

See to Touch: Learning Tactile Dexterity through Visual Incentives

Irmak Guzey^{1,†}

Yinlong Dai¹

Ben Evans¹

Soumith Chintala²

Lerrel Pinto¹

New York University¹, Meta AI Research²

see-to-touch.github.io

Abstract—Equipping multi-fingered robots with tactile sensing is crucial for achieving the precise, contact-rich, and dexterous manipulation that humans excel at. However, relying solely on tactile sensing fails to provide adequate cues for reasoning about objects’ spatial configurations, limiting the ability to correct errors and adapt to changing situations. In this paper, we present Tactile Adaptation from Visual Incentives (TAVI), a new framework that enhances tactile-based dexterity by optimizing dexterous policies using vision-based rewards. First, we use a contrastive-based objective to learn visual representations. Next, we construct a reward function using these visual representations through optimal-transport based matching on one human demonstration. Finally, we use online reinforcement learning on our robot to optimize tactile-based policies that maximize the visual reward. On six challenging tasks, such as peg pick-and-place, unstacking bowls, and flipping slender objects, TAVI achieves a success rate of 73% using our four-fingered Allegro robot hand. The increase in performance is 108% higher than policies using tactile and vision-based rewards and 135% higher than policies without tactile observational input. Robot videos are best viewed on our project website: <https://see-to-touch.github.io/>.

I. INTRODUCTION

Dexterity has played a crucial role in human development, enabling us to create and utilize tools effectively [1]. Although two-fingered grippers have been extensively studied in the field of robotics [2, 3, 4], they inherently lack the physical capabilities required for performing dexterous tasks that necessitate fine-grained manipulation at the fingertips. These additional capabilities facilitate a wider range of tasks in real-world scenarios; however, they also result in a higher dimensional actions. Furthermore, due to visual occlusion during such manipulation processes, effective utilization of tactile data becomes vital – an aspect that remains under-studied in the context of dexterity.

To train dexterous policies, several frameworks have been proposed, ranging from model-based control, in which models of the robot and object are used to optimize control behavior [5, 6], to simulation-to-reality transfer (sim2real), where a policy is trained in a simulator and then transferred to the real world [7, 8]. While the latter methods demonstrate impressive results, they require the ability to simulate sensory observations during manipulation. This becomes problematic when using rich tactile sensing, as modeling uncalibrated skin sensing is an open research problem in itself [9]. Consequently, much of the prior work in multi-fingered

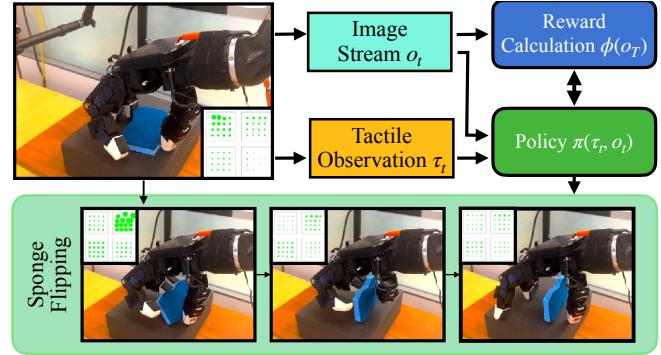


Fig. 1: TAVI learns dexterous policies through online learning. Both tactile and image is used to retrieve action while only image is used for reward calculation.

dexterity relies either exclusively on visual feedback or on weaker binary-touch signals [10].

To address the challenges associated with modeling dexterous behavior, recent approaches have focused on imitation-based methods. In these approaches, humans first teleoperate robots to collect demonstrations of dexterous behavior [11, 12, 13]. Then, offline imitation learning is used to obtain policies that fit these demonstrations. Dexterous policies trained using this approach can solve a wide range of tasks, from reorienting objects to precise picking. Importantly, since policies are trained on real observational data, they readily scale to skin-based tactile data, which is otherwise difficult to model. However, offline imitation is not a silver bullet. First, it requires the collection of significant amounts of difficult-to-collect demonstration data. Second, it needs the demonstrations to densely cover object configurations used in evaluation. Third, it does not have any mechanism to recover from errors during execution.

In this work, we present Tactile Adaptation from Visual Incentives (TAVI), a new framework for tactile-based dexterity that requires only one successful demonstration, can generalize to new object configurations, and can learn to correct behaviors from failures. The key insight in TAVI is to continuously adapt the dexterous policy by improving the Optimal-Transport (OT) match between sensory observations generated by the policy and those generated through human demonstrations. The adaptation algorithm is built on prior work in inverse reinforcement learning (IRL) [14], where the matching score corresponds to rewards and policy opti-

†Correspondence to irmakguzey@nyu.edu.

mization is done through RL.

However, unlike prior work in IRL, where observations can directly provide crisp signals of task completion, the use of tactile observations poses a unique challenge. Tactile signals often lack the necessary cues to reason about the spatial location of objects. For instance, the tactile signal obtained from picking up a slender object is very similar to that of pinch grasping. To address this, we only use visual cues to determine reward, which in turn provides a strong incentive for tactile-based policy learning. Moreover, to improve the quality of visual rewards, we use a novel contrastive learning objective that augments a time-contrastive loss with proprioceptive movements.

We experimentally evaluate TAVI on six contact-rich, dexterous tasks such as opening a mint box, unstacking bowls, and flipping slender objects. Through an extensive study, we present the following key insights:

- 1) TAVI improves upon prior state-of-the-art work in dexterous imitation [13] with an average of $5.5 \times$ improvement in success rate given 30 minutes of online interactions. This represents the first framework to learn dexterous policies from online tactile-based interactions (Section V-C).
- 2) Visual representations learned through our contrastive learning scheme achieve approximately 56% improvement in four of our tasks over prior representation methods on dexterous manipulation trajectories (Section V-D).
- 3) Ablations on different representation modules and sensor combinations show that the design decisions in TAVI are crucial for high performance (Sections V-C, V-D).

Robot videos generated by TAVI are best viewed on our website: <https://see-to-touch.github.io/>.

II. RELATED WORKS

a) Dexterous manipulation and tactile sensing: Control of dexterous, multi-fingered robots has been of longstanding interest to the field [15, 16, 17]. A recent approach is to learn a policy in simulation and transfer to the real world, which requires extensive randomization and does not simulate fine-grained touch sensors [7, 8, 10, 18]. Earlier work focuses on physics-based models of grasping [19, 20] to compute grasp stability from motor torque readings. Unfortunately, these methods are susceptible to noise due to the inherent interconnection between motor sensors and controllers. To mitigate this coupling, a number of tactile sensors have been created to endow robots with touch [21, 22, 23]. One such sensor, GelSight, has been used extensively for tasks like object classification [24], measuring surface properties [25], in-hand rotation [18] and pose estimation [26]. Due to GelSight’s difficulty to cover an entire multifingered hand, ‘skin’-like sensors [27] have been developed. These sensors can cover the entire hand, giving high-resolution tactile information that can aid learning dexterous policies.

b) Learning from tactile data: To leverage high-resolution readings from tactile sensors, a number of learning-based approaches have been used to solve tasks with two-fingered robot grippers [3, 28, 29, 30, 31]. These methods require a large amount of task-oriented data and are not applied to multi-fingered hands. Most similar to our work, T-DEX [13] learns a tactile representation for an entire hand by using self-supervision on a large, task-agnostic play dataset. TAVI uses the pretrained tactile encoder from T-DEX for our tasks.

