One-shot Visual Imitation via Attributed Waypoints and Demonstration Augmentation

Anonymous Author(s)
Affiliation
Address
email

Abstract

In this paper, we analyze the behavior of existing techniques and design new solutions for the problem of one-shot visual imitation. In this setting, an agent must solve a novel instance of a novel task given just a single visual demonstration. Our analysis reveals that current methods fall short because of three errors: the DAgger problem arising from purely offline training, last centimeter errors in interacting with objects, and mis-fitting to the task context rather than to the actual task. This motivates the design of our modular approach where we a) separate out task inference (what to do) from task execution (how to do it), and b) develop data augmentation and generation techniques to mitigate mis-fitting. The former allows us to leverage hand-crafted motor primitives for task execution which side-steps the DAgger problem and last centimeter errors, while the latter gets the model to focus on the task rather than the task context. Our model gets 100% and 48% success rates on two recent benchmarks, improving upon the current state-of-the-art by absolute 90% and 20% respectively.

1 Introduction

Consider a single video, demonstrating the task depicted in Figure 1 (left). Given just this input, as humans we can reliably execute the demonstrated task in the novel situation shown in Figure 1 (right). This is in spite of the differences in the task instance (location of relevant objects are different from where they were in the demonstration), embodiment (*e.g.* robot hand in demonstration *vs.* human hand), and the large ambiguity in what precisely the task was (was it to move the hand through those locations or was it to move the object). In this paper, we seek to imbue robotic agents with a similar capability: given a *single visual demonstration* of a *novel task*, the robot should execute the demonstrated task on a *novel instance* of the task. We refer to this problem as *one-shot visual imitation*.

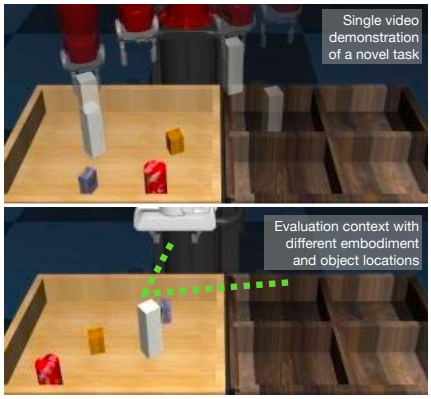

Figure 1: One-shot Visual Imitation

While humans are adept at this form of one-shot visual imitation, machine performance in this setting lacks considerably. For instance, the recent method from Dasari *et al.* [11] obtains a 10% success rate on a harder version of their pick-and-place task-set, and 28% on a one shot visual imitation benchmark constructed using Meta-world [46]. In this paper, we investigate what causes recent methods to underperform and develop algorithms to bridge this performance gap.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

We start by analyzing the behavior of current methods. Current works on this problem [11, 15, 29] cast it as a conditional policy learning problem (*i.e.* predict the *next action* conditioned on the demonstration and the execution so far) using meta-learning [15] or expressive neural network models [11, 29]. Models are trained on *offline* datasets of video demonstrations paired with expert executions. This immediately reveals two issues that hinder the performance of these past works. Purely offline and non-interactive training causes the learned policies to suffer from a form of distribution shift known as the *DAgger problem*, (going off-distribution due to compounding errors while imitating long-horizon behaviors) and near misses while executing fine motor control, or *last centimeter errors* (prior work has shown that learning generalizable polices for fine-motor control requires specialized architecture or thousands of online samples).

When we try to extend current methods to more diverse collection of tasks, a third, more subtle *mis-fitting* issue comes to light. As tasks are often contextual (*i.e.* one only interacts with a given object in limited number of ways), current models tend to make predictions based on the objects in the scene rather than the motion depicted in the demonstration. This causes them to generalize poorly to novel tasks.

These insights motivate the design of our method. To circumvent the first two problems, we employ a hierarchical and modular approach that separates out the task execution (*how* to do it) from task inference (*what* to do). This separation enables us to use robust and high-performing, hand-crafted motor primitives for task execution, while the use of learning for task inference allows the system to interpret the intent depicted in the provided demonstration, and synthesize a solution for the novel instance at test-time. More concretely, given the video demonstration and just a single image of the current scene, our learned model predicts a sequence of *attributed waypoints* that outline a trajectory to achieve the task. These *attributed waypoints* represent the 3D motion of the arm, along with additional attributes of the robot's state (such as "an object is in the gripper") at those waypoints. The predicted attributed waypoints are achieved using motor primitives (based on kinematic planning or classical grasping primitives using depth images from hand-in-eye cameras).

While this seemingly simple model works well for pick-and-place tasks (achieving 100% success rate on the task-set from Dasari *et al.* [11]), it still underperforms on the diverse tasks in Meta-world [46], due to the mis-fitting issue described above. To mitigate this, we propose novel demonstration augmentation schemes that generate training samples to break the correlation between tasks and their contexts.

We evaluate our proposed method on 4 benchmarks, representing a wide range of diverse tasks in simulation, and evaluation on real-world data [20]. In comparisons against 3 past methods (DAML [45], T-OSVI [11], and MOSAIC [29]) our model achieves strong results, surpassing all baselines. Notably, our method makes large improvements on two benchmarks, reaching success rates of 100% and 48%. These represent absolute improvements of 90% and 20% respectively, over the current state-of-the-art.

