Skill Generalization with Verbs

Rachel Ma[1*], Lyndon Lam[2], Benjamin A. Spiegel[1], Aditya Ganeshan[1],
Roma Patel[3], Ben Abbatematteo[1], David Paulius[1], Stefanie Tellex[1], George Konidaris[1] .

***Abstract*— It is imperative that robots can understand natural language commands issued by humans. Such commands typically contain verbs that signify what action should be performed on a given object and that are applicable to many objects. We propose a method for generalizing manipulation skills to novel objects using verbs. Our method learns a probabilistic classifier that determines whether a given object trajectory can be described by a specific verb. We show that this classifier accurately generalizes to novel object categories with an average accuracy of 76.69% across 13 object categories and 14 verbs. We then perform policy search over the object kinematics to find an object trajectory that maximizes classifier prediction for a given verb. Our method allows a robot to generate a trajectory for a novel object based on a verb, which can then be used as input to a motion planner. We show that our model can generate trajectories that are usable for executing five verb commands applied to novel instances of two different object categories on a real robot.**

I. Introduction

Robots that interact with humans should be equipped with the means to interpret and follow commands in natural language. Manipulation commands are commonly expressed as verbs applied to a given object. We therefore propose that robots that can efficiently learn how to perform various manipulation tasks from natural language commands must be able to generate a motor skill that matches a given verb, and apply it to manipulate a novel object. For instance, opening a door is similar to opening a microwave; therefore, a robot that has learned a skill appropriate for the verb "open" applicable to a door should be able to: 1) know what an "open" microwave looks like given a "closed" microwave, and 2) execute the "open" action on a microwave with minimal additional learning. However, most works in natural language grounding and generalization either do not apply multiple actions across multiple object categories [1, 2, 3], assume robots know goal states for primitive verbs [4, 5], or rely on demonstration data [5, 6, 4, 7, 8, 9].

To address the problem of generalizing verb-labeled skills to novel object categories, we propose a model[†] with two components—a classifier and an optimizer—for producing object trajectories given images of an object to manipulate and a desired verb to execute. The first component is a classifier for the trajectory identification task, where the goal is to identify the correct verb given image snapshots of an object trajectory, allowing us to exploit the effectiveness of neural networks for vision-based tasks. This classifier is used by the second component, a policy search algorithm that generates a trajectory from the initial state of an object (before the verb is applied) to a final state (after the verb has been applied), so as to maximize the verb probability estimated by the classifier.

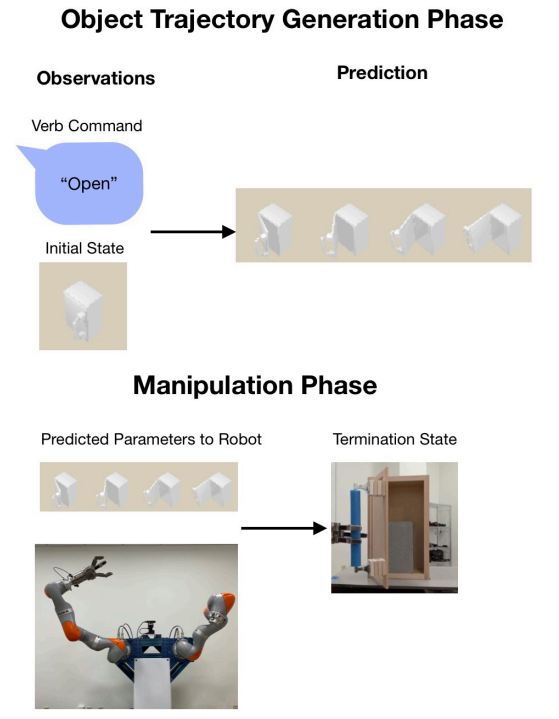


Fig. 1. Given an observation of an object (before the verb is applied) and a desired verb command, our model generates an object trajectory.

We applied our approach to a list of verbs obtained from VerbNet [10] and objects from the PartNet-Mobility dataset in SAPIEN [11]. We observed an average accuracy of 76.69% when generalizing to an unseen category across 13 object categories and 14 verbs. Along with this classifier, our trajectory optimizer generates plausible object trajectories. To show the efficacy of our model, we show that it can generate trajectories that are usable by a real robot (shown in Figure 1), through demonstrations with the KUKA LBR iiwa7 robot executing five verb commands applied to novel object instances of two different object categories.

[1]Brown University, Providence, RI, USA.
[2]California State Polytechnic University, Pomona, CA, USA.
[3]DeepMind, work done at Brown University, Providence, RI, USA
*Corresponding Author (*Email*: rachelm8@mit.edu)
[†]Code, dataset info and demo videos can be found at: https://rachelma80000.github.io/SkillGenVerbs/

II. Background and Related Work

A. Options Framework

Manipulation tasks are commonly modeled as Markov Decision Processes (MDP) [12], where given a task, a robot chooses an action and updates its action policy based on reward given its observations of the world. We are particularly motivated by the work of Rodriguez-Sanchez and Patel [13], which suggests that verbs ground to actions in MDPs. We focus on specifically grounding verbs to motor skills, via the options framework [14]. Each motor skill is modeled by an option o that consists of three components: the option policy π_o, which is executed and maps low-level states to low-level actions; the initiation set I_o, which describe states where the option can be executed; and the termination condition β_o, which is the probability of the option terminating in each state [12].

