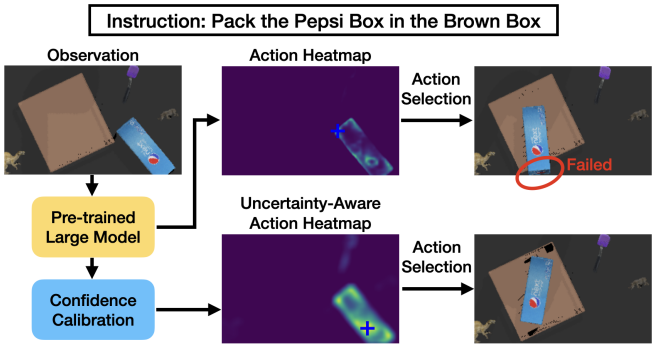


Fig. 1: Overview of uncertainty-aware action selection on top of pre-trained models. The blue cross indicates the maximum score in the heatmap. The upper path illustrate the standard deployment of pre-trained IL policies at test time in which the output of the pre-trained model directly predicts the action. The bottom path indicates our proposed approach, in which the model is calibrated using a small dataset of expert demonstrations from the task of interest before selecting the action in an uncertainty-aware manner.

Each of these steps is novel, and may be of independent interest. However, we view our primary contribution as the holistic integration of these two components into the decision making pipeline for language-conditioned IL policies.

We instantiate our approach on several robotic manipulation models, including the Perceiver-Actor model [7], the RVT model [8], and the CLIPort model [9] *without additional training*, and show that using a small calibration dataset of expert demonstrations from the target task can lead to a substantial improvement in task completion rate. Through comprehensive experimentation, we identify scenarios where integrating uncertainty quantification into the decision making process is most beneficial. Our experiments also find that existing pre-trained models provide poorly calibrated likelihood estimates when instantiated on any particular task of interest. Therefore, appropriately calibrating the model for the task of interest is essential to reap the rewards of uncertainty-aware decision making. An extended manuscript with additional details is available on arXiv [10].

B. Related Work

a) Generalist Robotic Policies: Early efforts [2] introduce a diverse and extensive robotics dataset, and use this dataset to train visual foresight and supervised inverse

[†]GRASP Lab, University of Pennsylvania: {bobwu, brucele, kostas, nmatni}@seas.upenn.edu
[‡]Robotics Department, University of Michigan: bucherb@umich.edu
BL and NM gratefully acknowledge support from NSF Award SLES 2331880, NSF CAREER award ECCS 2045834, and NSF EECS 2231349.
[1]We refer to calibration in the statistical sense [6].

models. By conditioning IL agents with a description of the task to complete, subsequent work uses such diverse datasets to enable sample-efficient IL [3, 11–13]. Recent developments have embraced transformer-based architecture to train multi-modal decision making agents [14, 15] and language-conditioned IL policies for robotic manipulation [4, 5, 7–9]. Efforts have also been made to extend to more general prompt conditioning using images and videos [16]. In Shah et al. [17], the authors explore the development of a generalist visual navigation model. For a review of generalist decision making agents, see Yang et al. [1]. Our work improves the performance of these generalist robotic policies by incorporating calibrated uncertainty estimates to enable more informed decisions at inference time.

b) Uncertainty Quantification: For robotic decision making, confidence scores quantifying the model output uncertainty enable downstream formulations of safe decisions in robotic planning problems [18–20]. However, it has been shown that the confidence scores output by modern classification models are poorly calibrated in that they are not representative of the true correctness likelihood [6]. While numerous approaches to improving calibration post-hoc exist, such as temperature scaling [6] and conformal prediction [21], these techniques are seldom applied in the realm of pre-trained language-conditioned imitation learning policies to enhance generalization performance. A notable exception is the recent work by Ren et al. [22] which applies conformal prediction to language-based planners to request operator clarification in the face of ambiguity. In contrast, we consider a setting in which there is no ambiguity in the task-conditioning, but rather the pre-trained model is imperfect. We calibrate the uncertainty of this pre-trained model, and use it to select actions in an uncertainty-aware manner. For further discussion of uncertainty quantification of foundation models for robotic decision making, see Firoozi et al. [23].