2 Related Work

Learning from demonstrations in robotics has taken many different forms over the years. One such setting is behavior cloning (BC), in which one is given many paired sensor and action trajectories for a single task, and the goal is to obtain a policy for this task [37]. Purely training on offline datasets of expert behavior is known to suffer from compounding errors at execution time, motivating improvements like DAgger [38]. Recent approaches to this problem have considered BC using only one trajectory with action labels [12, 13], and using meta-learning to adapt to novel tasks at test time [15].

One-shot visual imitation takes this problem a step further, where only a video (*i.e.* with no associated actions) of the expert's execution is available [15, 45, 34]. Researchers have explored many different variants, with demonstrations differing in embodiment [13, 45], viewpoint [41], or using natural language [20, 1]. Researchers have also pursued many different solutions: conditioning on task embeddings [19, 5], using meta-learning [13, 45], predicting sub-goals [34, 41], using expressive transformer architectures [11, 29], and contrastive training of visual features [29]. Huang *et al.* [18] and Sharma *et al.* [41] follow a hierarchical design similar to ours for one-shot visual imitation. However, our formulation can deal with tasks involving arbitrary objects and motions, unlike the method from [18] which only operates within a pre-defined set of discrete symbols and

motions. [41] synthesizes images to use as sub-goals, while our use of generalized waypoints sidesteps the need for image generation, which can be challenging for novel objects in complex environments.

We distinguish from another related line of work on inverse reinforcement learning (IRL) [42, 48, 21, 44, 17, 2]. IRL assumes interactive access to the underlying environment to learn policies, which our setting does not.

Hierarchical policies have been found to useful in many settings, indoor navigation [3, 6, 30], self-driving [31], drone control [24, 23], and manipulation [32, 14, 8, 10, 22, 36] among many others. In reinforcement learning settings, motion primitives have been incorporated as additional actions to speed up learning [33, 10, 8]. Instead, our work develops techniques to use motor primitives for one-shot visual imitation. While we found hand-designed primitives effective in our work, our method is agnostic to the form of motor primitives, and could benefit from the many recent works on discovering motor primitives from diverse trajectories [26, 40, 39, 25, 16].

Data augmentation techniques have been found to be effective in improving the generalization of learned models [7]. They have also been effective at improving generalization in robot learning, *e.g.* when learning policies via RL [43], or for pre-training representations [27, 29]. Our work also employs data augmentation techniques, inspired from mixup [49, 47, 4] for improving generalization. However, the specific form of overfitting in our one-shot visual imitation problem motivates the need for *asymmetrical* mixing of samples to decorrelate tasks from tasks contexts.

3 Diagnosing Errors Made by Current One-shot Visual Imitation Methods

The problem we are interested in is that of one-shot visual imitation. At test time, our agent will be given a video demonstration of a task not seen during training, and must perform the depicted task with no additional experience in the environment. Note that, the environment configuration may be different from that depicted in the example video, but the overall semantic task will be the same.

Samples in one-shot visual imitation learning datasets consist of: a) the video demonstration $\mathbf{v}$; and b) a robotic trajectory, $\{(o_1, s_1, a_1), \ldots\}$ that conducts the same task in a potentially different situation. These video demonstrations are not the same trajectory as the robotic trajectory and may differ in embodiment, or solution method, but they must be solving the same high-level task. In fact, the pairing of video demonstrations to robotic trajectories is what defines the notion of a task for the model being trained.

This is commonly cast as a supervised learning problem [11, 29, 15, 45], where a model is learned on demonstration-trajectory pairs, to predict the action at a given timestep, conditioned on the demonstration, and previous frames, $\pi(a_t|\mathbf{v}, o_{1:t}, s_{1:t})$. However, this approach can lead to undesirable behaviors on novel tasks. In this section, we characterize the failure modes of T-OSVI, the transformer-based per-time-step action prediction model from Dasari *et al.* [11] as a representative recent method. Specifically, we highlight 3 different failure modes that arise in this standard approach: the DAgger problem arising from purely offline training, last centimeter errors in interacting with objects, and mis-fitting to the task context rather than the depicted task.

Experimental Setting. We consider a harder version of the 4 object and 4 bin pick-and-place task family proposed in [11] (visualized in Figure 1), that has been modified to hold out tasks as opposed to task instances as originally done in [11]. That is, of the 16 possible tasks (picking one of the four objects and placing it in one of the four bins), we use 14 for training and hold out 2 for testing. In this modified setting, the success rate for T-OSVI [11] drops to 10% from the 88% reported in their paper. In this setting, we identify two consistent failure modes: a) failure to reliably reach the target object (about 88% trials) and often times (35% of trials) reaching a non-target object due to what we believe to be a version of the DAgger problem, and b) near misses in grasping the object (about 10% of all trials in which it reached any object).