We apply these concepts to develop a model for generating an object trajectory that achieves the intended goal of a verb given visual input of the object. Given the example of applying the verb "open" to a door, the initiation set would be the current image of the door, which captures the notion of the door in a closed state. The termination set would be an image of the final state of the door, which captures the notion of a door in its open state—specifically, an image of the door angled ajar. The option policy generates a predicted trajectory of the object when the verb is applied.

B. Verbs

Verbs play a crucial role in natural language commands. Hovav and Levin [15] propose that verbs can be classified as *manner verbs* that specify the manner of carrying out the action, and *result verbs*, that specify the reaching of a resulting state. Result verbs can be classified further into three categories: *change of state verbs*, which specify a change of state of a property of the object the verb is applied to, *inherently directed motion* verbs, which contain movement in relation to an object, and *incremental theme* verbs, which specify a change in volume or area of object. Following Gao et al. [16], and unlike Hovav and Levin, we consider changes of location, volume, and area to also constitute "change of state".

To realize the effects of *change in state verbs*, we propose that at least one of the following is required: termination state (the state of the object after the verb is applied), both initiation (the state of the object before the verb is applied) and termination state, object trajectory, and/or robot arm trajectory. Verbs like open and close can be minimally differentiated by termination state. Verbs like rotate can be minimally differentiated by initiation and termination state by looking at the difference of angles between the two steps, while verbs like throw and toss can be minimally differentiated by the combination of object and robot arm trajectories. We focus on *change in state* verbs that can be realized through initiation and termination, and/or object trajectory, where the agent manipulates a physical object.

C. Related Work

Existing work has focused on grounding language in a visual representation of objects in the world and generalizing manipulation skills for robots. We situate our work between these bodies of work and outline them below.

Grounding Language to Vision and Manipulation: Many works assume that robots already know the goal state of a given object after applying primitive/atomic verbs. Ichter et al. [4] and Sharma et al. [5] focused on breaking down complex natural language commands into simpler primitive actions or tasks. Ramesh et al. [17] created a model for text to image generation, which could be useful for predicting images of objects after manipulation. However, given the text prompt "closed oven", such a model produces mostly open ovens, and we have found that it has trouble with differentiating between physical states of an object. Paulius et al. [2] proposed a motion taxonomy for describing action verbs as binary strings known as motion codes, which can be used to describe action and discern between the meaning of actions in a manipulation-centric embedding space. However, they did not account for manipulation on objects across different categories. Other work in natural language and robotics addressed language-conditioned imitation learning [18] or trajectory modification with natural language commands [19, 20, 8]. However, these works do not address generalization of tasks or skills across objects or of skills. Rather, they focus on imitation learning, which requires large amounts of demonstration data, or active parsing and interpretation of commands from humans during a task to accomplish the skill.

Learning Generalizable Skills: Contrary to our work, where we focus on generalizing skills by the effect or action of the verb, a common approach is generalizing through object articulation. Eisner and Zhang [3] proposed a model to learn and predict 3D articulation flow for various objects; this output was then used to execute a motion planner to achieve the maximum articulation. However, there is no explicit integration of language nor mention of multiple verbs or skills being applied to each object instance. Abbatematteo et al. [21] investigated how to estimate the kinematic model and configuration of novel objects for manipulation; however, they do not explore generalization of skills. Hewlett et al. [1], Jang et al. [22], and Sugiura and Iwahashi [9] incorporated the presence of another human in the environment/loop or human demonstration data, which is inefficient and thus limits the amount of verbs that the approach can handle. Furthermore, while Hewlett et al. [1] required a human in the loop of identifying the verb for a demonstrated trajectory, our approach exploits deep neural networks for verb prediction.

III. Skill Generalization With Verbs

We propose a model that generalizes verb-labeled skills to novel object categories. Our goal is to train a model that takes as input a verb, paired with a kinematic model of an object and its initial state, and outputs a trajectory that can be applied on the object to undergo the effect of that verb.