c) Representation learning for visual observations: Learning meaningful, low-dimensional representations with limited or no data labels is an active area of interest in computer vision [32, 33, 34, 35]. These techniques aim to optimize an auxiliary objective that results in representations that are good for downstream tasks. Some tasks include maintaining consistency between augmentations of the same image [34], reconstruction of patches [36], and making sure similar examples are close to one another [37]. This has been successfully applied on computer vision benchmarks due to the availability of large amounts of unlabeled data [38, 39, 40]. Because of the limited availability of labeled data in robotics, unsupervised and semi-supervised representation learning techniques have grown in popularity for tasks like manipulation [41] and visual imitation [42, 43]. For our experiments, we use an InfoNCE-style [37] loss using time-contrastive [44] pairs to learn visual representations.

d) Online adaptation and imitation learning: Imitation learning (IL) is has been effective for solving real-world tasks [45, 46]. The simplest form, Behavior Cloning (BC) learns policies from offline expert demonstrations and it has been effective especially with large datasets [47, 48]. However, BC struggles with out-of-domain scenarios [49]. Inverse reinforcement learning (IRL) estimates expert reward functions, enhancing policy performance but at the cost of sample efficiency [50]. Many works have sought to improve the efficiency of IRL [50, 51, 52] and to extend it to the visual imitation setting [53, 54, 55, 56]. Our work leverages optimal transport IRL for efficient policy learning from visual inputs [14].

III. BACKGROUND

TAVI builds on several technical ideas in contrastive learning and optimal-transport imitation:

A. Contrastive Self-Supervised Learning

Self-supervised learning (SSL) seeks to learn compact representations for high-dimensional observations, such as images, to be used in downstream tasks. Contrastive methods for SSL seek to move representations between “positive” samples close together while moving “negative” samples further from one another.

InfoNCE [37] is a commonly used loss function employed in contrastive learning that distinguishes positive and negative pairs based on their density ratio. For an observation and its positive pair o_t, o_t^+ and set of n negative observations $\mathcal{D} =$

$\{o_1, \dots, o_n\}$, resulting in latents z_t, z_t^+ and $\{z_1, \dots, z_n\}$, the loss is defined as:

$$\mathcal{L}_{\text{NCE}}(z_t, z_t^+, \{z_1, \dots, z_n\}) = -\mathbb{E}_{\mathcal{D}} \left[\log \frac{h(z_t, z_t^+)}{\sum_{i=1}^n h(z_t, z_i)} \right] \quad (1)$$

where $h(x, y) = \exp(x \cdot y)$. Maximizing this loss causes the model to assign higher probabilities to positive pairs while pushing apart negative, resulting in discriminative representations.

B. Optimal-Transport Imitation Learning

Imitation learning seeks to find a policy from expert demonstrations. Recent methods [53, 57] have used optimal-transport to efficiently imitate expert trajectories from images. One of these methods, FISH [14], takes a weak base policy and an encoder that maps from high dimensional observations to a low dimensional latent space, and learns a residual policy that corrects the base policy by producing corrective offsets. It does this by using an optimal-transport-based reward function between an expert trajectory and the robots executed trajectory. Formally, given an expert trajectory $\mathcal{T}^e = \{o_1^e, \dots, o_T^e\}$ and an observed robot trajectory $\mathcal{T}^r = \{o_1^r, \dots, o_T^r\}$, latent representations for each $\{z_1^e, \dots, z_T^e\}$, $\{z_1^r, \dots, z_T^r\}$ are computed using the given encoder. A pairwise cost matrix between the two representations, C , can then be formed where C_{ij} corresponds to the cost of moving z_i^e to z_j^r . Optimal-transport finds the transport plan μ^* that best matches \mathcal{T}^e and \mathcal{T}^r , where μ_{ij}^* is the score of the match between the i th representation from the expert and j th representation from the robot. The optimal-transport reward is computed as

$$r^{\text{OT}}(o_t^r) = - \sum_{t'=1}^T C_{t,t'} \mu_{t,t'}^* \quad (2)$$

This allows for comparing behaviors in a time-invariant manner. If \mathcal{T}^r exactly matched \mathcal{T}^e , the cost would be zero everywhere and the reward would be maximized. If our robot trajectory was offset, say by repeating the first observation $\mathcal{T}^r = \{o_1^e, o_1^e, \dots, o_{T-1}^e\}$, we would only be lightly penalized because the optimal-transport would find good matches between adjacent observations. DDPG [58] is used to maximize this reward function, resulting in similar behavior to the expert.

IV. TACTILE ADAPTATION THROUGH VISUAL INCENTIVES (TAVI)

First, we collect data on a robot hand equipped with skin-based tactile sensors. Expert demonstrations are collected using the HOLO-DEX framework (Section IV-A). Next, we must obtain visual representations for OT reward calculation. This is done in a self-supervised manner with a modified InfoNCE loss (Section IV-B). Finally, we train a policy online to imitate the expert demonstration using an OT-based reward function with features from the learned visual encoder (Section IV-C).

A. Robot Setup and Expert Data Collection

Our robot is an arm-hand system with a 6-DOF Jaco arm and a 16-DOF AllegroHand (see Figure 1 (a)). The hand is fitted with 15 XELA uSkin tactile sensors [59], each with a 4x4 tri-axial force reading, and we place an RGB camera in the scene to capture visual information. We collect data using the HOLO-DEX framework [12], which uses a VR headset to track hand pose and re-targets to a similar pose on the robot morphology. During data collection, we record the position and orientation of the arm’s end effector, s_t^{arm} , the positions of all of the joints on the hand, s_t^{hand} , as well as the tactile and image information, τ, o . Since the robots and sensors all return data at different frequencies, we align the data using the collected timestamps, and combine the robot states $s_t = s_t^{\text{arm}} \oplus s_t^{\text{hand}}$ to produce aligned tuples (s_t, τ_t, o_t) . Similar to [13], we subsample the data to 10Hz and remove transitions where the cumulative movement is below 1 cm.

B. Representation Learning for Vision and Tactile Observations

In order to mitigate the need for explicit state estimation, we use self-supervised learning to learn a mapping from high dimensional observations to a lower dimensional latent state (see Section III-A for more details). The image encoder, which maps images o_t to latents z_t , is trained on demonstration data for the task and uses a combination of two losses. The first is the InfoNCE [37] loss trained using nearby observations as positive examples, following the methodology of Time-Contrastive Networks [44]. The second loss predicts the change in robot state between nearby observations using a small mlp head, $\hat{\Delta}(z_t, z_{t+k})$. The change loss function is $\mathcal{L}_{\Delta}(s_t, s_{t+k}, z_t, z_{t+k}) = \|s_{t+k} - s_t - \hat{\Delta}(z_t, z_{t+k})\|$ and differs from an inverse model [60] in that we predict a change in state over multiple steps instead of a single action. Our final loss function is

$$\mathcal{L} = \mathcal{L}_{\text{NCE}}(z_t, z_{t+k}, \{z_1, \dots, z_n\}) + \lambda \mathcal{L}_{\Delta}(s_t, s_{t+k}, z_t, z_{t+k})$$

For computational efficiency, we use the same observations for both the positive samples and to predict the change in joint angles, setting $k = 5$ for all our experiments. We scale the losses to be approximately the same magnitude. For the tactile encoder, we download and use a pretrained tactile encoder that was trained on 2.5 hours of tactile-based play data using self-supervised learning [13].