DAgger problem. T-OSVI can be viewed as a conditional policy of the form $\pi(a_t|\mathbf{v}, o_{1:t}, s_{1:t})$, trained through behavior cloning on an offline dataset of expert executions. Behavior cloning on expert data is known to suffer from poor execution performance due to compounding errors [38]. While this may explain the low reaching success rate for the target object, it doesn't explain the relatively high rate with which the policy reaches and attempts to grasp a non-target object.

Our belief is that this is a task execution error. The policy is trained with memory of its execution over the last 6 frames, and often it relies more on these recent execution frames, than the demonstration.

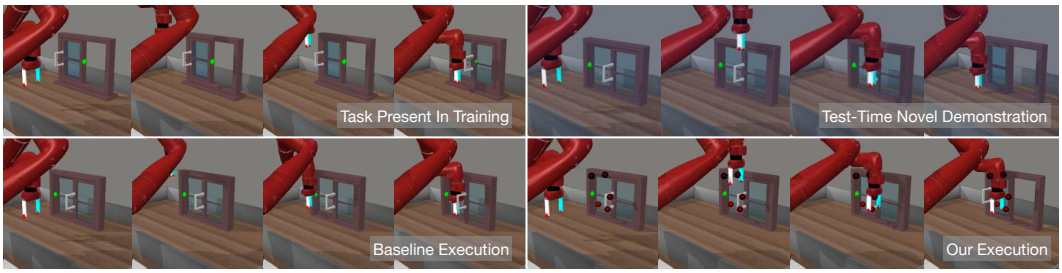

Figure 2: **Example of mis-fitting in Meta-world (Section 3)**: During training there is a task depicting the window being closed (top left). When presented with a novel task demonstration, opening the window (top right), action-prediction methods repeat the motion on the most similar training setting, trying to close the already closed window (bottom left). Our method successfully opens the window (bottom right). Predicted waypoints (red dots) correctly move to the right side of the handle and push left.

Consequently, we find that if the agent makes small errors early during execution, subsequent behavior is more in line with the current execution, at the cost of being inconsistent with the given demonstration.

We empirically verify this by keeping the demonstration fixed but *guiding* the policy at test time towards the target object (positive guidance) or towards a distractor non-target object (negative guidance), by taking steps with an oracle policy. Even 1 step of guidance drastically improves the reaching rate from 12% to 58% $\pm$ 2% for the target object, and from 27% to 50% $\pm$ 2% for the distractor object. Note that it takes on average 12 steps to reach the object, so 1 step of guidance is not much. The increase in success rate for *both* target and distractor objects reveals the preference of the policy towards past execution frames over the demonstration.

Last centimeter errors in grasping. Next, we discuss the second substantial error mode of T-OSVI on this task. While 12% of executions reach the correct object, only 10.5% of executions successfully lift the object, meaning 10% of attempted grasps fail. This is because the gripper grasps near the object, but misses, or acquires an unstable grasp. This is not surprising as we are attempting to learn a grasping policy from as few as 1400 training samples. Past works have shown that without specialized architectures or sensing, many thousands of trials are necessary to learn grasping policies that generalize [28, 35]. Other recent works [20, 29] also noted these fine-grained errors in one-shot visual imitation.

Mis-fitting to Task Context. In the harder, more diverse set of 50 tasks from Meta-world [46], a new failure mode of one-shot visual imitation methods arises. We find that, if the novel evaluation task involves objects that are visually similar to those seen in training tasks, models trained with T-OSVI perform the motion from the training task, not what is seen in the demonstration (visualized in Figure 2). We believe that this is because the model is predicting actions based on the task context (objects visible in the scene) as opposed to the task depicted in the demonstration.

Tasks are contextual, *i.e.* there are really only a few different things that one can do with a given object (*e.g.* opening a closed door). Besides, collecting training data for one-shot visual imitation is challenging as it requires paired demonstrations and trajectories. Thus, training datasets are small and don't showcase diverse interactions with the manipulated objects. Models can easily satisfy the training objective by fitting to the task context and ignoring the motion depicted in the demonstration. This causes problems when we seek to imitate a novel task which bears visual similarity to a training task. As depicted in Figure 2, existing methods will attempt to perform the task seen in training instead of that depicted in the demonstration.

The analysis of these three failure modes motivates the design of our method, as detailed in the next section. We address task execution errors (*i.e.* the DAgger problem and the last centimeter problem) through the use of hand-crafted motor primitives, and present data augmentation strategies that break the correlation between tasks and task contexts, mitigating mis-fitting.