II. Approach

A. Background: Language-conditioned IL for Robotic Manipulation

We consider robotic manipulation problems in a discrete-time decision making context. Let $o_t \in \mathcal{O}$ be the observation of the world at time t and $a_t \in \mathcal{A}$ be an action applied to the robot at time t. Here, $\mathcal{A}$ is a finite, discretized action space.

Language-conditioned IL assumes access to n expert demonstrations $\{\zeta_1, \ldots, \zeta_n\}$ accompanied by language-conditioning $\{c_1, \ldots, c_n\}$. Each demonstration consists of a trajectory of observations, and the corresponding actions taken by an expert. The language-conditioning c_i provides information about the task which the corresponding expert demonstration is completing. Using these demonstrations, the learner searches for a policy which imitates the expert behavior for the given language conditioning.

As the action space is discrete, the policy operates as a classifier for the observation-language pair. In particular, given some observation $o \in \mathcal{O}$ and language-conditioning c, the policy selects an action as

$$\hat{\pi}(o, c) = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\, \hat{f}_{j(a)}(o, c). \tag{1}$$

Here, $j : \mathcal{A} \to [|\mathcal{A}|]$ is a function enumerating $\mathcal{A}$, and $\hat{f}$ is a function which takes in an observation o and a language conditioning c to output a vector of logits $\hat{f}(o, c)$ with $\hat{f}_{j(a)}(o, c)$ representing its $j(a)^{\text{th}}$ element.

A policy of the form mentioned above may be determined from the expert demonstrations and language-conditioning using behavior cloning. In particular, the learner may choose $\hat{f}$ belonging to some function class to minimize the cross entropy loss, as in Brohan et al. [4] and Shridhar et al. [7]. The resultant function, $\hat{f}$, serves as a pre-trained model, which can be deployed on numerous different tasks. In particular, given some new language conditioning c_{n+1}, it is deployed to select the robot's actions as $a_t = \hat{\pi}(o_t, c_{n+1})$. In general, the target task need not be one of the tasks used for training; we may have $c_{n+1} \notin \{c_1, \ldots, c_n\}$. However, many of our experimental results are focused on the setting where $c_{n+1} \in \{c_1, \ldots, c_n\}$.

Pre-trained models using the aforementioned strategy to select an action neglect pertinent information output by the model about the distribution of potential actions. This is in part because such pre-trained models tend to be poorly calibrated and fail to reflect the actual correctness likelihood of prediction matching the expert demonstration. Accordingly, we address these problems by calibrating the model and then following an uncertainty-aware action selection strategy, which is discussed in Section II-B and Section II-C.

B. Target Task Calibration

The logits can be viewed as a representation of a probability distribution p over possible actions by applying the softmax function σ_{SM} to the output of the model $\hat{f}$: $p = \sigma_{SM}(\hat{f}(o, c))$. While p is a viable probability distribution, it may not be well-aligned with the true correctness likelihood for imitating the expert, especially for the specific task c_{n+1}.

To improve the alignment between the distribution p and the correctness likelihood for task c_{n+1}, we collect a calibration set from task c_{n+1}. In particular, we assume that the target language-conditioning c_{n+1} is drawn randomly from some distribution $c_{n+1} \sim \mathcal{D}_{\mathsf{test}}$, and that we have access to a calibration set of expert demonstrations $\{\zeta_1^{\mathsf{cal}}, \ldots, \zeta_k^{\mathsf{cal}}\}$ along with their associated conditioning $\{c_1^{\mathsf{cal}}, \ldots, c_k^{\mathsf{cal}}\}$, where c_i^{cal} are drawn from the same distribution as c_{n+1}: $c_i^{\mathsf{cal}} \sim \mathcal{D}_{\mathsf{test}}$. Calibrated models may be obtained from this set with various methods, but we propose using temperature scaling [6]. Specifically, we determine the optimal temperature parameter $\hat{T}$ as the value of T which minimizes the cross entropy loss of the model $\frac{1}{T}\hat{f}$ over the calibration set. Given this temperature parameter, we define the calibrated model outputting the probability scores as $\tilde{p} = \tilde{f}(o, c) = \sigma_{SM}\left(\frac{\hat{f}(o,c)}{\hat{T}}\right)$, with components $\tilde{p}_k = \tilde{f}_k(o, c)$.