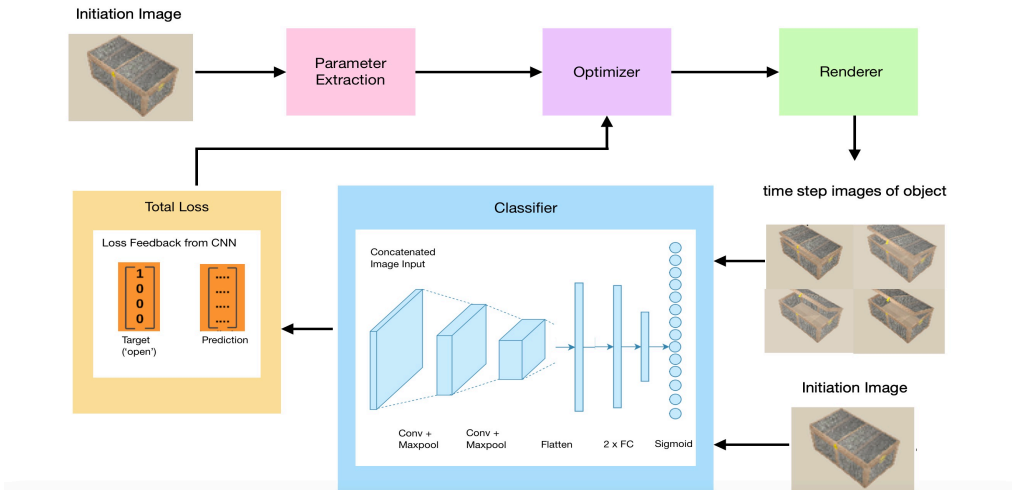


Fig. 2. Diagram of the planning portion of the proposed model. Parameters of the initiation image are extracted and then manipulated by the optimizer, which relies on the categorical cross-entropy loss calculated on the probabilities from the classifier and the target array. Trajectory timestep images (excluding the initiation image) are rendered and given to the classifier, along with the initiation image for computing the loss.

In order to realize these *change of state* verbs, robots must be able to: 1) predict the goal/target state of the object if a verb were to be applied, and 2) adapt verb skills (specifically the object trajectory) across multiple (and potentially novel) object categories with minimal additional learning. To achieve these, the model should be able to realize and differentiate the verb that is depicted for a given trajectory, and be able to change the state of the object to achieve the desired verb. We do this through two main components after extracting kinematic parameters for an object instance: a *classifier* that will output a predicted probability for a given verb command and images of the object trajectory, coupled with an *optimizer* which uses this classifier to generate a trajectory from a given initiation state of a novel object to achieve the verb command. Figure 2 shows an overview of the planning portion of our model pipeline.

A. Classifier

1) Architecture: Our experiments use a simple CNN with the following structure: two CNN layers (32 and 64 units respectively with 3×3 kernel sizes and max pooling layers) and three fully connected layers (64, 32, and N units, where N is the total number of verbs). The inputs to the classifier are sequences of RGB images of a trajectory sequence (where each image has dimensions of $128 \times 128 \times 3$). Each image of a sequence is concatenated together in the network. The final layer is a softmax layer of N units that will output probabilities of the sequence achieving the effects of N verbs; the predicted verb is that with the highest probability value. Other state-of-the-art activity and object recognition methods can be used (refer to Section V).

2) Training: Given k object categories, we perform training and testing in an all-but-one procedure (i.e., k-fold cross validation), where we train the classifier using $k-1$ object categories and test the classifier on a unseen k-th object category. During training, 80 percent of images from the selected object categories will be used as the train set, and 20 percent will be used as the validation set after shuffling. We use the Adam Optimizer for training.

3) Prediction: When given an input RGB trajectory sequence for an object instance of an novel object category, the classifier outputs a per-verb probability array.

4) Accuracy: The accuracy of the classifier is measured by generating predictions on RGB trajectory sequences from the novel test object categories, and comparing the verb label of the most likely prediction with the ground truth verb label associated with the trajectory.

B. Trajectory Optimizer

The second step of our model generates an object trajectory for the desired verb command given a URDF description of the object representing its links and joints. The optimizer searches over trajectories of the degrees of freedom of the object (i.e., 6-DoF pose and articulated state) to maximize the classifier's returned probability for the desired verb. We used the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) optimizer algorithm [23], an efficient state-of-the-art optimization algorithm, to search over object trajectories.

1) Trajectory Parameterization: A candidate trajectory is parameterized by a vector of length $(6+n)$, corresponding to the per-timestep change in 6-DoF position and orientation of the object as well as the state of its n articulated joints. Articulated states are clipped to lie within the joint limits provided in the URDF description. This trajectory is then rendered and scored by the classifier.

2) Measuring Loss: The loss minimized by CMA-ES is calculated as the categorical cross-entropy between the verb probability array produced from our CNN classifier and a one-hot target array indicating the target verb.

C. Verb Selection and Data Collection

To train and evaluate our model, we require visual data of objects undergoing manipulations corresponding to verbs, and their underlying kinematic and geometric descriptions.

1) Visual Data Collection: We require an environment that provides the following: multiple object categories and multiple instances of objects in those categories, the capability to manipulate parts or the entire object for a variety of verbs, and a way to extract multiple images of the scene.