C. Policy Learning through Online Imitation

We utilize the FISH [14] imitation algorithm on a single demonstrated trajectory to learn a policy (see Section III-B for more details). The base policy we choose is simply an open loop rollout of the expert demonstration, executing the previously executed actions in sequence. This policy completely fails if the environment is not the same as in the expert demonstration, but it serves as a decent base from which to learn a residual policy. The offset policy receives both o_t and τ_t , which are augmented with random resized crops before passing to the visual and tactile encoders. Crucially, we only use visual information, o_t to calculate



Fig. 2: Rollouts of trained policies from TAVI on six tasks. Videos are best viewed on our website <https://see-to-touch.github.io/>.

the optimal-transport reward. We find that including the tactile information in the reward results in convergence to sub-optimal solutions like activating the touch sensors by pinching the fingers together. Details of the reinforcement learning (RL) agent that is used in TAVI can be found in Appendix A.

a) Final frame matching for rewards: We differ from FISH in that we do not use the entire length of trajectories to calculate the reward. Instead, we match the last 10 frames of the robot trajectory to the last frame of the expert trajectory. While this does not give us immediate feedback if the executed trajectory differs from the expert, the sparse reward of distance to the expert’s final frame has enough signal to learn to complete the tasks. Including all of the frames results in matching starting robot frames to ending expert frames, and vice-versa, preventing the policy from completing the task. We further explain this behavior in Section V-E and show an illustrative figure in Appendix B.

b) Exploration strategy: Since the method learns a residual policy, we can naturally enable or disable learning on subsets of the action space, i.e., we can explore along only the dimensions of the action space that we need to. We detail which parts of the action space are enabled in Section V. To effectively explore the space, we use additive OU noise [61], which prevents motor jitter.

V. EXPERIMENTAL EVALUATION

We experimentally evaluate TAVI to answer the following questions: (a) How well does TAVI perform on dexterous tasks? (b) Does the contrastive encoder improve visual representation quality? (c) How well does TAVI generalize to unseen objects?

A. Task Descriptions

We experiment with six dexterous tasks that require precise control, visualized in Figure 2.

a) Peg Insertion: The robot must locate and pick up a peg before inserting it into a cup on the table. The cup stays in the same position for all trials. We learn a residual policy on the base joints of all of the fingers.

b) Sponge Flipping: The robot must find a sponge lying flat on the table and manipulate it to balance it on its side. This is challenging since minor errors will result in the object tipping over. We learn a residual policy on the base of the thumb, index, and middle fingers.

c) Eraser Turning: The robot must pick a whiteboard eraser lying flat and rotate it 180° to lie flat on its opposite face. We learn a residual policy on the last two joints of the middle finger and the top three tip joints of the thumb.

d) Bowl Unstacking: The robot must locate a stack of bowls and remove a bowl from the stack. This task requires shear force to separate the bowls from one another. The residual policy learns the side-to-side offset of the arm end effector position and the thumb base joint.

e) Plier Picking: The robot must locate and pick up a pair of pliers on the table. This task is especially difficult due to the precision required when placing the fingers. The residual policy learns offsets for the last pointer joint, the base and tip joints of the middle finger, and the two base joints of the thumb.

f) Mint Opening: The robot must locate and open a metal mint box by using tips of the thumb, index and middle finger. This task requires robot to stabilize the box with the middle and thumb fingers and carefully opening the top of the mint box.

B. Baselines and Evaluation Metrics

We study the effectiveness of our method and compare against the baselines described below:

- 1) **T-DEX** [13]: We implement and run a state-of-the-art method for learning dexterous policies that utilizes self-supervised image and tactile encoders with nearest neighbors imitation. For the sake of fairness, we use the same image encoder used in TAVI.

TABLE I: Success rates of TAVI and our baselines for evaluations run on the Allegro hand.

	T-DEX [13]	BC-BeT [62]	Tactile Only	Image + Tactile Reward	AVI [14]	TAVI
Peg Insertion	2/10	0/10	6/10	6/10	6/10	8/10
Sponge Flipping	1/10	0/10	8/10	4/10	3/10	8/10
Eraser Turning	2/10	0/10	0/10	2/10	0/10	5/10
Bowl Unstacking	1/10	0/10	5/10	0/10	3/10	9/10
Plier Picking	0/10	0/10	4/10	4/10	6/10	7/10
Mint Opening	4/10	0/10	0/10	5/10	1/10	7/10
Avg. Success Rate	0.16	0.0	0.38	0.35	0.31	0.73

- 2) **BC-BeT** [62]: We implement and run a state-of-the-art behavior cloning method Behavior Transformers. Again, we use the same encoders as our method and do not update the encoder parameters during training.
- 3) **Tactile Only**: We only use the tactile representations for inputting to the policy and the calculating the optimal transport reward. This studies our choice of having image representations in TAVI.
- 4) **Tactile and Image Reward**: To study the effect of our choice of reward function, we experiment with calculating the optimal transport reward from both tactile and visual features. We concatenate both features at each timestep and then run OT matching on it.
- 5) **No Tactile information (AVI)** [14]: To study the value of tactile feedback, we train our method without tactile information given to the policy. While the reward calculation is the same as our main method, the policy must infer contact from vision alone.

a) *Evaluating robot performance*: We allow all online imitation methods to train online with one expert demonstration until the reward converges or for up to 30 minutes. We evaluate all methods by running 10 rollouts with varying position and orientation of the manipulated objects. For fairness, we use the same 10 positions for each method.

b) *Evaluating visual representations*: In order to evaluate our approach in learning visual representations we have run robot experiments on 4 of our tasks with 5 different set of encoders. In addition to the vision encoder in TAVI, we evaluate the following encoders on our framework:

- 1) **Contrastive Only** [63]: Similar training framework to TAVI but the loss doesn't include the change loss function. So the final loss only includes the InfoNCE loss between the temporal frames.

$$\mathcal{L} = \mathcal{L}_{\text{NCE}}(z_t, z_{t+k}, \{z_1, \dots, z_n\})$$

- 2) **Joint Difference** [64]: Similar training framework to TAVI but the loss doesn't include the InfoCNE loss function. So the final loss only includes the change loss function.

$$\mathcal{L} = \lambda \mathcal{L}_\Delta(s_t, s_{t+k}, z_t, z_{t+k})$$

- 3) **BYOL** [34]: We use the self-supervised learning algorithm; Bootstrap Your own Latent (BYOL) to train encoders on the task data.
- 4) **BC** [65]: We receive visual representations from a simple 3-layered CNN and map them to actions applied during demonstrations. We train this end-to-end on the task data for each task.

TABLE II: Success rates of different visual representations on TAVI learning framework

Encoder	Plier	Bowl	Sponge	Peg	Average
TAVI	7/10	9/10	7/10	8/10	7.75/10
Contrastive Only [63]	0/10	7/10	2/10	7/10	4/10
Joint Difference [64]	4/10	7/10	6/10	5/10	5.5/10
BYOL [34]	6/10	5/10	9/10	6/10	6/10
BC [65]	0/10	6/10	3/10	6/10	4/10
Pretrained	5/10	6/10	2/10	5/10	4.5/10

- 5) **Pretrained**: We use a Resnet-18 [66] with weights pre-trained on the ImageNet [67] task with no finetuning.