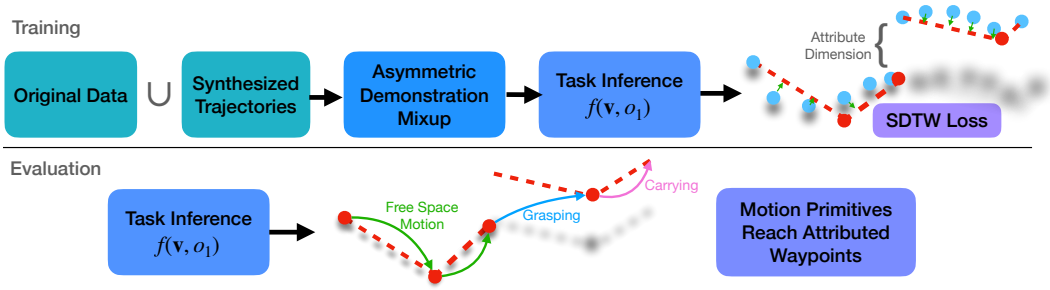

Figure 3: **Bottom:** AWDA is a modular approach for one-shot visual imitation that separates task inference and task execution. Task inference function $f(\mathbf{v}, o_1)$ predicts a sequence of *attributed* waypoints (red points) that are achieved using hand-defined motion primitives (colored solid lines). **Top Right:** $f(\mathbf{v}, o_1)$ learns to predict attributed waypoints by aligning them with ground-truth attributed trajectories using SDTW (Section 4.1). **Top Left:** To prevent overfitting of $f(\mathbf{v}, o_1)$ to task contexts, we synthesize additional demonstrations and employ asymmetric demonstration miuxup (Section 4.2).

4 Visual Imitation via Attributed Waypoints and Demonstration Augmentation

Our approach, AWDA, is a hierarchical and modular approach that separates out task inference and task execution. The task inference module takes the given demonstration video and a single image of the scene (depicting an instance of the task, different from that in the demonstration) and outputs the full execution plan, expressed as a sequence of *attributed waypoints*. Task execution happens simply by invoking the appropriate motor primitive to convey the robot end effector between each consecutive pair of predicted waypoints.

4.1 Task Inference and Execution via Attributed Waypoints and Motor Primitives

Attributed Waypoints. Our task inference and execution modules are interfaced via attributed waypoints. Typical waypoints used in robotics (*e.g.* for navigation [3]) only capture the 3D (or 6D) pose of the robot end-effector. This is quite restrictive for manipulation tasks, as purely kinematic guidance of the end-effector will not be able to interact with objects *e.g.* to pick them up or to exert forces on them. We overcome this limitation by assigning additional *attributes* to each waypoint, *e.g.* is an object in the end-effector, or is the end-effector experiencing force in a particular direction. We consider attributed waypoints to be $3 + k$ dimensional, where k is the number of additional attributes associated with each 3D waypoint. Attributed waypoints are are powerful tool for expressing solutions to kinematic tasks. For instance, just 1 single attribute, of whether there is an object in the gripper or not, allows us to express all 50 tasks in the Meta-world task-set as a sequence of these 4D waypoints (3D for end-effector location and 1D for *"is there an object in the gripper"*). We will use this attribute as a running example for explanation, but other attributes could be added.

Motor Primitives. Given a pair of attributed waypoints, our method uses motor primitives to convey the robot between pairs of attributed waypoints. While conveying the end-effector between 3D waypoints in space is well understood (inverse kinematics and motion planning), moving between our proposed attributed waypoints is more involved as it can involve a change in "attributes" along the way. Thankfully, changes in attributes correspond to well-studied basic skills in robotics literature. For instance, using the same 4D grasping example as above, going from waypoint $[\mathbf{p}; \text{false}]$ to $[\mathbf{q}; \text{true}]$ involves grasping an object near location $\mathbf{p}$ and taking it to location $\mathbf{q}$; while going from $[\mathbf{q}; \text{true}]$ to $[\mathbf{p}; \text{false}]$ corresponds to releasing the currently held object and then going to location $\mathbf{p}$. In general, for k attributes, this corresponds to 2^{k+1} motor primitives. Our method is agnostic to the exact implementation of motor primitives. For our experiments, we found that hand-crafted primitives were sufficient to solve the pick-and-place task-set from [11] and all 50 Meta-world tasks [46]. We implemented 4 hand-crafted primitives: a) free space motion without any object in hand, b) grasping an object, c) dropping an object, and d) free space motion with an object in hand; using eye-in-hand depth cameras.

Training the Model to Output Augmented Waypoints. Augmented waypoints and corresponding motor primitives let us express manipulation tasks as a sequence of waypoints. We next describe

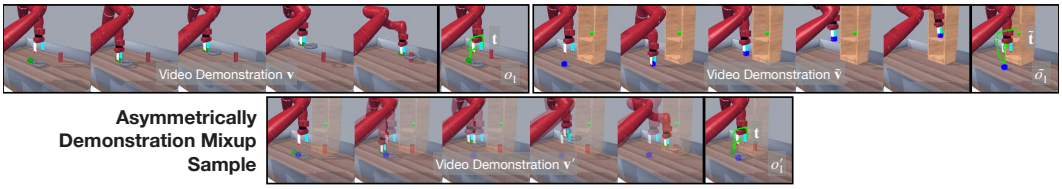

Figure 4: **Asummetric Demonstraion Mixup (ADM) augmentation:** Two demonstration-trajectory pairs (row 1) are combined into one (row 2). The first frame of demonstration 2 is blended into every frame of demonstration 1. The end effector trajectory that serves as supervision for training on the combined sample is entirely from trajectory 1.