We considered datasets such as ALFRED [24], AI2THOR [25], New Brown Corpus [26], SAPIEN [11], and RL Bench [27]. We chose the PartNet-Mobility Dataset from SAPIEN due to the presence of 2347 object instances over 46 object categories as URDF files. Each file describes the kinematics (links and joints) and geometry of an object. Our approach performs verb-based manipulation of each object by simulating them in SAPIEN and taking RGB snapshots. For each object-verb pairing, we generate 21 RGB images/snapshots (each 128 by 128) representing 21 timesteps along the object trajectory when the verb is applied to the object. We generate these for our chosen verbs applied for a total of 812 object instances present in the 13 chosen object categories: Box, Dishwasher, Door, Laptop, Microwave, Oven, Refrigerator, Safe, Stapler, Storage Furniture, Toilet, Trash Can, and Washing Machine. A total of 41688 trajectories are generated.

We assume maximally distinct initiation and termination states for collecting data needed to train the CNN classifier. For instance, the initiation state for the "open" verb on door is when the door is completely closed, while the termination state for the "open" verb on a door is when the door is open to the upper joint limit. We also assume that for "open" and part-based translation verbs applied to objects with multiple non-fixed joints, only one joint is manipulated. To prevent the model from generating object trajectories where multiple verbs occur simultaneously, data for a "none" category is generated, which features objects undergoing multiple verbs at once in a single trajectory.

2) Verb Selection: Our requirements when selecting verbs are whether they can be achieved via the URDF files of object instances and if they can be applied to multiple object categories that are present in SAPIEN. With these objectives in mind, we examined the verbs in VerbNet for *change in state* verbs. The final selected verbs are: translation verbs (specifically *Push*, *Pull*, *Raise*, *Lower*, *TranslateLeft* and *TranslateRight*, and *RemoveWhole*—when removing an object from the scene), part-based translation (*RemovePart*—when a single part of the object is removed—and *InsertPart*—when a single part of the object is inserted), *Open*, *Close*, and rotation verbs (*Roll*, *Turn*, *Flip* by 270 degrees).

IV. Experiments

The aim of our evaluation is to test the hypothesis that our model can successfully transfer verbs to novel object categories, both in simulation and with a real robot. We do so by selecting object categories for training the classifier, and then selecting a novel object category for evaluation by the classifier and for manipulation with the optimizer.

A. Classifier

We measure the accuracy of the classifier on RGB image trajectories from the test object categories. The object categories that we use for our experiments are: Box, Dishwasher, Door, Laptop, Microwave, Oven, Refrigerator, Safe, Stapler, Storage Furniture, Toilet, Trash Can, and Washing Machine.

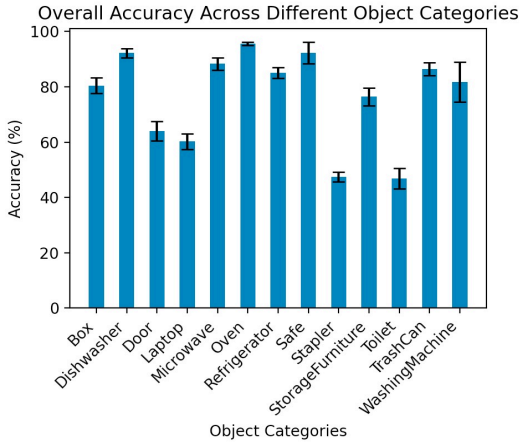


Fig. 3. Overall Verb Accuracy Across Object Categories. We perform $k-$fold cross validation, where there will be $k-1$ object categories used for training the classifier, and a unseen k-th object category reserved for testing. Each of the 13 categories take their turn being the k-th category. "None" verb trajectories are included in training and testing.

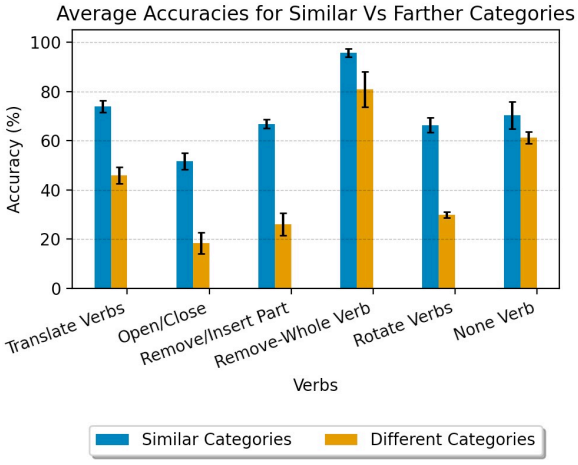


Fig. 4. Average Accuracies for Similar versus Farther Categories. "None" verb trajectories are included in training and testing. For testing the "similar" categories: average of k-fold cross validation across categories of Dishwasher, Door, Microwave, Refrigerator, Safe, Washing Machine, where the k-th category for testing was one of those, and $k-1$ were training categories. The "farther" categories: average of k-fold cross validation when the k-th category for testing is chosen out of Box, Laptop, Stapler, Toilet, and Trash Can, and the training categories were Dishwasher, Door, Microwave, Refrigerator, Safe, Washing Machine.