C. How well does TAVI perform on dexterous tasks?

In Table I we report the success rates of TAVI and baselines. We see that BC-BeT is unable to complete any of the tasks, quickly going out of distribution and failing to recover. T-DEX is only able to solve at most 4 of the 10 runs, failing because it is unable to update the policy when the object has moved out of the demonstration set. While the combined image and tactile reward or tactile-only are able to solve more tasks than T-DEX, the noise introduced into the reward from the tactile information halves the success rate when compared to TAVI, highlighting the importance of computing rewards from visual information only.

AVI almost matches the performance of our method on the peg insertion and plier picking tasks, but is unable to succeed at all on eraser turning and mint opening and has degraded performance on the other tasks. Neither peg insertion nor plier picking require precise force feedback to succeed, while eraser turning, mint opening, sponge flipping, and bowl unstacking all require a level of precision, taking care not to exert too much force on the manipulated object(s). These results underscore the importance of incorporating tactile feedback into dexterous policies.

We showcase the TAVI training rollouts and the corresponding OT rewards for each task in Appendix C.

D. Does the contrastive encoder improve visual representation quality?

We report the success rates of experimented visual representations in Table II. We observe that *Pretrained*, *Contrastive Only* and *BC* encoders are not performing well due to failure in capturing the configuration between the object and the robot hand. We observe that *Joint Difference Only* baseline performs relatively well since the encoder learns how to differentiate the impact of the object and actions to the hand pose but not as high as TAVI since it's lacking the temporal information coming from the contrastive loss.

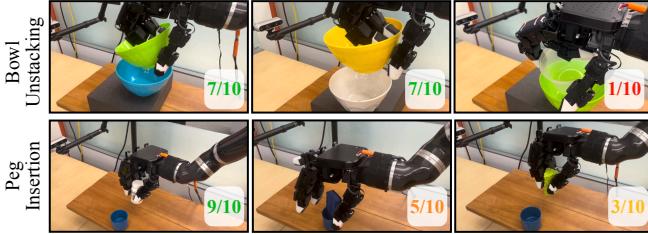


Fig. 3: We show success rates of TAVI on a variety of objects not seen during demonstration collection.

BYOL training on the task data has been our most successful baseline after TAVI, we believe this is due to *BYOL* augmentations being able to force the visual representations to focus on the manipulation. Given the highest score in these experiments we choose to train our encoders with the combined contrastive and joint-prediction loss.

E. How does the number of frames included in the reward impact the results?

As was mentioned in the Section IV-C, we only match the last frame of the expert demonstration and the last 10 frames of the robot trajectory in OT reward calculation. We ran additional experiments with all of our tasks where we include all of the frames of both the robot and the expert trajectory and evaluated the policy with a similar evaluation setup. We show the results of this experiment in Table III.

TABLE III: Success rates of our learning framework with variant number of frames included in reward calculation.

Frames	Bowl	Peg	Sponge	Mint	Plier	Eraser
All frames inc.	4/10	5/10	6/10	0/10	3/10	0/10
TAVI	9/10	8/10	8/10	7/10	7/10	5/10

During OT reward calculation the best plan μ^* that matches two trajectories is not time-dependant, since the matching is done regardless of the timestep of each representation, hence, when all the frames are included to the reward calculation, policy can converge to a local minimum where a failed robot trajectory has high matches with expert trajectories that have similar frames throughout different stages of the trajectory. That is why we observe low performance when all the frames are included to the reward calculation, more details are shown in Appendix B.

F. Does TAVI generalize to new objects?

We study the ability of TAVI to generalize to unseen objects. For each task, we modify the experiment by replacing one of the objects with a new object with different shape, color, and inertial properties. We run 3 new objects (visualised in Figure 3) for the peg insertion and bowl picking tasks, training the policy in the same manner as the original task so it has a chance to adapt. For the bowl unstacking task, we get a success rate of 50% and for the peg insertion task we succeed 57% of the time. The policy is able to generalize on some, but not all of the new objects. When the shape or mass of the object changes substantially,

the policy is not able to offset the fingertips enough from the base policy to complete the task.

G. Can TAVI be used for long-horizon tasks?

Due to very large action space of dexterous hands, training long-horizon tasks is a very challenging problem which is why one of the used approaches is to sequence different sub-policies [68]. In order to sequence sub-policies, each policy learned should also be robust enough for different perturbations coming from each sub-policy.

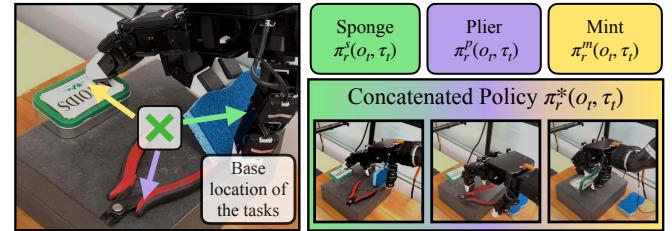


Fig. 4: We show an illustration of our long-horizon policy sequencing. TAVI shows robustness when different tasks are sequenced and successfully applies the learned policies separately.

We evaluated TAVI to see if it gives robust enough policies to sequence different tasks and give longer horizon tasks. We trained 3 separate policies on our Sponge Flipping, Plier Picking and Mint Opening tasks. We have enabled additional axes on wrist positions during the training and concatenated these three policies during rollout. TAVI manages to separately flip the sponge, pick the plier and open the mint box zero-shot. We illustrate this on Figure 4.

H. How robust is TAVI to visual perturbations?

In order to further analyze how image encoders trained in TAVI handles changes in camera view, we ran additional experiments for our task bowl unstacking where we move the camera around 2cm - 15cm with different orientations and trained TAVI with the representations received from those camera positions. We do not collect new expert demonstrations from the new camera positions which causes the episode camera views to gradually drift from the expert.

TABLE IV: Success rates of TAVI with different camera views.

Positional Variations	None	2cm	2-10cm	12cm+orientation
Bowl Unstacking	9/10	6/10	2/10	1/10

We see that with small variations, TAVI is still performative. However, with larger variations, the performance drops significantly as the vision-based representations are not trained to be consistent from multiple-views. This causes calculated rewards to be inconsistent with the success of the robot trajectories which makes the policy harder to train.

VI. LIMITATIONS AND DISCUSSION

In this paper, we introduced TAVI, which leverages tactile feedback for dexterous manipulation through optimal-transport imitation learning. We demonstrated its superior performance compared to visual-only policies, identified

challenges related to tactile information in reward calculation, and examined key components. Despite its current strengths, we acknowledge three limitations. First, our observation representation lacks historical context; incorporating a transformer could enhance performance but requires solving the challenge of training with limited demonstration data. Second, performance of TAVI seems very dependant on the camera view due to the matching between the expert and the trajectory. Incorporating tactile to the reward or training more robust visual representations to different camera views could mitigate this. Finally, the exploration mechanism requires knowing which dimensions in the action space to enable. Automating this process could reduce the need for domain expertise. These areas present exciting opportunities for extending TAVI.

ACKNOWLEDGEMENTS

We thank Aadhithya Iyer, Raunaq Bhirangi, Siddhant Haldar, Jyo Pari and Jeff Cui for valuable feedback and discussions. This work was supported by grants from Honda, Meta, Amazon, and ONR awards N00014-21-1-2758 and N00014-22-1-2773.