C. Uncertainty-Aware Action Selection

The greedy action selection approach of equation (1) does not make use of the calibrated confidence scores from

Model	Baseline	Uncalibrated UA	Calibrated UA
PerAct	0.382 ± 0.012	0.385 ± 0.012	$\mathbf{0.414 \pm 0.012}$
RVT	0.602 ± 0.012	0.607 ± 0.012	$\mathbf{0.623 \pm 0.011}$
CLIPort	0.803 ± 0.005	$\mathbf{0.833 \pm 0.005}$	$\mathbf{0.833 \pm 0.005}$

TABLE I: Summary of results for calibrated and uncalibrated uncertainty-aware approach and the uncertainty oblivious (baseline) approach on PerAct, RVT, and CLIPort. For PerAct and RVT, the values in the table represent the portion of successfully completed tasks. For CLIPort, the values represent the mean reward, where the maximum reward for an episode is one if the desired task is completed successfully. Standard errors are shown.

Section II-B, and overlooks important structure of the action space. Specifically, distinct elements in $\mathcal{A}$ may represent similar actions. This structure should be leveraged in the manipulation problem, as many tasks can be completed with distinct, but similar actions, e.g. a robotic gripper can open a drawer by pulling at different positions of the handle.

We propose Algorithm 1 to remedy the above shortcoming. Rather than selecting the action with the highest confidence score, Algorithm 1 selects the action for which the neighboring region of potential actions has the highest sum of confidence scores. The neighboring region of an action $a \in \mathcal{A}$ is defined as $\{a' : d(a, a') \leq \tau\}$, where the metric d is selected based on the problem at hand to capture similarity between two actions in $\mathcal{A}$, and τ is a similarity threshold. While both the metric d and the threshold τ are hand-designed in this work, they could be chosen in a data-driven manner in future work. Note that while the greedy action selection approach in equation (1) is only sensitive to the ordering of highest confidence score output by the model, Algorithm 1 needs a well-calibrated model to adequately determine the confidence of the neighboring region.

Algorithm 1 Uncertainty-Aware Action Selection

1: **Input:** Calibrated model $\tilde{f}$, conditioning c, observation o, robustness threshold τ
2: Compute $\tilde{p}_1, \ldots, \tilde{p}_{|\mathcal{A}|} = \tilde{f}(o, c)$.
3: Set $\hat{a} = \text{argmax}_{a \in \mathcal{A}} \sum_{a' \in \mathcal{A}} \tilde{p}_{j(a')} \mathbf{1}(d(a, a') < \tau)$
4: **Output:** Uncertainty-aware action $\hat{a}$

In Algorithm 1, our method assumes to take a calibrated model $\tilde{f}$, a language conditioning of the task c, the observation o, and the robustness threshold τ as input. At each time step, the calibrated model $\tilde{f}$ produces a vector of logits $\tilde{p}$, based on task conditioning and observation. Then, the approach calculates the uncertainty-aware score for the actions by aggregating the neighboring confidence scores of each candidate, where two actions a and a' are considered "neighbors" if $d(a, a') < \tau$. Finally, the model makes the decision about the action based on the highest confidence score produced in the previous step.

Note that the approach for selecting the action from the calibrated model in Line 3 of Algorithm 1 is just one possibility for aggregating the confidence of neighboring actions. An alternative is to convolve the output of the calibrated model with a Gaussian kernel. Another alternative is to construct prediction sets using conformal prediction, and then select a representative from the set.

A key feature of the proposed approach is that the weights of the pre-trained model are not updated, as it can be both computationally and statistically prohibitive to fine-tune the model. As a result, the efficacy of an agent using the proposed approach is still bottle-necked by the quality of the pre-trained agent. We therefore expect improvements in the performance from leveraging uncertainty-aware decision making to be limited to some degree by the quality of the pre-trained model. In our experiments, we highlight several cases where the performance boost is most notable, and offer explanations for why this may be the case.

III. Experimental Results

We evaluate the method discussed in Section II by deploying it on three pre-trained models: the PerAct model [7], the RVT model [8], and the CLIPort model [9]. PerAct and RVT are evaluated on the RLBench dataset [24], while CLIPort is evaluated in the Ravens benchmark [25]. Each of these datasets breaks down the desired behavior of the robot into tasks. A task consists of a small collection of related language conditionings. For example, the RLBench task "stack blocks" consists of language-conditionings:[2]

- "place three of the blue cubes on top of each other"
- "build a tall tower out of four green cubes"
- "place two of the red cubes on top of each other"

Within each task for both the RLBench dataset and the Ravens dataset, the variations of language-conditionings are sampled uniformly at random from the set of task-specific variations built into the environments. We therefore calibrate the models for particular tasks by obtaining expert demonstrations corresponding to language-conditionings sampled uniformly at random from a particular task.

