Ergodic LfD: Learning from what to do and what *not* to do

Aleksandra Kalinowska*, Ahalya Prabhakar*, Kathleen Fitzsimons, Todd Murphey
Department of Mechanical Engineering, Northwestern University

***Abstract*— With growing access to versatile robotics, it is beneficial for end users to be able to teach robots tasks without needing to code a control policy. One possibility is to teach the robot through successful task executions. However, near-optimal demonstrations of a task can be difficult to provide and even successful demonstrations can fail to capture task aspects key to robust skill replication. Here, we propose a learning from demonstration (LfD) approach that enables learning of robust task definitions without the need for near-optimal demonstrations. We present a novel algorithmic framework for learning task specifications based on the ergodic metric—a measure of information content in motion. Moreover, we make use of negative demonstrations—demonstrations of what *not* to do—and show that they can help compensate for imperfect demonstrations, reduce the number of demonstrations needed, and highlight crucial task elements improving robot performance. In a proof-of-concept example of cart-pole inversion, we show that negative demonstrations alone can be sufficient to successfully learn and recreate a skill. Through a human subject study with 24 participants, we show that consistently more information about a task can be captured from combined positive and negative (*posneg*) demonstrations than from the same amount of just positive demonstrations. Finally, we demonstrate our learning approach on simulated tasks of target reaching and table cleaning with a 7-DoF Franka arm. Our results point towards a future with robust, data-efficient LfD for novice users.**

I. Introduction

Many assistive robots being deployed in people's homes or on factory floors are capable of performing a wide variety of tasks. As such, it is beneficial for end users to be able to customize these robots by teaching them tasks specific to their needs. However, it is often not possible to provide high-quality task demonstrations. This could be because the task is challenging for a person to perform, e.g., cart-pole inversion due to its unintuitive dynamics, or the person is limited by a low-dimensional control interface, such as a joystick, for providing demonstrations to a 7-DoF robotic arm. Although successful approaches exist for imitation learning, including Dynamic Motion Primitives (DMPs) [1], inverse reinforcement learning [2], and others—as we describe in more detail in Section II—few of the LfD frameworks allow for reliable learning from novice task demonstrations.

Our approach stems from the idea that one can characterize movement by asking how much information about a task is encoded in motion—quantifying this using a measure of ergodicity. We propose ergodic LfD for robust learning from imperfect demonstrations. In this approach, we define tasks through spatial distributions in state-based feature space. Through successive demonstrations, we learn the underlying distribution corresponding to a task and generate robot behavior via ergodic control [3] with respect to the learned distributions. This learning framework allows us to combine multiple novice demonstrations into a successful objective and use model predictive control (MPC) to recreate trajectories for new, previously unencountered scenarios. A desirable property of the proposed method is that demonstrations need not be either temporally aligned or of the same duration. As a result, the information content of the demonstrations is additive without temporal pre-processing. Finally, it is worth noting that ergodic LfD does not focus on imitating trajectories directly—instead it emphasizes imitating trajectory *statistics*. As a result, the method learns well from imperfect demonstrations and is robust to noise in individual demonstrations (e.g., corrective motions or perturbations).

Moreover, we propose imitation learning using negative demonstrations. In some cases, it might be easier for a person to demonstrate what *not* to do rather than to provide an exemplary task execution. In other scenarios, aspects of a task might not be apparent from a successful demonstration and so presenting common pitfalls might be more valuable than repetitively showing the same correct movements. Demonstrating things to avoid is something that people already intuitively do when teaching new skills to others. As we show in this work, robotic LfD can also largely benefit from incorporating negative demonstrations into the learning process. What is more, ergodic LfD is a particularly suitable algorithmic framework, because it enables combining positive and negative demonstrations into a well-posed task objective.

As part of this study, we validate our learning approach on two test beds: a virtual 2-dimensional cart-pole system and a simulated 7-DoF robotic arm. We find that ergodic LfD (1) enables robust skill reconstruction that outperforms the provided demonstrations and (2) generalizes to different robot tasks. Additionally, we test the utility of negative demonstrations in an experiment with 24 participants. Our results show that there is consistent benefit to soliciting combined *posneg* demonstrations compared to only positive demonstrations. The algorithmic framework, more detailed experimental results, and a discussion of future work are described in Sections III, IV, and V, respectively.