how we train the task inference module to predict these augmented waypoints from a given demonstration, and a single image of the novel task instance. Our task inference model f takes as input, a demonstration $\mathbf{v}$ and instance image o_1, and outputs n attributed waypoints $\{\mathbf{w}_1, \ldots, \mathbf{w}_n\}$, each waypoint being $k+3$ dimensional. Supervision for these waypoint predictions is derived from the trajectories in the dataset as follows. We process the given robotic trajectory, $\{(o_1, s_1, a_1), \ldots\}$, into 3D end-effector locations using forward kinematics. We also assign to each time step, the appropriate attributes that the agent experiences in that frame. These attributes are not labeled by hand, but rather mined automatically from the robot state s_t and a_t provided in the robotic trajectories. For example, we label that an object has been grasped in a robotic trajectory when the commanded action is to close the gripper, but the gripper jaws do not close. This gives us end-effector's *attributed trajectory*: $\mathbf{t} = \{\mathbf{t}_1, \ldots, \mathbf{t}_T\}$, $t_i \in \mathbb{R}^{k+3}$. We derive supervision for waypoints $\{\mathbf{w}_1, \ldots, \mathbf{w}_n\}$ by constructing a trajectory $\hat{\mathbf{t}}$ by linearly interpolating $\mathbf{w}$'s and comparing it to the ground truth attributed trajectory $\mathbf{t}$. We use soft dynamic time-warping (SDTW) [9] to compute the loss between the predicted and ground truth trajectory. Minimizing this objective aligns the predicted trajectory with the ground truth trajectory.

Testing. Given a novel video $\mathbf{v}$, and task instance, as observed in image o_1, we use the task inference model $f(\mathbf{v}, o_1)$ to predict attributed waypoints. The appropriate motor primitives are used to convey the robot from one predicted attributed waypoint to the next, until all waypoints are exhausted.

4.2 Demonstration Augmentation for Improved Task Inference

We next look at tackling the *mis-fitting* issue highlighted in Section 3. This mis-fitting happens because of the strong correlation between the task and its context in the training data. We design two augmentation strategies that break this correlation by generating training samples with the same context but different task motion.

Asymmetric Demonstration Mixup. Our first strategy creates new training samples by mixing existing samples in the dataset. This is reminiscent of mixup [49] but has modifications to break the aforementioned correlation. Naively mixing samples as done in original mixup wouldn't break the correlation to aid out-of-distribution generalization. Instead, we leverage the temporal nature of video demonstrations to *asymmetrically* blend samples. Given a sample $(\mathbf{v}, o_1, \mathbf{t})$, we use another sample $(\tilde{\mathbf{v}}, \tilde{o}_1, \tilde{\mathbf{t}})$ to produce a new decorrelated sample by: a) blending all frames in $\mathbf{v}$ with the first frame of the video $\tilde{\mathbf{v}}$ to generate new video $\mathbf{v}'$, b) blending o_1 with $\tilde{o}_1$ to generate o_1', and c) retaining $\mathbf{t}$ as is (as depicted in Figure 4). Specifically,

$$\mathbf{v}_t' = \alpha \mathbf{v}_t + (1-\alpha)\tilde{\mathbf{v}_0}, \quad o_1' = \alpha o_1 + (1-\alpha)\tilde{o_1}, \quad \mathbf{t}' = \mathbf{t}$$

We use a blending ratio $\alpha \sim [0.3, 1.0]$, biased towards retaining all of o_1 and $\mathbf{v}$, since the trajectory is always $\mathbf{t}$. This asymmetric blending, where one of the demonstrations is frozen in time while the other is moving, breaks the correlation between objects present in the scene and the task being conducted on them. f can't just look at o_1', but it has to track how the hand moves through in $\mathbf{v}'$ to make correct predictions. Including unaltered samples in training lets us use demonstrations and observations as is at test time.

Additional Demonstrations via Trajectory Synthesis.
Additionally, we can break the correlation between tasks and task contexts by simply generating synthetic tasks involving free space motions for the robot, in various contexts. We do this by sampling a small number of points (1 to 3) uniformly at random within the agent's workspace and moving the end effector sequentially through these points using an inverse kinematics solver. Training samples are created by pairing each trajectory with itself, *i.e.* $\mathbf{v}$, o_1 and $\mathbf{t}$ all come from the same trajectory.

Table 1: Success rates on held-out tasks on the Pick-and-place and Meta-world benchmarks

	DAML [45]	**T-OSVI** [11]	**Full-1**	**Full-2**	**single head**	**no AD**	**no ADM**	**no way points**	**only waypoints**
Asymm. Demo Mixup (ADM)?			✓	✓	✓	✓	✗	✓	✗
Additional Data (AD) Source?			TS	BC-Z	TS	✗	TS	TS	✗
Waypoints?			✓	✓	✓	✓	✓	✗	✓
Pick-and-place	0.01	0.10	**1.00**	**1.00**	0.99	1.00	0.98	0.01	0.98
Meta-world [easy]	0.04	0.50	0.66	**0.73**	0.33	0.74	0.03	0.33	0.14
Meta-world [hard]	0.08	0.07	**0.30**	0.17	0.29	0.11	0.19	0.06	0.02
Meta-world [all]	0.06	0.28	**0.48**	0.45	0.31	0.42	0.11	0.19	0.08