PerAct and RVT were trained on a dataset consisting of 100 expert demonstrations for each of 18 distinct RLBench tasks. We evaluate our approach on each model on all 18 of the tasks they were trained on. For each task, we calibrate the model using temperature scaling with a calibration set of 25 expert demonstrations, and evaluate on 100 episodes.

CLIPort was trained on 1000 demonstrations from each of 9 distinct tasks from the Ravens dataset. We evaluate our approach on four of the nine tasks that the model was trained on (discussed in Section III-A). On each task, we calibrate the model on 100 expert demonstrations using temperature scaling. Evaluations are performed on 1000 episodes.

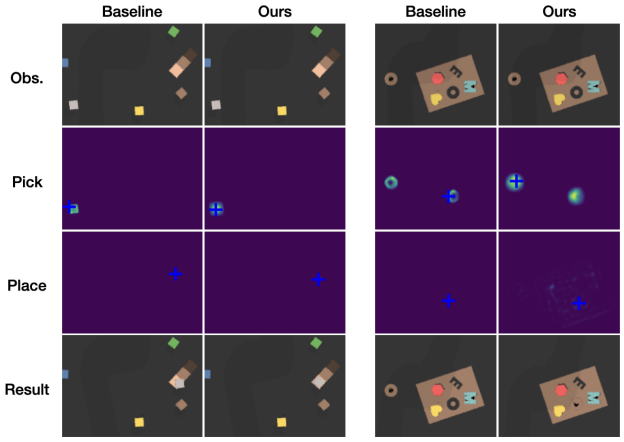
The results from running uncertainty-aware action selection on the three models are summarized in Table I. We see that the uncertainty-aware approach offers a benefit to the overall task completion rate for all models. To investigate exactly when the approach provides a benefit, the subsequent sections study three pertinent questions, summarized below.

1) **Section III-A: Why does uncertainty-aware action selection help?** Uncertainty-aware action selection prevents the end effector from moving to positions described by isolated, high-confidence locations output

[2]See [24] for all language-conditionings of all tasks.

Task	Baseline	UA
stack-block-pyramid-seq seen-colors	0.9475 ± 0.0047	$\mathbf{0.9815 \pm 0.0027}$
packing-seen-google objects-seq	0.8090 ± 0.0104	$\mathbf{0.8275 \pm 0.0100}$
packing-seen-google objects-group	0.8846 ± 0.0080	$\mathbf{0.9108 \pm 0.0072}$
assembling-kits-seq seen-colors	0.5706 ± 0.0116	$\mathbf{0.6134 \pm 0.0115}$
Task with Distractors	**Baseline**	**UA**
stack-block-pyramid-seq seen-colors-diff-sizes	0.7963 ± 0.0073	$\mathbf{0.9711 \pm 0.0031}$
assembling-kits-seq seen-colors-diff-sizes	0.4832 ± 0.0028	$\mathbf{0.5544 \pm 0.0030}$

TABLE II: Task-specific mean rewards for CLIPort on four tasks from the original Ravens dataset and two tasks with small distractors added to the original tasks.



(a) Put Gray Block on the Lightest Brown Block

(b) Put the Brown Ring in the Brown Box

Fig. 2: Qualitative results for CLIPort with and without uncertainty-aware action selection. The language commands are in the subfigure captions. From top to bottom, each row represents the model's observation, the heatmap in pick phase, the heatmap in place phase, and the observation after executing the actions. In 2a, our method forces the model to choose a central location on the target object instead of the edges. In 2b, our method corrects the model by smoothing out the action with highest raw score.

by the pre-trained model. This can prevent the gripper from completely missing the object, and encourages a better grip when it is successful in gripping the object.

2) **Section III-B: Is calibration necessary?** It depends on the model. For simple models trained on very large datasets (CLIPort), the pre-trained model appears to be fairly well calibrated, and calibration is not necessary to achieve the benefit of uncertainty-aware decisions. Both PerAct and RVT are poorly calibrated on any given task, and calibrating the model is critical to achieve the benefit of uncertainy-aware decisions.
3) **Section III-C: Can uncertainty-awareness enable generalzation to new settings?** We demonstrate that small distractor objects to the scene can have a devastating performance of the pre-trained model. However, uncertainty-aware action selection achieves essentially the same performance it does in the absence of distractors. This provides a strong indication that accounting for uncertainty-aware action selection can improve generalization to certain task distribution shifts.