The snapshots selected for use are at regular intervals. We noticed a general increase in accuracy as the number of steps used for the trajectory is increased. We use 5 timesteps

TABLE I

ACCURACY BY VERB ACROSS UNSEEN OBJECT CATEGORIES (TRAINED BY k-FOLD CROSS VALIDATION)

Object Types	**Verb Accuracy (%)**					
	Translate Verbs	*Open/Close*	*Remove/Insert Part*	*Remove Whole*	*Rotate Verbs*	*None*
Box	96.2 ± 0.7	78.3 ± 7.4	75.8 ± 7.3	99.4 ± 1.5	62.3 ± 5.2	85.3 ± 2.5
Dishwasher	97.6 ± 1.5	82.8 ± 5.8	98.2 ± 2.6	99.0 ± 1.1	89.4 ± 1.8	91.3 ± 1.3
Door	83.2 ± 3.6	37.0 ± 7.8	75.2 ± 8.6	93.1 ± 5.2	47.0 ± 9.0	74.2 ± 6.4
Laptop	86.8 ± 4.4	26.4 ± 11.8	67.0 ± 14.8	92.4 ± 2.1	27.1 ± 4.8	84.0 ± 8.7
Microwave	93.1 ± 2.3	96.4 ± 3.7	98.2 ± 3.0	99.0 ± 2.6	80.7 ± 5.5	79.4 ± 3.0
Oven	96.4 ± 2.4	98.0 ± 0.6	99.0 ± 1.3	100.0 ± 0.0	86.0 ± 2.3	96.4 ± 1.4
Refrigerator	89.6 ± 3.0	72.6 ± 8.8	97.1 ± 1.2	96.6 ± 3.7	84.1 ± 3.2	82.6 ± 4.4
Safe	98.8 ± 1.1	89.0 ± 16.8	96.4 ± 1.1	99.4 ± 1.4	84.6 ± 3.2	93.6 ± 2.9
Stapler	54.6 ± 6.1	72.6 ± 8.9	63.1 ± 14.1	88.4 ± 7.6	16.1 ± 5.1	47.6 ± 11.9
StorageFurniture	94.1 ± 3.3	61.6 ± 5.9	89.5 ± 6.0	98.9 ± 0.4	79.3 ± 4.5	79.9 ± 10.5
Toilet	50.0 ± 14.7	42.4 ± 15.6	57.8 ± 11.3	60.9 ± 27.5	32.2 ± 7.5	55.2 ± 12.9
TrashCan	93.7 ± 5.5	73.5 ± 11.0	81.7 ± 9.8	97.1 ± 2.0	80.9 ± 2.8	90.9 ± 2.8
WashingMachine	87.8 ± 2.8	91.1 ± 4.2	89.2 ± 5.9	94.1 ± 5.3	84.6 ± 4.1	63.2 ± 19.4

as a default for the remainder of our experiments, as we empirically observed that using 5 timesteps provides good accuracy while keeping overall training time relatively low.

We perform k-fold cross validation, where there will be $k-1$ object categories used for training the classifier, and an unseen k-th object category that will be reserved for testing. The number of epochs is kept constant at 40, and the number of steps is kept at 5 out of the 20 total steps present for each trajectory in the dataset (initiation, Step 5, Step 10, Step 15, and termination snapshots). Overall, the average is 76.69% accuracy in identifying 14 verbs across 13 object categories. In Figure 3, we see the object categories that do the poorest overall are Stapler and Toilet, likely due to their larger differences in shape and usage in comparison to the other object categories. Table I shows the accuracy by verb for each object category. In the table, translate verbs include *Lower, Raise, Push, Pull, TranslateLeft,* and *TranslateRight*, while rotate verbs include *Roll, Turn,* and *Flip*. *RemoveWhole* does consistently does well for classification, likely due to a mixture of being a large difference between the initiation and termination states and that the termination state, an object-free environment, appears the same for each object category. In these experiments, the *None* verb trajectories are included in training and testing.

We conducted another experiment that compares the performance of the classifier when the test object category is similar to the object categories that are used for training, in comparison to a test category that is different. During training, the number of epochs is set to 40 and the number of timesteps is set to 5 (the initiation, Step 5, Step 10, Step 15, and the termination snapshots). *None* verb trajectories are included in training and testing. As seen in Figure 4, the average accuracy with the k-fold cross validation across the object categories Dishwasher, Door, Microwave, Refrigerator, Safe, and Washing Machine, when the k-th category is one of those categories, is higher then the k-th test category being one of the categories Box, Laptop, Stapler, Toilet, and Trash Can. We believe this is due to the objects sharing similar features of being rectangular and have doors that open in the same orientation.