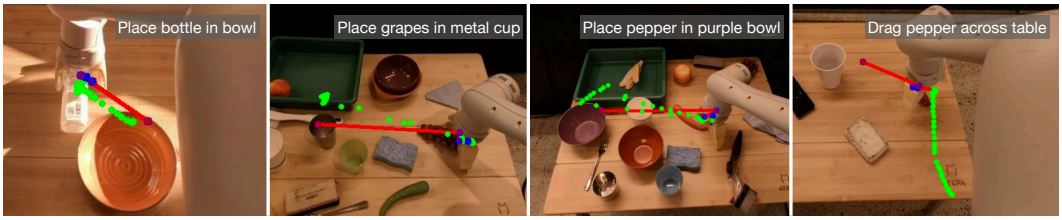

Figure 5: We visualize 2D projections of 3D waypoints in blue, and the interpolated trajectory in red, as predicted by our model for 4 different held-out tasks (noted on top right of each image) from the BC-Z dataset [20]. Predicted trajectories match the ground truth trajectories (in green). Interestingly, for the *"drag pepper across table"* task, though our prediction does not match the specific ground truth, it is still consistent with the semantics of the depicted task.

To make correct predictions on these trajectories, the model must attend to the motion of the arm and ignore background elements, thus breaking the undesired correlations. Note that this is a simple data collection procedure that can be easily done in an unsupervised manner. We simply add in these additional samples for training, and find they boost performance, particularly for substantially out-of-distribution tasks.

We note that this procedure produces samples that are not driven by a semantically-meaningful, object-centered task. Thus, including these samples in training could also impact the model's ability to learn a meaningful prior over tasks. To mitigate this, we modify final layer of the model to have two heads. One makes predictions for original samples in the dataset, while the other makes predictions for the synthesized trajectories. This nudges the overall network to look at the motion of the hand while also letting the last layer learn the necessary priors from the task driven samples in the dataset.

5 Experiments

We design and conduct experiments to demonstrate the effectiveness of our proposed method with respect to prior work, and evaluate our various design choices.

Tasks, Environments, and Datasets. We conduct experiments on 4 datasets: a) **Pick-and-place** task-set from [11] (shown in Figure 1) but modified to hold 2 of 16 possible tasks as novel testing tasks; b) **Meta-world** task-set [46] (sample observations shown in Figure 2) where we hold out 4 of 50 tasks as novel testing tasks; c) **MOSAIC** task-set [29] containing 6 tasks, evaluating performance on each task with a model trained only using demonstrations from the other tasks; and d) **BC-Z** dataset [20] that has 17213 *real-world* trajectories (sample images in Figure 5) spanning 90 tasks of which we hold out 5 for testing.

Pick-and-place and MOSAIC use different embodiments for demonstration and execution (Sawyer and Panda respectively). Meta-world and BC-Z use the same embodiment. We conduct interactive evaluation of the learned policies on Pick-and-place, Meta-world and MOSAIC task-sets and report success rate. For BC-Z, we do offline evaluation and report the accuracy of predicted trajectories on a held-out validation set. We note that as all tasks are set up in the same environment for the Pick-and-place task set, so it doesn't suffer from correlations between tasks and task contexts, or the mis-fitting error described in Section 3. However, different tasks in Meta-world and MOSAIC involve different objects making them suffer from the mis-fitting error.

Implementation Details. We follow [11] to construct data for training. We collect 100 successful trajectories with each robot using hand-defined expert controllers and arbitrarily pair them up to

Table 2: Success rates on held-out tasks from MOSAIC [29] task-set

Task	Door	Drawer	Button	Stack Block	Basketball	Nut Assembly	All
MOSAIC [29]	0.05	0.15	**0.05**	0	0	0	0.04
Full-1	**0.10**	**0.29**	0.01	0	0	**0.02**	**0.07**

construct 10K training samples per task. We train our models for 500K iterations. Following [29], we report the success rate on held-out tasks, averaged over 5 snapshots from the end of training. Our attributed waypoints use the *"is object in hand"* attribute. This leads to 4 motor primitives: free space motion without an object, moving an object, grasping a nearby object, releasing the object. We implement the first two using inverse kinematics and motion planning; the third is implemented by analyzing depth images to identify objects and centering the gripper to grasp the nearest object; for the fourth we just open the gripper.

The neural network design uses the same feature extraction process as T-OSVI [11]. We extract image features using ResNet-18, which remain spacial, and have a sinusoidal positional encoding added, before being processed by a transformer module. Finally, the temporally processed features are projected down into waypoints by two separate heads, one to predict waypoints for task-driven trajectories, and one for trajectories synthesized as described in Section 4.2.

Results. We report results on the Pick-and-place and Meta-world task sets in Table 1. We break down results on Meta-world into two splits based on the similarity of the novel task to tasks seen in training: Meta-world [easy] (*Button-Press-V2*, *Pick-Place-Wall-V2*, which differ from training tasks *Buttom-Press-Wall-V2* and *Pick-Place-V2* only due to presence/absence of distractor), and Meta-world [hard] (*Window-Open-V2*, *Door-Unlock-V2*, require categorically different solutions than training tasks). The results on the MOSAIC benchmarks are presented in Table 2. Each column reports the performance on the indicated held-out task. The models for MOSAIC experiments are trained all MOSAIC tasks except the held-out task. The column *All* reports the mean performance across all held-out task experiments. We summarize our key takeaways below.