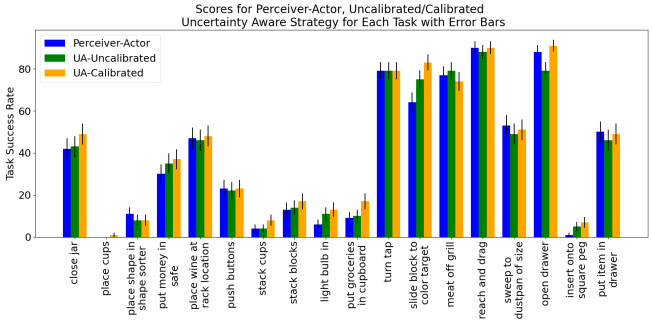


Fig. 3: Success rate score for Perceiver-Actor, uncertainty-aware strategy with calibrated/uncalibrated confidence for each task

A. *Why Does Uncertainty-Aware Action Selection Help?*

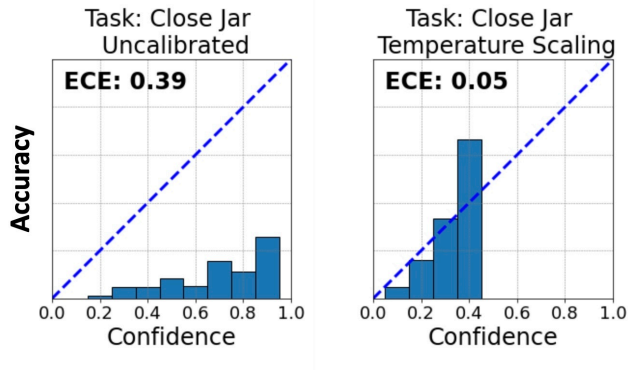
To investigate why uncertainty-aware action selection helps, we first look at the per-task success rates of the CLIPort model. In the first four rows of Table II, we compare uncertainty-aware action selection with the baseline on four tasks from the Ravens dataset. From the results, we find that incorporating uncertainty is particularly useful when the CLIPort task requires precise placement of target objects. This is important for stacking blocks, packing objects, and assembling kits. The reason that uncertainty awareness is beneficial in these settings is that it enables conservative decisions in the picking phase by encouraging the end effector to pick up the object at the center instead of at the edges, as illustrated in Figure 1 and Figure 2a.

In these examples, our method takes the calibrated score map of the baseline (left column of the second row), and generates uncertainty-aware score calculation (right column of the second row). Conditioned on the more conservative picking location, the agent picks up the object more easily and finds a better placing location, as shown in the third row. The bottom row of each step shows the result of the manipulation at the current time step.

Additionally, by smoothing confidence scores across possible actions, the uncertainty-aware approach prevents the end effector from selecting isolated high-confidence actions. The benefit of this is illustrated in the second row of Figure 2b.

The per task success rates of PerAct are shown in Figure 3. Due to the higher complexity of the tasks from RLBench and the higher standard error in the results, it is difficult to draw concrete conclusions about when the benefit of accounting for uncertainty is most important. However, we do see the most pronounced improvement on tasks with a lower success rate. This benefit appears to be due to the larger discrepancy between the action with the highest confidence score and the action with the largest sum of neighboring confidence scores on tasks for which the model is overall less confident.

In light of the above experimental results, we conclude that by encouraging the model to select actions with higher neighboring confidence scores, our method prevents the model from choosing aggresive action candidates, thereby resulting in better grasping performance.



(a) uncalibrated (b) Calibrated

Fig. 4: Reliability diagrams are shown for PerAct on two of the RLBench tasks. Figure 4a illustrates the reliability diagrams before calibration, while Figure 4b shows the reliability diagrams after running temperature scaling on a calibration set of 25 expert demonstrations from the task. Overlayed on the plot is the value for the expected calibration error (ECE, see Guo et al. [6] for further details). We see that temperature scaling provides a marked improvement in the calibration.