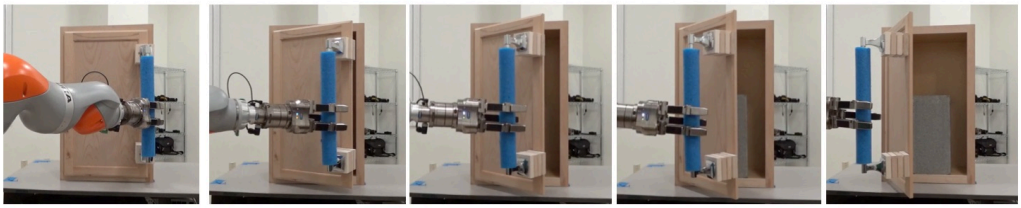
(a) Trajectory Optimizer Result for *Open* Applied to a Novel Cabinet.

(b) Trajectory Optimizer Result for *TranslateRight* Applied to a Novel Box.

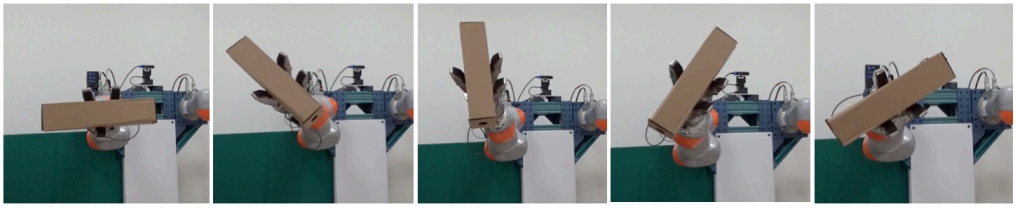
Fig. 5. Examples of trajectory optimizer results. (a) The trajectory optimizer correctly decided to manipulate the joint limit parameter by 0.40 radians for each timestep, thus producing a correct *Open*. (b) The trajectory optimizer correctly decided to manipulate the y parameter by -0.10 for each timestep, thus producing a correct *TranslateRight*.

B. Object Trajectory Optimization

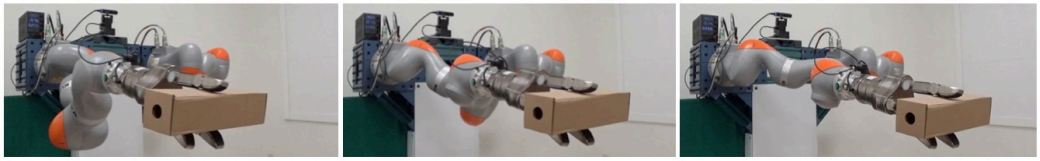
With the trained models from our classifier, we run the CMA-ES optimizer on object instances and qualitatively assess the performance. We set the initial covariance parameter to 0.33, the population size to 40, and the number of generations to 60. Since we trained our CNN classifier with five timesteps sampled at regular intervals (such that the change is equal between chosen timesteps), the optimizer predicts a single change in state (of size $6 + n$) that is applied at each timestep to generate the predicted trajectory. Further, we identified that the verbs studied include change in at most one degree of freedom; therefore, to prevent extraneous motion, we take only the maximum predicted per-timestep change in state and set the other dimensions to zero when scoring trajectories. This property is straightforward to compute from the data but could be learned for each verb.



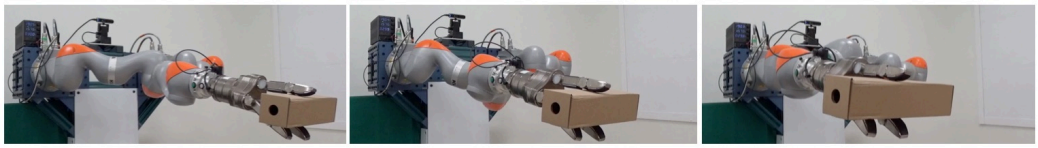
(a) *Open* Applied to a Novel Cabinet



(b) *Turn* Applied to a Novel Box



(c) *TranslateRight* Applied to a Novel Box (only 3 out of 5 generated timesteps shown)



(d) *TranslateLeft* Applied to a Novel Box (only 3 out of 5 generated timesteps shown)

Fig. 6. Snapshots of robot demonstrations using our model to generate trajectories for motion planning. With our model, the robot is able to manipulate novel object instances (viz., a Box and a Cabinet) for a variety of verbs (*Open*, *Turn*, *TranslateLeft*, and *TranslateRight*).

The resulting trajectory is rendered and scored as described in Section III-B. Qualitative results are shown in Figure 5.

C. Robot Demonstration

Finally, we demonstrate that our model enables a real robotic system to execute verb commands on novel object categories. The robot is a KUKA LBR iiwa7 with a Schunk Dexterous Hand 3-fingered gripper. Five commands were executed: *Open* (applied to a novel instance of the StorageFurniture object category) as well as *Turn*, *TranslateRight*, *TranslateLeft*, and *Push* (applied to a novel instance of the Box object category). After performing the offline trajectory optimization, the robot executes the desired object trajectories using motion planning. Plans for the *Turn*, *Push*, and *Translate* verbs were computed as simple motions in Cartesian space with the object placed in the robot's gripper. The *Open* command was executed by providing the robot with a grasp on the cabinet door (e.g., as though from an off-the-shelf grasp detection algorithm [28]), and computing a trajectory that moves the end-effector to execute the object trajectory produced by our system with MoveIt! [29].