• **AWDA outperforms prior work by a large margin.** Our full system (denoted *Full-1*) completely solves the Pick-and-place task (improving upon the 10% obtained by T-OSVI, 1% by DAML), and obtains 48% for Meta-world [all] *vs.* 28% for T-OSVI, 6% for DAML, while quadrupling the performance on the hard tasks 30% *vs.* 7% for T-OSVI. Performance gains are maintained even if we omit synthesized trajectory data altogether (denoted *no AD*), or when using data from other datasets instead (denoted *Full-2*). On the MOSAIC tasks (Table 2), we match or outperform the current state-of-the-art for this benchmark [29] on all tasks except for one, yielding superior overall performance (7% *vs.* 4%).

• **Attributed waypoints with motor primitives eliminate all errors on Pick-and-place.** Our models without asymmetric demo mixup (*no ADM*), or without additional data (*no AD*), or without both (*only waypoints*) obtain close to perfect performance on the Pick-and-place task. This demonstrates the effectiveness of our proposed modular policy architecture. It also boosts performance on Meta-world [all] by an absolute 29% from 19% with *no waypoints vs.* 48% with (*Full-1*).

• **Additional data via trajectory synthesis or from other datasets helps improve generalization**. Using additional data via trajectory synthesis (*Full-1*) or from other datasets (*Full-2*) improves upon not using any additional data (denoted *no AD*), particularly for the Meta-world [hard] tasks that require entirely novel motion at test time (11% for *no AD vs.* 30% for *Full-1* and 17% for *Full-2*). Furthermore, fitting this additional data through another head is crucial for maintaining performance on Meta-world [easy] tasks, which bear more similarity to training tasks: 66% for the two headed model *Full-1 vs.* 33% for the *single head* model.

• **Asymmetric demonstration mixup improves performance** beyond the standard image augmentations (random flip, crop, translation, color jitter, *etc.*) that are already in use for T-OSVI, DAML and all our models (*Full-1 vs. no ADM*).

• **AWDA gets good performance on real data from robots.** On the BC-z dataset [20], our method is able to predict the final interaction point (grasp, release, or reach point) to within 10 cm for $63\% \pm 4\%$ samples of *held-out* tasks. This clearly identifies what objects need to be interacted with. We expect appropriately designed motor primitives on physical robots will be able to successfully execute some of these tasks. Figure 5 shows some sample predictions.

6 Conclusion and Future Work

In this paper we analysed the major failure modes of state-of-the-art action prediction methods for one-shot visual imitation. We find that they suffer from the DAgger problem, last centimeter errors, and mis-fitting to task contexts. Our proposed method, utilizing attributed waypoints and demonstration augmentation, is able to significantly boost success rates on existing benchmarks, even completely solving one.

As is, our system is limited to kinematic tasks, but could be expanded to reasoning about forces, given the proper motion primitives. While our motion primitives are closed-loop and account for slight changes in object locations, the high level plan cannot adjust to large changes in the scene after initial waypoint predictions. We leave this to future work.

References

[1] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

[2] Shikhar Bahl, Abhinav Gupta, and Deepak Pathak. Human-to-robot imitation in the wild. *RSS*, 2022.

[3] Somil Bansal, Varun Tolani, Saurabh Gupta, Jitendra Malik, and Claire Tomlin. Combining optimal control and learning for visual navigation in novel environments. *CoRL*, 2019.

[4] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019.

[5] Alessandro Bonardi, Stephen James, and Andrew J Davison. Learning one-shot imitation from humans without humans. *IEEE Robotics and Automation Letters*, 5(2):3533–3539, 2020.

[6] Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to explore using active neural slam. *arXiv preprint arXiv:2004.05155*, 2020.

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020.

[8] Rohan Chitnis, Shubham Tulsiani, Saurabh Gupta, and Abhinav Gupta. Efficient bimanual manipulation using learned task schemas. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[9] Marco Cuturi and Mathieu Blondel. Soft-dtw: a differentiable loss function for time-series. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2017.

[10] Murtaza Dalal, Deepak Pathak, and Russ R Salakhutdinov. Accelerating robotic reinforcement learning via parameterized action primitives. *Advances in Neural Information Processing Systems*, 34, 2021.

[11] Sudeep Dasari and Abhinav Gupta. Transformers for one-shot visual imitation. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2020.

[12] Yan Duan, Marcin Andrychowicz, Bradly Stadie, OpenAI Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. One-shot imitation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[13] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1126–1135, 2017.

[14] Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2786–2793. IEEE, 2017.

[15] Chelsea Finn, Tianhe Yu, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot visual imitation learning via meta-learning. In *Conference on robot learning*. PMLR, 2017.

[16] Abhishek Gupta, Vikash Kumar, Corey Lynch, Sergey Levine, and Karol Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2019.

[17] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 29, 2016.

[18] De-An Huang, Danfei Xu, Yuke Zhu, Animesh Garg, Silvio Savarese, Li Fei-Fei, and Juan Carlos Niebles. Continuous relaxation of symbolic planner for one-shot imitation learning. In *International Conference on Intelligent Robots and Systems*, 2019.