II. Related Work

Machine learning techniques such as inverse reinforcement learning (IRL) can be used to generate a reward function that reflects the policy for the task [2], [4]–[6]. While these methods can successfully represent and learn various skills, they cannot generate safety guarantees for the resulting

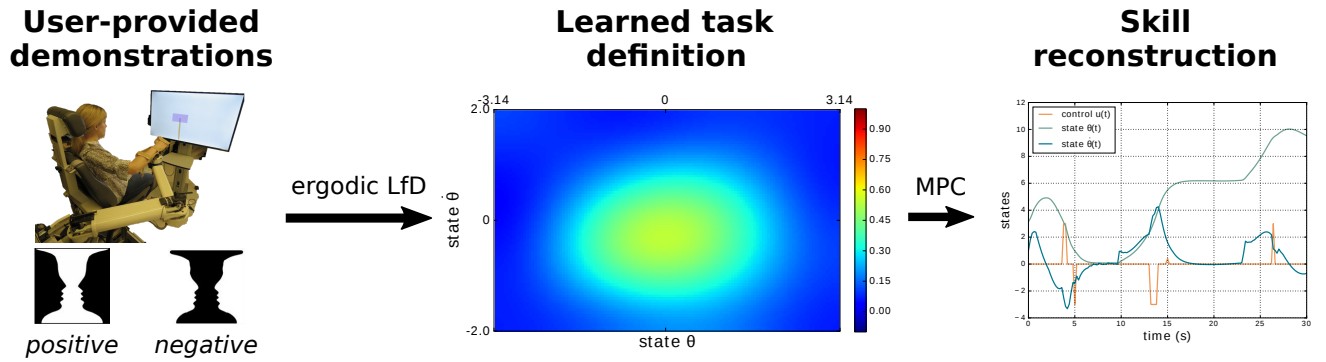

Fig. 1: An overview of the learning process using ergodic LfD on the example of the cart-pole inversion task. Positive and negative demonstrations are combined to form *posonly*, *negonly*, or *posneg* task definitions.

learned policy nor guarantee the dynamic feasibility of the generated trajectories. In contrast, the proposed algorithm inherits formal algorithmic properties from ergodic control and standard MPC methods. These formal properties are as follows: (1) The ergodic cost is globally convex w.r.t. distributions so long as the metric used is on a Sobolev space. The distance from ergodicity can be measured by several metrics, see [7] or [8]. Here, we use the spectral approach as in [9]. (2) The ergodic LfD approach inherits asymptotic convergence from ergodic control [3]. In the cart-pole example described in Section III, this implies that when the goal distribution is defined as a delta function at the unstable equilibrium, the statistics of the trajectory will asymptotically approach the delta function—the pole could occasionally fall, but the amount of time spent at the inverted equilibrium would approach 100% as time goes to infinity. (3) Safety sets can be specified through the use of barrier functions. This property is inherited from standard MPC methods rather than ergodic control specifically.

Furthermore, IRL and inverse optimal control methods assume that the demonstrations representing the task are generated by an expert demonstrator providing optimal (or near-optimal) solutions for the task in order to generate feasible solutions [2], [4]–[6]. Only some proposed approaches allow for non-optimal input, e.g., by executing a number of imitation learning iterations before switching to reinforcement learning methods [10]. Other approaches—such as those that use probabilistic methods to represent a task from demonstrations and replicate robot motions based on that representation—also rely on highly skilled demonstrators, accounting for imperfections with relatively small-scale noise in the probabilistic representation. These methods include dynamical movement primitives (DMP) [11], probabilistic movement primitives (ProMP) [12], Fourier movement primitives [13], Gaussian mixture regression (GMR) [14], Gaussian process regression (GPR) [15], and GMR-based Gaussian process regression (GMR-GP) [16]. In our approach, we explicitly allow for suboptimal demonstrations and treat them as positive (as long as they ultimately achieve the task). A unique aspect of the current work is that one can improve task learning by incorporating negative demonstrations—unsuccessful task executions representing explicit examples of what not to do. This means that users can teach a task when they have difficulty providing even suboptimal demonstrations.

To enable task learning that more closely captures human preferences and that accounts for imperfect or incomplete demonstrations, active learning methods have been developed [17]–[20]. In these approaches, the human is treated as an oracle that the autonomy can query, improving learning quality. However, there is an inherent cost of time and effort to querying the user for corrections and previous studies have found that the preference-learning process can be prohibitively frustrating [21], [22]. Moreover, the user tends to have a preference towards online learning approaches (e.g., [20]) that do not require *post-hoc* corrections to learned robot policies. In this paper, we propose a novel learning method that takes advantage of people's ability to convey information about a task through demonstrations of what *not* to do during runtime and we present an LfD algorithm that allows for combining positive and negative demonstration into a successful task definition.