We highlight some of the simulated trajectories that were generated by our method as Figure 5. We also show images of the KUKA robot executing some of the generated trajectories in Figure 6. We provide demonstration videos in our supplementary materials.

V. Discussion

In our experiments, we used a small set of manipulation verbs that can be realized either through object trajectory or the difference between initiation and termination states. However, our model can easily incorporate verbs realized through both robot arm trajectory and object arm trajectory by generating RGB trajectory sequences that include both the robot arm and the object (e.g., the verbs *Throw* and *Toss*).

Our implementation allows for flexibility in the length of the object trajectories and the frequency at which they are rendered. For most of our experiments, we used only five out of the total 21 generated timesteps (initiation step, Step 5, Step 10, Step 15, termination step), and we assume that there is a constant amount of time for all the object instances used for training our networks. One could prefer

to generate trajectories of length two to only produce the goal/termination state of the verb, but this was observed to decrease classifier performance empirically. One could also adjust the model to consider more timesteps, or pick irregular timesteps (not separated by a constant amount) for each verb. In other words, the number of timesteps can be thought of as way-points for the generated trajectory. Furthermore, we do not consider adverbs in our model; however, there is the possibility that certain adverbs can be incorporated into the model, such as adverbs that describe the speed of an action (e.g., slowly, quickly, carefully), by generating object trajectories where the goal state of a verb is reached at a comparatively earlier or later timestep.

Due to the flexibility of our model, it is possible that the correct verb goal state is produced even when given a non-canonical initial state of the object, e.g., a door can be opened from a initial state of being slightly ajar rather than completely closed. This could be achieved by running the trajectory optimizer step given a non-canonical state on an object, or training the classifier on more trajectories that incorporate non-canonical initial states.

The training trajectories contain ideal scenarios such as having objects in isolation and single fixed-view angles of each trajectory. To improve our model's performance, more trajectory images taken from different angles may be added to the training data. Furthermore, other state-of-the-art activity and object recognition vision classification methods can be used on trajectory videos or image sequences such as RAANet [30], YOLO [31], and other methods as described in the survey by Zaidi et al. [32].

VI. Conclusion

We have proposed a two-part model consisting of a classifier and an optimizer to generalize manipulation skills to novel object categories using verbs. We present a classifier that can recognize which verb is being performed in a given trajectory, and enables verb generalization to new object instances and new object categories. This classifier achieves an average of 76.69% accuracy over 13 object categories and 14 verbs. The optimizer is responsible for finding kinematic trajectories of an object that scores highly on the classifier for the desired verb command. Our model can generalize skills across novel objects, and we conducted robot demonstrations to show that robots can use our model with motion planning for execution on novel objects.

VII. Acknowledgements

Special thanks to our mentors and advisors for their support and advising throughout this project. Part of this research was conducted using computational resources and services at the Center for Computation and Visualization, Brown University. This work is supported by NSF under grant numbers IIS-1955361, IIS-1956221, and IIS-1941808; ONR under grant numbers N00014-21-1-2584 and N00014-22-1-2592; AFOSR DURIP FA9550-21-1-0308, and Echo Labs. Disclosure: George Konidaris is the Chief Roboticist of Realtime Robotics, a robotics company commercializing real-time motion planning.

References

[1] D. Hewlett, T. J. Walsh, and P. Cohen, "Teaching Robots to Execute Verb Phrases," in *Workshop on The State of Imitation Learning: Understanding its Applications and Promoting its Adoption at Robotics: Science and Systems*, 2011.

[2] D. Paulius, N. Eales, and Y. Sun, "A Motion Taxonomy for Manipulation Embedding," in *Robotics: Science and Systems*, 2020.

[3] B. Eisner*, H. Zhang*, and D. Held, "FlowBot3D: Learning 3D Articulation Flow to Manipulate Articulated Objects," in *Robotics: Science and Systems*, 2022.

[4] B. Ichter, A. Brohan, Y. Chebotar, C. Finn, K. Hausman, A. Herzog, D. Ho, J. Ibarz, A. Irpan, E. Jang, R. Julian, D. Kalashnikov, S. Levine, Y. Lu, C. Parada, K. Rao, P. Sermanet, A. T. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, M. Yan, N. Brown, M. Ahn, O. Cortes, N. Sievers, C. Tan, S. Xu, D. Reyes, J. Rettinghouse, J. Quiambao, P. Pastor, L. Luu, K.-H. Lee, Y. Kuang, S. Jesmonth, N. J. Joshi, K. Jeffrey, R. J. Ruano, J. Hsu, K. Gopalakrishnan, B. David, A. Zeng, and C. K. Fu, "Do As I Can, Not As I Say: Grounding Language in Robotic Affordances," in *Proceedings of The 6th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, K. Liu, D. Kulic, and J. Ichnowski, Eds., vol. 205. PMLR, 14–18 Dec 2023, pp. 287–318.