B. Do we need calibration?

We see in Table I that for both PerAct and RVT, the uncertainty-aware action provides only a very small improvement in the absence of calibration (0.3% and 0.5% boost in task success rate, respectively); however, they yield a significant task success rate if a calibration step is included (3.2% and 2.1% boosts, respectively).

To understand why this is the case, we construct reliability diagrams in Figure 4. The figure depicts this reliability diagram for PerAct on two tasks, with both the uncailbrated model and the model calibrated using temperature scaling. The y-axis is the model accuracy (the correctness likelihood) at each confidence level, which is the number of predicted actions that are the same as the expert demonstration, divided by the total number of predictions made with confidence within the thresholding interval. The x-axis depicts the model confidence, which is maximum value in the probabilty distribution representing the calibrated model: $\max_{a \in \mathcal{A}} \tilde{f}_a(o, c)$. The confidence is perfectly calibrated when the accuracy and the confidence are linearly related with slope of one. For more details on reliability diagrams, see Guo et al. [6]. These diagrams illustrate the value of temperature scaling for improving model calibration using a calibration set of 25 expert demonstrations from the task of interest. From these results, we conclude that PerAct has very poorly calibrated uncertainty measurements, but applying temperature scaling using a small number of demonstrations from a specific task improves calibration substantially.

The importance of this calibration is further reflected by the improvement offered by the calibrated uncertainty-aware approach over the uncalibrated uncertainty-aware approach in Table I. As in Table I, RVT benefits from temperature scaling. However, CLIPort does not see a benefit of temperature scaling in improving the success rate of uncertainty-aware action selection. We find that the calibrating cross-entropy loss barely changes during from calibration, indicating that the pre-trained model is already fairly well-calibrated for each of the individual tasks.

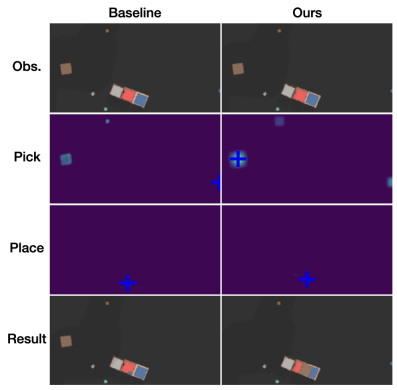


Fig. 5: Qualitative results of CLIPort on instruction "Put the brown block on the blue and red blocks" when distractors are introduced.

C. Can Uncertainty-Awareness Improve Generalization?

To determine whether uncertainty-aware decisions can help pre-trained models generalize to out-of-distribution settings, we designed two new tasks. One of these is a modification of stack blocks, and the other one is a modification of the assemble kits task. Both of the new tasks introduce a collection of smaller-sized target objects of various colors and shapes, which serve as distractors to the model. In the last two rows of Table II, we find that the pre-trained method is very fragile to such distractors, and they cause a massive drop in performance from 94.75% completion to 79.63% for stacking blocks, and from 57.06% to 48.32% for assembling kits. However, the uncertainty-aware approach is robust to such distractors: performance degrades only from 98.15% and 61.34% completion to 97.11% and 55.44%, respectively.

In Figure 5, we demonstrate the impact of the distractors on the pre-trained model. We see that the precense of distractors causes the pre-trained model to pick up one of the distractor blocks which has a single pixel of high confidence. The uncertainty-aware approach smooths the heatmap, thereby determining that the isolated high confidence event is due to a distractor, and that the best course of action is to select a region where the model output a collection of neighboring actions with high confidence.

More generally, we believe these results indicate the potential value of incorporating uncertainty estimates for decision making in out-of-distribution scenarios.

IV. Conclusion

In this paper, we proposed the use of simple classification model calibration techniques in conjunction with an uncertainty-aware action selection approach to improve the success rates for language-conditioned imitation learning in robotic manipulation. Our results suggest that by appropriately calibrating the model and incorporating an uncertainty-aware decision making strategy, we can benefit the pre-trained "foundation models" in both seen and unseen tasks. This approach does not require any additional training or

fine-tuning of the original model and is scalable to versatile robot manipulation models and platforms.

REFERENCES

[1] S. Yang, O. Nachum, Y. Du, J. Wei, P. Abbeel, and D. Schuurmans, "Foundation models for decision making: Problems, methods, and opportunities," *arXiv preprint arXiv:2303.04129*, 2023.