REFERENCES

- [1] F. A. Karakostis, D. Haeufle, I. Anastopoulou, K. Moraitis, G. Hotz, V. Tourloukis, and K. Harvati, “Biomechanics of the human thumb and the evolution of dexterity,” *Current Biology*, vol. 31, no. 6, pp. 1317–1325.e8, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0960982220318935>
- [2] C. Ferrari and J. Canny, “Planning optimal grasps,” in *ICRA*, 1992.
- [3] A. Murali, Y. Li, D. Gandhi, and A. Gupta, “Learning to grasp without seeing,” in *Proceedings of the 2018 International Symposium on Experimental Robotics*, J. Xiao, T. Kröger, and O. Khatib, Eds. Cham: Springer International Publishing, 2020, pp. 375–386.
- [4] L. Pinto and A. Gupta, “Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours,” *ICRA*, 2016.
- [5] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine, “Learning complex dexterous manipulation with deep reinforcement learning and demonstrations,” in *RSS*, 2018.
- [6] A. Nagabandi, K. Konoglie, S. Levine, and V. Kumar, “Deep dynamics models for learning dexterous manipulation,” *arXiv*, 2019.
- [7] OpenAI, M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, and W. Zaremba, “Learning dexterous in-hand manipulation,” 2019.
- [8] A. Handa, A. Allshire, V. Makoviychuk, A. Petrenko, R. Singh, J. Liu, D. Makoviichuk, K. Van Wyk, A. Zhurkevich, B. Sundaralingam, Y. Narang, J.-F. Lafleche, D. Fox, and G. State, “Dextreme: Transfer of agile in-hand manipulation from simulation to reality,” *arXiv*, 2022.
- [9] H. Lee, H. Park, G. Serhat, H. Sun, and K. J. Kuchenbecker, “Calibrating a soft eft-based tactile sensor with a multiphysics model and sim-to-real transfer learning,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1632–1638.
- [10] Z.-H. Yin, B. Huang, Y. Qin, Q. Chen, and X. Wang, “Rotating without seeing: Towards in-hand dexterity through touch,” *arXiv preprint arXiv:2303.10880*, 2023.
- [11] S. P. Arunachalam, S. Silwal, B. Evans, and L. Pinto, “Dexterous imitation made easy: A learning-based framework for efficient dexterous manipulation,” *arXiv preprint arXiv:2203.13251*, 2022.
- [12] S. P. Arunachalam, I. Güzey, S. Chintala, and L. Pinto, “Holodex: Teaching dexterity with immersive mixed reality,” 2022.
- [13] I. Güzey, B. Evans, S. Chintala, and L. Pinto, “Dexterity from touch: Self-supervised pre-training of tactile representations with robotic play,” 2023.
- [14] S. Haldar, J. Pari, A. Rai, and L. Pinto, “Teach a robot to fish: Versatile imitation from one minute of demonstrations,” 2023.
- [15] M. T. Ciocarlie, C. Goldfeder, and P. K. Allen, “Dexterous grasping via eigengrasps : A low-dimensional approach to a high-complexity problem,” in *Dexterous Grasping via Eigengrasps : A Low-dimensional Approach to a High-complexity Problem*, 2007.
- [16] V. Kumar, Y. Tassa, T. Erez, and E. Todorov, “Real-time behaviour synthesis for dynamic hand-manipulation,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 6808–6815.
- [17] S. Shigemi, *ASIMO and Humanoid Robot Research at Honda*. Dordrecht: Springer Netherlands, 2018, pp. 1–36. [Online]. Available: https://doi.org/10.1007/978-94-007-7194-9_9-2
- [18] H. Qi, B. Yi, S. Suresh, M. Lambeta, Y. Ma, R. Calandra, and J. Malik, “General in-hand object rotation with vision and touch,” *arXiv preprint arXiv:2309.09979*, 2023.
- [19] A. M. Okamura, N. Smaby, and M. R. Cutkosky, “An overview of dexterous manipulation,” in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 1. IEEE, 2000, pp. 255–262.
- [20] L. U. Odhner, L. P. Jentoft, M. R. Claffee, N. Corson, Y. Tenzer, R. R. Ma, M. Buehler, R. Kohout, R. D. Howe, and A. M. Dollar, “A compliant, underactuated hand for robust manipulation,” *The International Journal of Robotics Research*, vol. 33, no. 5, pp. 736–752, 2014.
- [21] S. Wang, Y. She, B. Romero, and E. Adelson, “Gelsight wedge: Measuring high-resolution 3d contact geometry with a compact robot finger,” 2021. [Online]. Available: <https://arxiv.org/abs/2106.08851>
- [22] R. M. Bhirangi, T. L. Hellebrekers, C. Majidi, and A. Gupta, “Reskin: versatile, replaceable, lasting tactile skins,” *CoRR*, vol. abs/2111.00071, 2021. [Online]. Available: <https://arxiv.org/abs/2111.00071>
- [23] A. Alspach, K. Hashimoto, N. Kuppuswamy, and R. Tedrake, “Soft-bubble: A highly compliant dense geometry tactile sensor for robot manipulation,” in *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*. IEEE, 2019, pp. 597–604.
- [24] R. Patel, R. Ouyang, B. Romero, and E. Adelson, “Digger finger: Gelsight tactile sensor for object identification inside granular media,” in *Experimental Robotics: The 17th International Symposium*. Springer, 2021, pp. 105–115.
- [25] S. Dong, W. Yuan, and E. H. Adelson, “Improved gelsight tactile sensor for measuring geometry and slip,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 137–144.
- [26] T. Kelestemur, R. Platt, and T. Padir, “Tactile pose estimation and policy learning for unknown object manipulation,” *arXiv preprint arXiv:2203.10685*, 2022.
- [27] R. Dahiya, N. Yugeswaran, F. Liu, L. Manjakkal, E. Burdet, V. Hayward, and H. Jörn tell, “Large-area soft e-skin: The challenges beyond sensor designs,” *Proceedings of the IEEE*, vol. 107, no. 10, pp. 2016–2033, 2019.
- [28] R. Calandra, A. Owens, D. Jayaraman, J. Lin, W. Yuan, J. Malik, E. H. Adelson, and S. Levine, “More than a feeling: Learning to grasp and regrasp using vision and touch,” *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3300–3307,