Lastly, we note that existing methods inherently specify the task representation in a time-dependent manner. As a result, they require temporal modulation or pre-processing to align the demonstrations for task learning; the time-dependent specification makes it difficult to incorporate negative task demonstrations. Some approaches, such as behavioral cloning via DMPs, do not even enable learning a task objective and as a result allow skill reconstruction only in open-loop w.r.t. the task goal (and closed-loop w.r.t. the behavior). We propose ergodic LfD as one possible approach that enables defining objectives as state distributions. While existing methods, such as IRL, could be adapted to learn an objective function over distributions, the nominal complexity of IRL is already exponential ($O(n^2 log(nk))$ [23]. If one were to perform IRL over the set of distributions, the algorithm would further increase in computational complexity likely becoming intractable. Ergodic LfD allows us to avoid the computational complexity of IRL, as well as other pitfalls, such as issues with converging on local minima due to many possible reward functions. What is more,

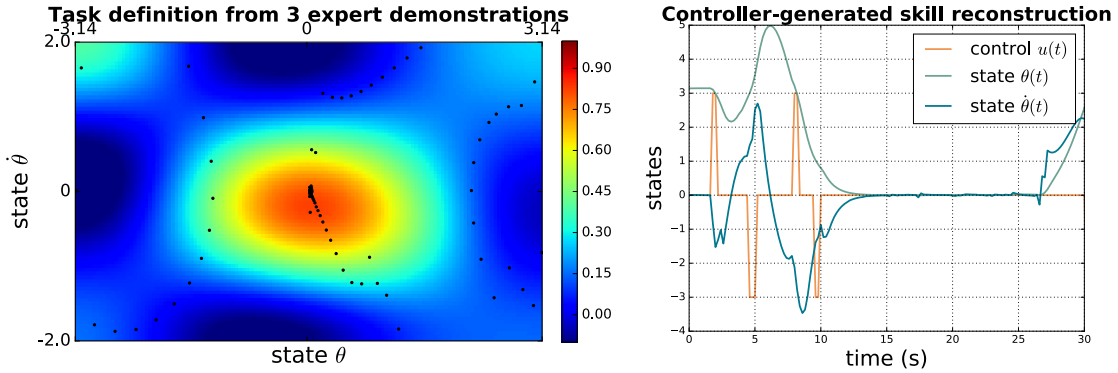

Fig. 2: Example task definition (left) and skill reconstruction (right) learned from 3 expert trajectories. An optimal controller is used to recreate the task given the learned goal distribution. Green indicates success—time when the cart-pole is inverted. The controller-generated trajectory is plotted on the right and overlayed on the task distribution on the left with black dots—note that the trajectory closely represents the underlying distribution subject to constraints imposed by system dynamics.

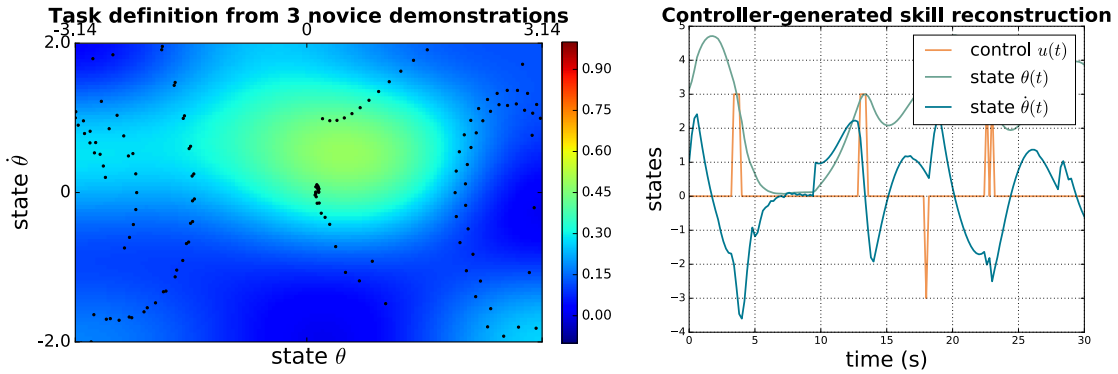

Fig. 3: Example task definition and skill reconstruction learned from 3 novice trajectories from Subject 6. Note that the controller just barely succeeds (for less than 5 seconds), exhibiting comparable performance to the original demonstrations, which had an average success time of 5.6 seconds.

the proposed framework allows for positive and negative distributions to be simply added together. In the results section, we focus on showcasing the value of incorporating negative demonstrations into the learning process.