[19] Stephen James, Michael Bloesch, and Andrew J Davison. Task-embedded control networks for few-shot imitation learning. In *Conference on robot learning*, pages 783–795. PMLR, 2018.

[20] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pages 991–1002. PMLR, 2022.

[21] Jun Jin, Laura Petrich, Masood Dehghan, and Martin Jagersand. A geometric perspective on visual imitation learning. In *International Conference on Intelligent Robots and Systems*, 2020.

[22] Leslie Pack Kaelbling and Tomás Lozano-Pérez. Hierarchical planning in the now. In *Workshops at the Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

[23] Elia Kaufmann, Mathias Gehrig, Philipp Foehn, René Ranftl, Alexey Dosovitskiy, Vladlen Koltun, and Davide Scaramuzza. Beauty and the beast: Optimal methods meet learning for drone racing. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 690–696. IEEE, 2019.

[24] Elia Kaufmann, Antonio Loquercio, Rene Ranftl, Alexey Dosovitskiy, Vladlen Koltun, and Davide Scaramuzza. Deep drone racing: Learning agile flight in dynamic environments. In *Conference on Robot Learning*, pages 133–145. PMLR, 2018.

[25] Thomas Kipf, Yujia Li, Hanjun Dai, Vinicius Zambaldi, Alvaro Sanchez-Gonzalez, Edward Grefenstette, Pushmeet Kohli, and Peter Battaglia. Compile: Compositional imitation learning and execution. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2019.

[26] Ashish Kumar, Saurabh Gupta, and Jitendra Malik. Learning navigation subroutines by watching videos. In *CoRL*, 2019.

[27] Misha Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[28] Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *IJRR*, 2018.

[29] Zhao Mandi, Fangchen Liu, Kimin Lee, and Pieter Abbeel. Towards more generalizable one-shot visual imitation learning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

[30] Xiangyun Meng, Nathan Ratliff, Yu Xiang, and Dieter Fox. Neural autonomous navigation with riemannian motion policy. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8860–8866. IEEE, 2019.

[31] Matthias Müller, Alexey Dosovitskiy, Bernard Ghanem, and Vladlen Koltun. Driving policy transfer via modularity and abstraction. *arXiv preprint arXiv:1804.09364*, 2018.

[32] Ashvin V Nair, Vitchyr Pong, Murtaza Dalal, Shikhar Bahl, Steven Lin, and Sergey Levine. Visual reinforcement learning with imagined goals. *Advances in neural information processing systems*, 31, 2018.

[33] Soroush Nasiriany, Huihan Liu, and Yuke Zhu. Augmenting reinforcement learning with behavior primitives for diverse manipulation tasks. *arXiv preprint arXiv:2110.03655*, 2021.

[34] Deepak Pathak, Parsa Mahmoudieh, Guanghao Luo, Pulkit Agrawal, Dian Chen, Yide Shentu, Evan Shelhamer, Jitendra Malik, Alexei A Efros, and Trevor Darrell. Zero-shot visual imitation. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 2050–2053, 2018.

[35] Lerrel Pinto and Abhinav Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. *ICRA*, 2016.

[36] Sören Pirk, Karol Hausman, Alexander Toshev, and Mohi Khansari. Modeling long-horizon tasks as sequential interaction landscapes. *arXiv preprint arXiv:2006.04843*, 2020.

[37] Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1988.

[38] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *AISTATS*, pages 627–635, 2011.

[39] Tanmay Shankar and Abhinav Gupta. Learning robot skills with temporal variational inference. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020.

[40] Tanmay Shankar, Shubham Tulsiani, Lerrel Pinto, and Abhinav Gupta. Discovering motor programs by recomposing demonstrations. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[41] Pratyusha Sharma, Deepak Pathak, and Abhinav Gupta. Third-person visual imitation learning via decoupled hierarchical controller. *Advances in Neural Information Processing Systems*, 32, 2019.

[42] Laura Smith, Nikita Dhawan, Marvin Zhang, Pieter Abbeel, and Sergey Levine. Avid: Learning multi-stage tasks via pixel-level translation of human videos. *arXiv preprint arXiv:1912.04443*, 2019.

[43] Aravind Srinivas, Michael Laskin, and Pieter Abbeel. CURL: Contrastive unsupervised representations for reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020.

[44] Haoyu Xiong, Quanzhou Li, Yun-Chun Chen, Homanga Bharadhwaj, Samarth Sinha, and Animesh Garg. Learning by watching: Physical imitation of manipulation skills from human videos. In *International Conference on Intelligent Robots and Systems*, 2021.

[45] Tianhe Yu, Chelsea Finn, Annie Xie, Sudeep Dasari, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot imitation from observing humans via domain-adaptive meta-learning. *arXiv preprint arXiv:1802.01557*, 2018.

[46] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2020.

[47] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6023–6032, 2019.

[48] Kevin Zakka, Andy Zeng, Pete Florence, Jonathan Tompson, Jeannette Bohg, and Debidatta Dwibedi. Xirl: Cross-embodiment inverse reinforcement learning. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2022.

[49] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.