[2] S. Dasari, F. Ebert, S. Tian, S. Nair, B. Bucher, K. Schmeckpeper, S. Singh, S. Levine, and C. Finn, "Robonet: Large-scale multi-robot learning," *arXiv preprint arXiv:1910.11215*, 2019.

[3] F. Ebert, Y. Yang, K. Schmeckpeper, B. Bucher, G. Georgakis, K. Daniilidis, C. Finn, and S. Levine, "Bridge data: Boosting generalization of robotic skills with cross-domain datasets," *arXiv preprint arXiv:2109.13396*, 2021.

[4] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu *et al.*, "Rt-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022.

[5] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," *arXiv preprint arXiv:2307.15818*, 2023.

[6] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International conference on machine learning*. PMLR, 2017, pp. 1321–1330.

[7] M. Shridhar, L. Manuelli, and D. Fox, "Perceiver-actor: A multi-task transformer for robotic manipulation," in *Conference on Robot Learning*. PMLR, 2023, pp. 785–799.

[8] A. Goyal, J. Xu, Y. Guo, V. Blukis, Y.-W. Chao, and D. Fox, "Rvt: Robotic view transformer for 3d object manipulation," *arXiv preprint arXiv:2306.14896*, 2023.

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

[10] B. Wu, B. D. Lee, K. Daniilidis, B. Bucher, and N. Matni, "Uncertainty-aware deployment of pre-trained language-conditioned imitation learning policies," *arXiv preprint arXiv:2403.18222*, 2024.

[11] Y. Ding, C. Florensa, P. Abbeel, and M. Phielipp, "Goal-conditioned imitation learning," *Advances in neural information processing systems*, vol. 32, 2019.

[12] S. Stepputtis, J. Campbell, M. Phielipp, S. Lee, C. Baral, and H. Ben Amor, "Language-conditioned imitation learning for robot manipulation tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 13 139–13 150, 2020.

[13] C. Lynch and P. Sermanet, "Language conditioned imitation learning over unstructured data," *arXiv preprint arXiv:2005.07648*, 2020.

[14] 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.

[15] 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.

[16] Y. Jiang, A. Gupta, Z. Zhang, G. Wang, Y. Dou, Y. Chen, L. Fei-Fei, A. Anandkumar, Y. Zhu, and L. Fan, "Vima: General robot manipulation with multimodal prompts," *arXiv preprint arXiv:2210.03094*, 2022.

[17] D. Shah, A. Sridhar, N. Dashora, K. Stachowicz, K. Black, N. Hirose, and S. Levine, "Vint: A foundation model for visual navigation," *arXiv preprint arXiv:2306.14846*, 2023.

[18] L. Lindemann, M. Cleaveland, G. Shim, and G. J. Pappas, "Safe planning in dynamic environments using conformal prediction," *IEEE Robotics and Automation Letters*, 2023.

[19] A. Dixit, L. Lindemann, S. X. Wei, M. Cleaveland, G. J. Pappas, and J. W. Burdick, "Adaptive conformal prediction for motion planning among dynamic agents," in *Learning for Dynamics and Control Conference*. PMLR, 2023, pp. 300–314.

[20] J. Lekeufack, A. A. Angelopoulos, A. Bajcsy, M. I. Jordan, and J. Malik, "Conformal decision theory: Safe autonomous decisions from imperfect predictions," *arXiv preprint arXiv:2310.05921*, 2023.

[21] G. Shafer and V. Vovk, "A tutorial on conformal prediction." *Journal of Machine Learning Research*, vol. 9, no. 3, 2008.

[22] A. Z. Ren, A. Dixit, A. Bodrova, S. Singh, S. Tu, N. Brown, P. Xu, L. Takayama, F. Xia, J. Varley *et al.*, "Robots that ask for help: Uncertainty alignment for large language model planners," *arXiv preprint arXiv:2307.01928*, 2023.

[23] R. Firoozi, J. Tucker, S. Tian, A. Majumdar, J. Sun, W. Liu, Y. Zhu, S. Song, A. Kapoor, K. Hausman *et al.*, "Foundation models in robotics: Applications, challenges, and the future," *arXiv preprint arXiv:2312.07843*, 2023.

[24] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison, "Rlbench: The robot learning benchmark & learning environment," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3019–3026, 2020.

[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*. PMLR, 2021, pp. 726–747.