III. Methods

A. Ergodic Task Definitions and Control

In ergodic LfD, we generate a representation of an unknown task using spatial statistics. Since we avoid specifying temporal dependencies, we synthesize robotic controls that successfully achieve a task without necessarily replicating the demonstration's time-evolving trajectory. We define a single demonstration, represented as d_i, as the distribution of points in the state space making up the the state trajectory $x(t)$ for a given set of time $t \in [t_0, t_f]$, and the set of demonstrations as $D = d_1, ..., d_m$. This set may contain both positive and negative demonstrations, so we also store a label array $E = e_1, ..., e_m$ corresponding in length to D.

Learning from positive demonstrations. A positive demonstration is defined to be a person's attempt at a task that is at least somewhat successful. Oftentimes user-provided positive demonstrations are incomplete or highly sub-optimal with multiple attempts at the task and corrective actions within the demonstration. This was the case in the user studies used in this work.

During the task learning process, we use the demonstration trajectories to generate a task definition $\phi(x)$ by representing the spatial statistics of each demonstration trajectory d with the Fourier decomposition ϕ_k, as described in Eq. 2. We then average the ϕ_k values of the demonstrations to represent the collective spatial statistics of all the demonstrations. Regions of the state space where more time is spent in the trajectories have a higher density in the distribution than regions where less time is spent. If a set of demonstrations for the task of reaching an equilibrium state were used to generate a distribution, there would be a peak at the equilibrium state. As more demonstrations are added, the collective time spent at the equilibrium state would generate a higher peak at the state s, asymptotically approaching a delta function. Examples of the cart-pole inversion task learned from an expert demonstration and from novice imperfect demonstrations are showed in Fig. 2 and Fig. 3, respectively.

To generate the distribution $\phi(x)$ from the demonstration trajectories $x(t)$, we calculate spatial Fourier coefficients of

$x(t)$ using Fourier basis functions of the form

$$F_k(x) = \frac{1}{h_k} \prod_{i=1}^{n} cos\left(\frac{k_i \pi}{L_i} x_i\right), \tag{1}$$

where k is a multi-index over n dimensions, h_k is a normalizing factor [9], and L_i is a measure of the length of the dimension. We then compute the coefficients of a time-averaged trajectory using Eq. 2.

$$c_k = \frac{1}{T} \int_0^T F_k(x(t))dt \tag{2}$$

The coefficients of the demonstration trajectories are combined to form the coefficients that describe the task definition.

$$\phi_k = \sum_{j=1}^{m} w_j c_{k,j} \tag{3}$$

The demonstrations may be weighted depending on the relative quality or may be given equal weight by setting $w_j = 1/m$ for each demonstration as is done in this paper. Note that other representations of a distribution could be used instead of Fourier coefficients, including wavelets or Gaussian Mixtures, as long as a comparison metric of two distributions can be defined and meets the conditions for global convexity.

Learning from negative demonstrations. In this work, we also employ negative demonstrations, defined as both unsuccessful task attempts and explicit demonstrations of what *not* to do. Negative demonstrations can include good-faith attempts at a task where the demonstrator performs poorly, or explicit examples of actions that are far from the desired behavior. As with positive demonstrations, the demonstrated trajectories are represented by the Fourier decomposition c_k calculated using Eq. 2. However, they are combined through subtraction—$w_j < 0$ in Eq. 3—such that regions of the state space where more time is spent in the trajectories have a lower density in the distribution than regions where less time is spent. We show that negative demonstrations can be used to both construct a successful representation of a task by themselves or to improve a task definition when used in conjunction with positive demonstrations (see Fig. 4 and 5 for examples). Mathematically, we define the *posneg* distribution as $\phi_{posneg} = \gamma_1 \phi_{pos} - \gamma_2 \phi_{neg}$, while *negonly* distributions learned from just negative demonstrations are defined by subtracting ϕ_{neg} from a uniform distribution. γ_1 and γ_2 represent normalization factors that weigh the contribution of positive and negative demonstrations to the final task definition.