- 2018.
- [29] M. Zambelli, Y. Aytar, F. Visin, Y. Zhou, and R. Hadsell, “Learning rich touch representations through cross-modal self-supervision,” in *Conference on Robot Learning*. PMLR, 2021, pp. 1415–1425.
- [30] Y. She, S. Wang, S. Dong, N. Sunil, A. Rodriguez, and E. Adelson, “Cable manipulation with a tactile-reactive gripper,” 2019. [Online]. Available: <https://arxiv.org/abs/1910.02860>
- [31] S. Wang, J. Wu, X. Sun, W. Yuan, W. T. Freeman, J. B. Tenenbaum, and E. H. Adelson, “3d shape perception from monocular vision, touch, and shape priors,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1606–1613.
- [32] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” *arXiv preprint arXiv:2002.05709*, 2020.
- [33] X. Chen, H. Fan, R. Girshick, and K. He, “Improved baselines with momentum contrastive learning,” *arXiv preprint arXiv:2003.04297*, 2020.
- [34] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, et al., “Bootstrap your own latent-a new approach to self-supervised learning,” *NeurIPS*, 2020.
- [35] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, “Unsupervised learning of visual features by contrasting cluster assignments,” *arXiv preprint arXiv:2006.09882*, 2020.
- [36] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, “Masked autoencoders are scalable vision learners,” 2021.
- [37] A. van den Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” 2019.
- [38] A. Bardes, J. Ponce, and Y. LeCun, “Vicreg: Variance-invariance-covariance regularization for self-supervised learning,” *arXiv preprint arXiv:2105.04906*, 2021.
- [39] D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, and A. Zisserman, “With a little help from my friends: Nearest-neighbor contrastive learning of visual representations,” *arXiv preprint arXiv:2104.14548*, 2021.
- [40] M. Assran, Q. Duval, I. Misra, P. Bojanowski, P. Vincent, M. Rabbat, Y. LeCun, and N. Ballas, “Self-supervised learning from images with a joint-embedding predictive architecture,” 2023.
- [41] L. Manuelli, Y. Li, P. Florence, and R. Tedrake, “Keypoints into the future: Self-supervised correspondence in model-based reinforcement learning,” *arXiv preprint arXiv:2009.05085*, 2020.
- [42] S. Young, D. Gandhi, S. Tulsiani, A. Gupta, P. Abbeel, and L. Pinto, “Visual imitation made easy,” 2020.
- [43] J. Pari, N. M. Shafullah, S. P. Arunachalam, and L. Pinto, “The surprising effectiveness of representation learning for visual imitation,” 2021.
- [44] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, and S. Levine, “Time-contrastive networks: Self-supervised learning from video,” *Proceedings of International Conference in Robotics and Automation (ICRA)*, 2018. [Online]. Available: <http://arxiv.org/abs/1704.06888>
- [45] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, “A survey of robot learning from demonstration,” *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.
- [46] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, “Imitation learning: A survey of learning methods,” *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, pp. 1–35, 2017.
- [47] D. Pomerleau, “An autonomous land vehicle in a neural network,” *Advances in Neural Information Processing Systems*, vol. 1, 1998.
- [48] F. Torabi, G. Warnell, and P. Stone, “Recent advances in imitation learning from observation,” *arXiv preprint arXiv:1905.13566*, 2019.
- [49] S. Ross, G. Gordon, and D. Bagnell, “A reduction of imitation learning and structured prediction to no-regret online learning,” in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011, pp. 627–635.
- [50] I. Kostrikov, K. K. Agrawal, D. Dwibedi, S. Levine, and J. Tompson, “Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning,” *arXiv preprint arXiv:1809.02925*, 2018.
- [51] J. Fu, K. Luo, and S. Levine, “Learning robust rewards with adversarial inverse reinforcement learning,” *arXiv preprint arXiv:1710.11248*, 2017.
- [52] H. Xiao, M. Herman, J. Wagner, S. Ziesche, J. Etesami, and T. H. Linh, “Wasserstein adversarial imitation learning,” *arXiv preprint arXiv:1906.08113*, 2019.
- [53] S. Haldar, V. Mathur, D. Yarats, and L. Pinto, “Watch and match: Supercharging imitation with regularized optimal transport,” *arXiv preprint arXiv:2206.15469*, 2022.
- [54] E. Cetin and O. Celiktutan, “Domain-robust visual imitation learning with mutual information constraints,” *arXiv preprint arXiv:2103.05079*, 2021.
- [55] S. Toyer, R. Shah, A. Critch, and S. Russell, “The magical benchmark for robust imitation,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 18284–18295, 2020.
- [56] R. Rafailov, T. Yu, A. Rajeswaran, and C. Finn, “Visual adversarial imitation learning using variational models,” *Advances in Neural Information Processing Systems*, vol. 34, 2021.
- [57] S. Cohen, B. Amos, M. P. Deisenroth, M. Henaff, E. Vinitsky, and D. Yarats, “Imitation learning from pixel observations for continuous control,” 2022. [Online]. Available: <https://openreview.net/forum?id=JLbXkHkLCG6>
- [58] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, “Continuous control with deep reinforcement learning,” *arXiv preprint*, 2015.
- [59] T. P. Tomo, M. Regoli, A. Schmitz, L. Natale, H. Kristanto, S. Somlor, L. Jamone, G. Metta, and S. Sugano, “A new silicone structure for uskin—a soft, distributed, digital 3-axis skin sensor and its integration on the humanoid robot icub,” *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2584–2591, 2018.
- [60] P. Agrawal, A. V. Nair, P. Abbeel, J. Malik, and S. Levine, “Learning to poke by poking: Experiential learning of intuitive physics,” in *NIPS*, 2016.
- [61] G. E. Uhlenbeck and L. S. Ornstein, “On the theory of the brownian motion,” *Phys. Rev.*, vol. 36, pp. 823–841, Sep 1930. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRev.36.823>
- [62] N. M. M. Shafullah, Z. J. Cui, A. Altanza, and L. Pinto, “Behavior transformers: Cloning \$k\\$ modes with one stone,” in *Advances in Neural Information Processing Systems*, 2022. [Online]. Available: <https://openreview.net/forum?id=agTr-vRQsa>
- [63] A. v. d. Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” *arXiv preprint*, 2018.
- [64] D. Brandfonbrener, O. Nachum, and J. Bruna, “Inverse dynamics pretraining learns good representations for multitask imitation,” *arXiv preprint arXiv:2305.16985*, 2023.
- [65] D. A. Pomerleau, “Alvinn: An autonomous land vehicle in a neural network,” in *NeurIPS*, 1989, pp. 305–313.
- [66] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [67] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.
- [68] Y. Chen, C. Wang, L. Fei-Fei, and C. K. Liu, “Sequential

- dexterity: Chaining dexterous policies for long-horizon manipulation,” in *Conference on Robot Learning*, 2023.
- [69] D. Yarats, R. Fergus, A. Lazaric, and L. Pinto, “Mastering visual continuous control: Improved data-augmented reinforcement learning,” *arXiv preprint arXiv:2107.09645*, 2021.

APPENDIX

A. Model Details

We use DrQv2 [69] as the reinforcement learning (RL) algorithm to train our policy. The input to the policy is the concatenation of the tactile and image representations. This learner uses DDPG [58] to maximize the reward function. We showcase the parameters and details used in Table V

Parameter	Value
Optimizer	Adam
Learning Rate	$1e^{-4}$
Standard Dev. Schedule	$1e^{-1}$
Standard Dev. Clip	$3e^{-1}$
Critic Target Tau	$1e^{-2}$
Update Actor Freq.	4
Update Critic Freq	2
Update Critic Target Freq.	4
Batch Size	256
Replay Buffer Size	150000
Exploration Steps	1000
Aug. (Image)	RandomShiftsAug pad = 4
Expert Frame Matches	1
Episode Frame Matches	10

TABLE V: DrQv2 Hyperparameters.

B. Reward Details

In order to further support our decision on choosing the last 10 frames of the episode and the last frame of the expert demonstration, we show the cost matrix C_{ij} when all of the frames are included in Figure 5 for a failed and a successful trajectory. Both of these trajectories receive the reward of -11 with this way of calculation. We observe that when all the frames are included, due to the time independant nature of optimal transport, when there are close representations in different times of the rollouts we receive high scores matches. This problem mostly arises when the hand pose of the trajectory and the expert rollout are similar whereas the objects are in different positions.

In order to tackle this we are only include

C. Training Rollouts

We showcase the training rollouts of each task and the corresponding rewards for each rollout in Figures [6, 7, 8, 9, 10, 11].