[5] P. Sharma, A. Torralba, and J. Andreas, "Skill Induction and Planning with Latent Language," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 1713–1726.

[6] A. Eisermann, J. H. Lee, C. Weber, and S. Wermter, "Generalization in multimodal language learning from simulation," in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–8.

[7] T. Kollar, S. Tellex, D. Roy, and N. Roy, "Grounding Verbs of Motion in Natural Language Commands to Robots," in *Experimental Robotics*. Springer, 2014, pp. 31–47.

[8] Y. Cui, S. Karamcheti, R. Palleti, N. Shivakumar, P. Liang, and D. Sadigh, "No, to the Right – Online Language Corrections for Robotic Manipulation via Shared Autonomy," in *Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*, 2023, pp. 93–101.

[9] K. Sugiura and N. Iwahashi, "Learning object-manipulation verbs for human-robot communication," in *Proceedings of the 2007 workshop on Multimodal interfaces in semantic interaction*, 2007, pp. 32–38.

[10] K. K. Schuler, "VerbNet: A broad-coverage, comprehensive verb lexicon," Ph.D. dissertation, 2005.

[11] F. Xiang, Y. Qin, K. Mo, Y. Xia, H. Zhu, F. Liu, M. Liu, H. Jiang, Y. Yuan, H. Wang *et al.*, "SAPIEN:

A simulated part-based interactive environment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 097–11 107.

[12] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. The MIT Press, 2018.

[13] R. Patel, R. Rodriguez-Sanchez, and G. Konidaris, "On the Relationship Between Structure in Natural Language and Models of Sequential Decision Processes," in *Language in Reinforcement Learning Workshop at International Conference on Machine Learning*, 2020.

[14] R. S. Sutton, D. Precup, and S. Singh, "Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning," *Artificial Intelligence*, vol. 112, no. 1-2, pp. 181–211, 1999.

[15] M. R. Hovav and B. Levin, "Reflections on Manner/Result Complementarity," *Lexical Semantics, Syntax, and Event Structure*, pp. 21–38, 2010.

[16] Q. Gao, M. Doering, S. Yang, and J. Chai, "Physical Causality of Action Verbs in Grounded Language Understanding," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 1814–1824.

[17] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical Text-Conditional Image Generation with CLIP Latents," *arXiv preprint arXiv:2204.06125*, 2022.

[18] C. Lynch and P. Sermanet, "Language Conditioned Imitation Learning over Unstructured Data," *Robotics: Science and Systems*, 2021.

[19] B. A. Spiegel and G. Konidaris, "Guided Policy Search for Parameterized Skills using Adverbs," *arXiv preprint arXiv:2110.15799*, 2021.

[20] A. Bucker, L. Figueredo, S. Haddadin, A. Kapoor, S. Ma, and R. Bonatti, "Reshaping Robot Trajectories Using Natural Language Commands: A Study of Multi-Modal Data Alignment Using Transformers," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022, pp. 978–984.

[21] B. Abbatematteo, S. Tellex, and G. Konidaris, "Learning to Generalize Kinematic Models to Novel Objects," in *Proceedings of the Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, L. P. Kaelbling, D. Kragic, and K. Sugiura, Eds., vol. 100. PMLR, 30 Oct–01 Nov 2020, pp. 1289–1299.

[22] 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 *Proceedings of the 5th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, A. Faust, D. Hsu, and G. Neumann, Eds., vol. 164. PMLR, 08–11 Nov 2022, pp. 991–1002.

[23] N. Hansen, "The CMA Evolution Strategy: A Tutorial," *arXiv preprint arXiv:1604.00772*, 2016.

[24] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Mottaghi, L. Zettlemoyer, and D. Fox, "ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks," in *The IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[25] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi, "AI2-THOR: An Interactive 3D Environment for Visual AI," *arXiv preprint arXiv:1712.05474*, 2017.

[26] D. Ebert and E. Pavlick, "A Visuospatial Dataset for Naturalistic Verb Learning," *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pp. 143–153, 2020.

[27] S. James, Z. Ma, D. Rovick Arrojo, and A. J. Davison, "RLBench: The Robot Learning Benchmark & Learning Environment," *IEEE Robotics and Automation Letters*, 2020.

[28] A. Ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1455–1473, 2017.

[29] S. Chitta, I. Sucan, and S. Cousins, "Moveit![ros topics]," *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 18–19, 2012.

[30] Y. Lu, X. Hao, Y. Li, W. Chai, S. Sun, and S. Velipasalar, "Range-aware attention network for lidar-based 3d object detection with auxiliary point density level estimation," *arXiv preprint arXiv:2111.09515*, 2022.

[31] S. Shinde, A. Kothari, and V. Gupta, "YOLO based human action recognition and localization," *Procedia computer science*, vol. 133, pp. 831–838, 2018.

[32] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar, and B. Lee, "A survey of modern deep learning based object detection models," *Digital Signal Processing*, vol. 126, p. 103514, 2022.