Ergodic Control. When recreating the learned skills, we use a model predictive controller (MPC) to synthesize controls that generate a trajectory to match the spatial statistics of the distribution representing the demonstration set. In defining the task objective, we use ergodicity, which relates the temporal behavior of a signal to a pre-defined distribution. Ergodicity can be measured by several metrics [7], [8]; here we use the spectral approach [9], which characterizes ergodicity by comparing spatial Fourier coefficients of $x(t)$ to coefficients of $\phi(x)$. Assume we have an autonomous agent whose movements are governed by a dynamic model that is either known a priori or learned from data and is of the form

$$\dot{x} = f(x, u) = g(x) + h(x)u \tag{4}$$

where $x \in R^n$ is the state of the agent and $u \in R^m$ is the control input or "actions" the robot can take. A trajectory $x(t)$ is ergodic with respect to a distribution $\phi(x)$ if, for every neighborhood $\mathcal{N} \subset \mathcal{X}$, the amount of time $x(t)$ spends in $\mathcal{N}$ is proportional to the measure of $\mathcal{N}$ provided by $\phi(x)$. On a long enough time horizon, measuring a perfectly ergodic $x(t)$ gives a complete description of $\phi(x)$. Here, we ask that $x(t)$ be maximally ergodic, by introducing a metric on the distance from ergodicity into the objective function, so that when $x(t)$ captures the statistics of $\phi(x)$ in a specified time horizon T the metric is lower. Ergodicity can be quantified as the sum of the weight square distance between Fourier coefficients of the distribution ϕ_k and the coefficients representing the trajectory c_k as defined below:

$$\varepsilon = \sum_{k_1=0}^{K} ... \sum_{k_n=0}^{K} \Lambda_k |c_k - \phi_k|^2, \tag{5}$$

where there are n dimensions and $K+1$ coefficients along each dimension and the coefficients c_k can be calculated using Eq. 2. The coefficient $\Lambda_k = \frac{1}{(1+||k||^2)^s}$, where $s = \frac{n+1}{2}$, places larger weights on lower frequency information.

We define the task objective as

$$J = q\varepsilon + \int_0^T \frac{1}{2} u(t) R u(t) dt \tag{6}$$

with a cost to minimize the ergodic metric and a cost on the control effort used over time.

Now, using the defined ergodic objective function, we frame the control problem as an MPC problem, following work [3]. The algorithm is described in Algorithm 1.

Algorithm 1 Ergodic Control Algorithm for LfD

Input: initial time t_0, initial state x_0, set of demonstrations $\{d_1, ..., d_m\}$ with positive/negative labels $\{e_1, ..., e_m\}$, final time t_f

Output: ergodic trajectory $x(t) \rightarrow X$

Define: ergodic cost weight Q, highest order of coefficients K, control weight R, search domain bounds $\{L_1, ...L_n\}$, sampling time t_s, desired rate of change α_d, time horizon T

Initialize: nominal control u_{nom}, step $i = 0$

Generate distribution $D(s)$ from set of demonstrations $\{d_1, ..., d_m\}$.

Calculate ϕ_k from distribution $D(s)$

while $t_i < t_f$ **do**

  Compute u_i^* using MPC

  Apply u_i^* for $t \in [t_i, t_i + t_s]$ to get $x \forall t \in [t_i, t_i + t_s]$.

  Define $t_{i+1} = t_i + t_s, x_{i+1} = x(t_{i+1})$

  $i \leftarrow i + 1$

end while

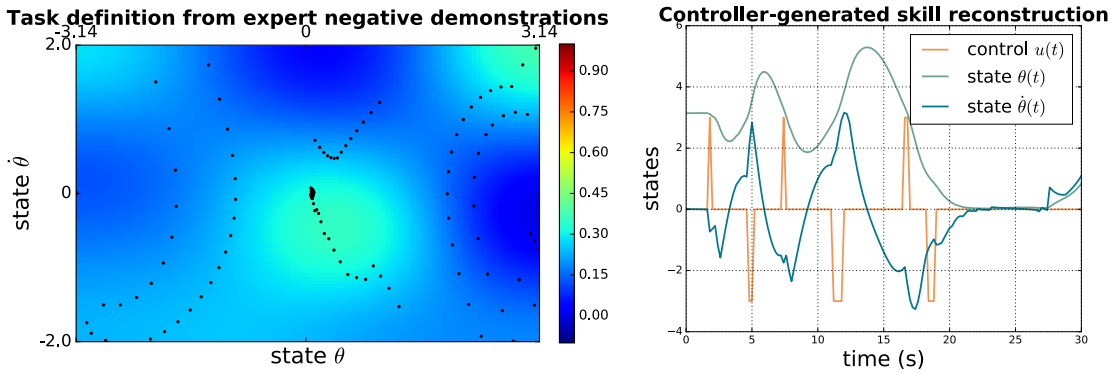

Fig. 4: Example task definition and skill reconstruction learned from 3 *negative* demonstrations. Note that a negative demonstration includes only movements of what *not* to do and—with this low-dimensional task—suffices for learning a sub-optimal, yet successful task definition.