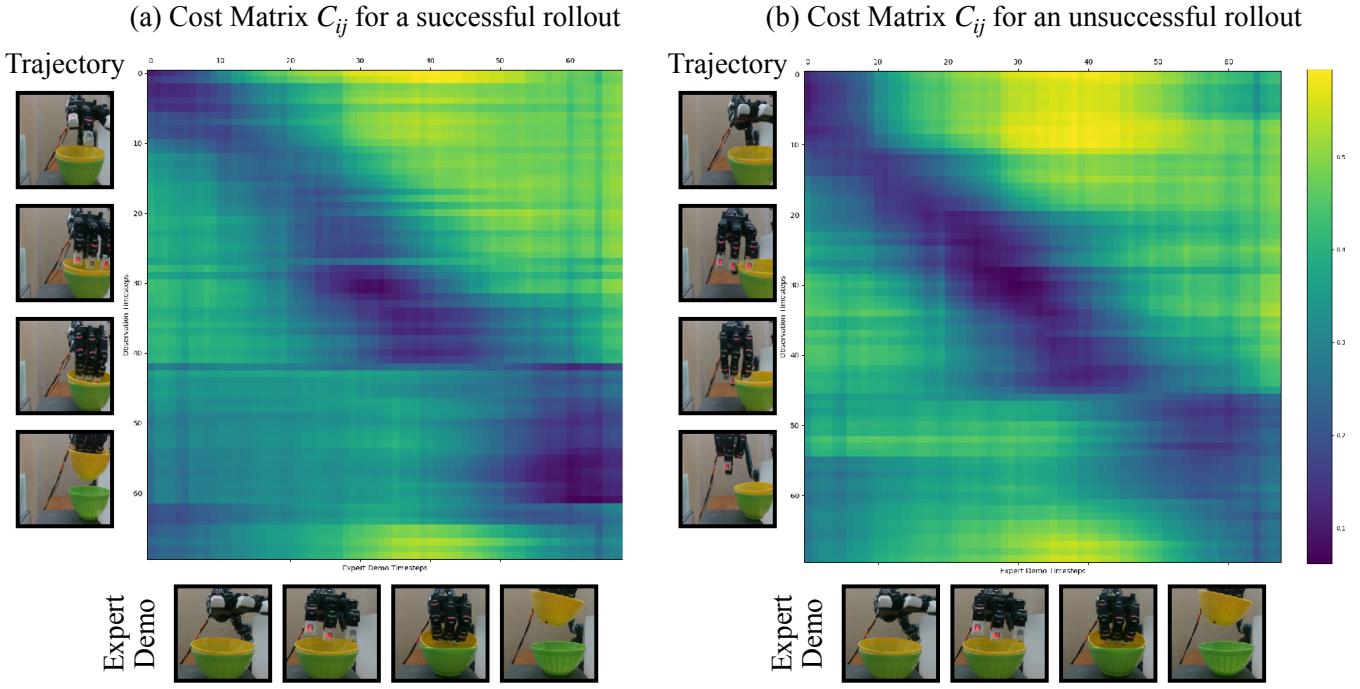


Fig. 5: Cost matrix C_{ij} for a failed and a successful trajectory. Darker colors represent low costs and lighter colors represent higher costs. Note the large area of darker colors at the middle of the unsuccessful rollout and the larger area of darker colors at the end of the successful rollout. When OT matching is applied these low cost areas compensate for each other giving an equal reward of -11 for both of these demonstrations. Also note the similarity of the hand pose between the unsuccessful and the expert demonstration which explains the similarity of the representations.

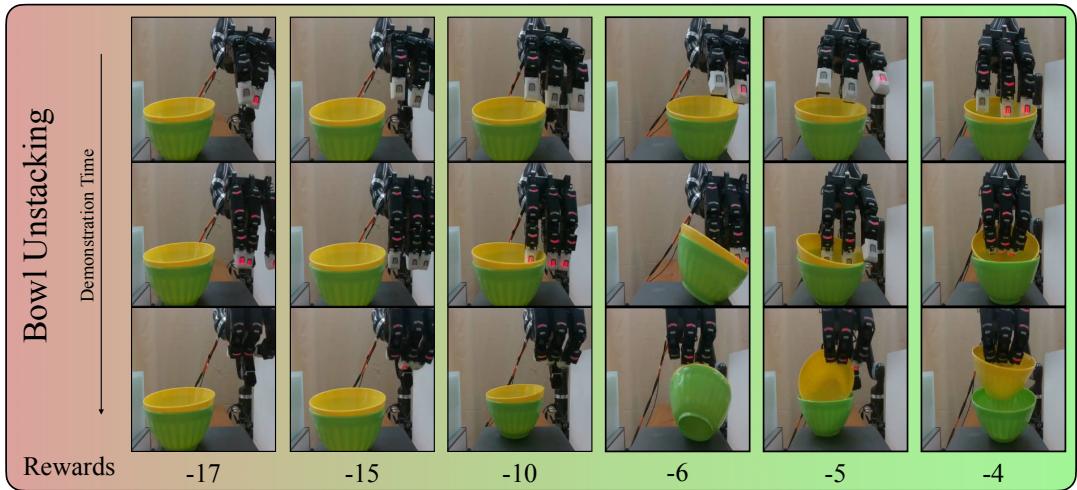


Fig. 6: Additional rollouts and corresponding rewards for the Bowl Unstacking task. Note the increase in the reward as the policy improves.

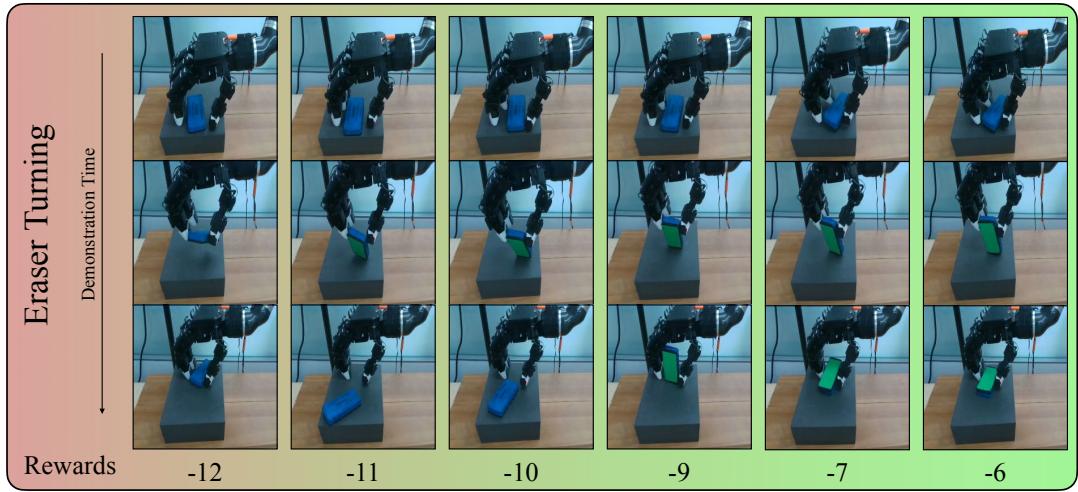


Fig. 7: Additional rollouts and corresponding rewards for the Eraser Turning task. Note the increase in the reward as the policy improves.

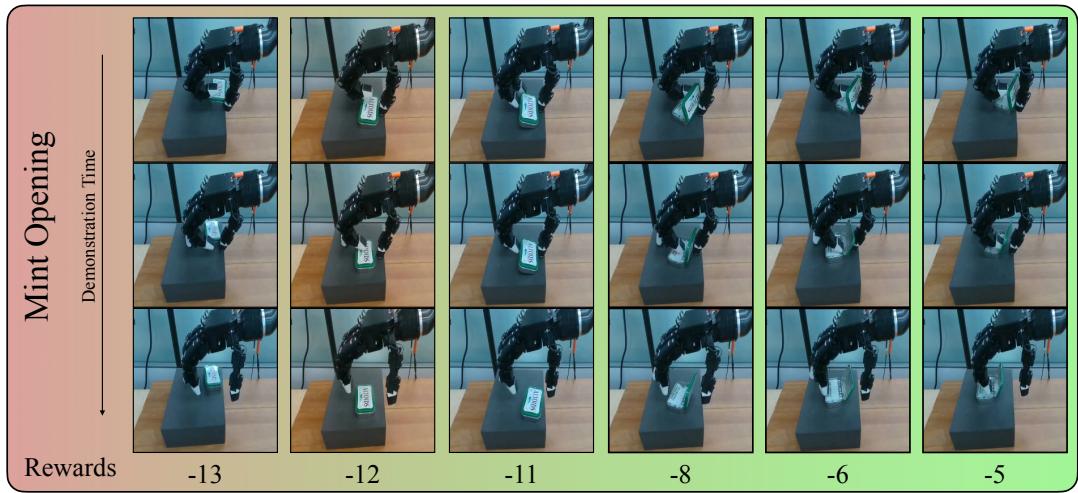


Fig. 8: Additional rollouts and corresponding rewards for the Mint Opening task. Note the increase in the reward as the policy improves.

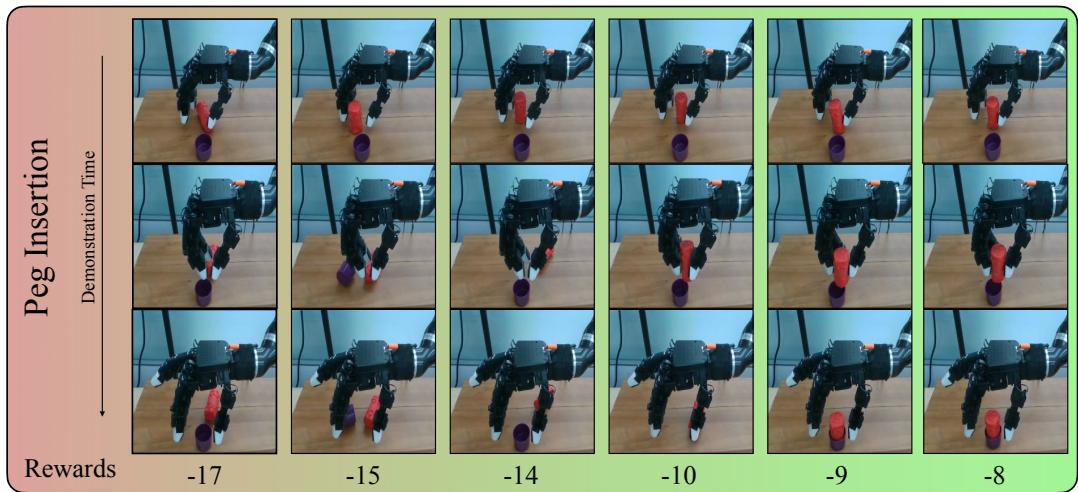


Fig. 9: Additional rollouts and corresponding rewards for the Peg Insertion task. Note the increase in the reward as the policy improves.

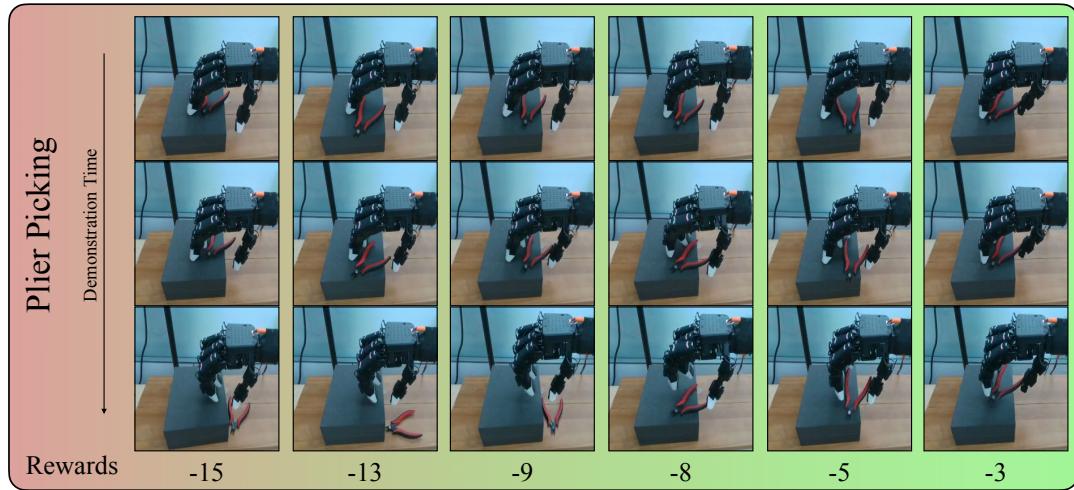


Fig. 10: Additional rollouts and corresponding rewards for the Plier Picking task. Note the increase in the reward as the policy improves.

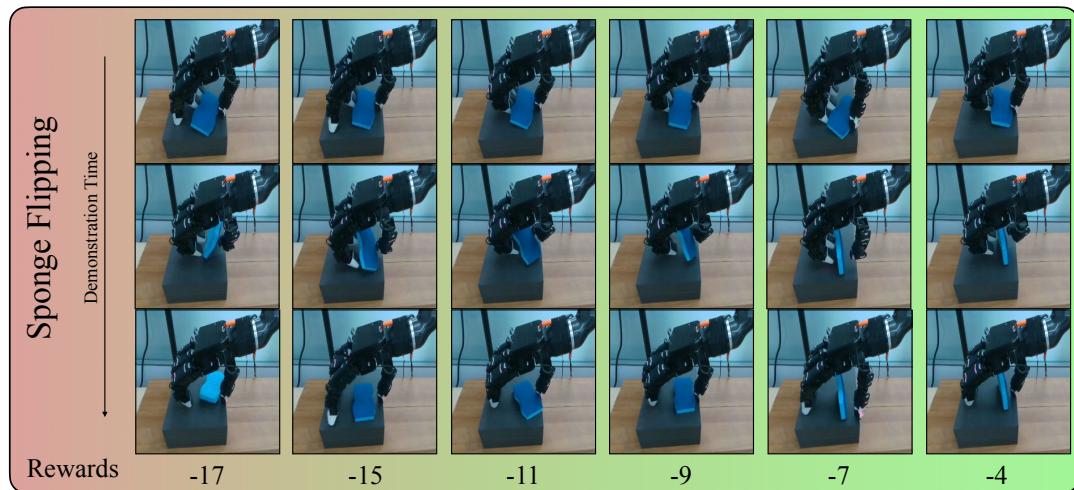


Fig. 11: Additional rollouts and corresponding rewards for the Sponge Flipping task. Note the increase in the reward as the policy improves.