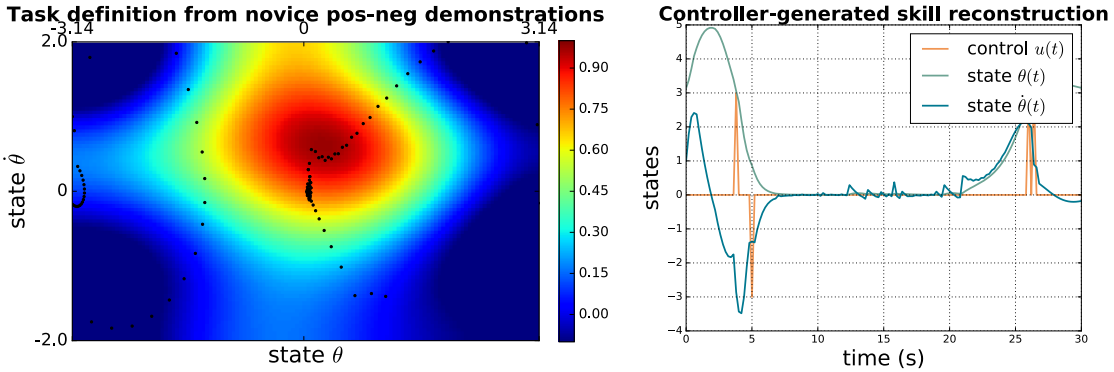

Fig. 5: Example task definition and skill reconstruction learned from positive and negative demonstrations from Subject 6. Note that while the *posonly* controller exhibits performance similar to the original demonstrations, the *posneg* controller significantly outperforms them. Although Subject 6 is still a novice at this task, we can learn a task representation comparable to the one learned from an expert trajectory (see Fig. 2) by soliciting both positive and negative demonstrations.

B. Experimental platforms

We use two simulated experimental platforms and three benchmark tasks for algorithm validation. Similar to [24]–[26], we employ a cart-pole system for initial validation with end users. Inverting and balancing the cart-pole is a great example of a task that is reliably difficult for people, particularly novices, to accomplish. We also test two household tasks on a robot arm. These include reaching with object avoidance (similar to [27], [28]) as well as cleaning or wiping a surface (similar to [29], [30]. These are good examples of real-world assistive tasks that encounter high variability in task execution during demonstrations. The platforms are described in more detail below.

Cart-pole System. A simulated cart-pole system with state vector $x = [\theta, \dot{\theta}, x_c, \dot{x_c}]$ and input $\ddot{x_c}$ was used in a previous study of 24 participants. Participants were each given 3 sets of 30 30-second attempts to invert the pole from its resting state to the unstable equilibrium . The data from this experiment—details of which can be found in [32] and [33]—are used as the novice task demonstrations in this work.

For cart-pole inversion, a demonstration is defined as successful when during the 30-second attempt the participant reaches a state near the unstable equilibrium, specifically $|\theta| < 0.4$ rad and $|\dot{\theta}| < 0.75$ rad/s. We take the best demonstrations each user provided in set 3 (on average the best set) as positive demonstrations and take unsuccessful demonstrations from set 1 (on average the worst set) as negative demonstrations. Finally, we also test our approach on expert demonstrations—the positive expert demonstrations are generated using an optimal controller, whereas the negative expert demonstrations are generated by one of the authors. The true task definition for cart-pole inversion is defined as a Dirac delta function around $[\theta, \dot{\theta}] = [0, 0]$.

Robot Arm Simulator. We develop a pybullet simulation of the Franka Emika Panda Robot Arm to evaluate ergodic LFD on basic table-top tasks, specifically target reaching and table cleaning. In the simulation, we generate demonstrations for robot motion using a keyboard control interface. The keys control the desired end-effector position of the robot in the $[x, y]$ dimensions at a fixed end-effector height z_d. The demonstrations consist of the resulting executed end-effector trajectories, from which we learn a task distribution. After that task definition is learned, we use ergodic MPC as a motion planner for the end-effector by generating desired end-effector positions $[x, y, z_d]$ over time. For the ergodic

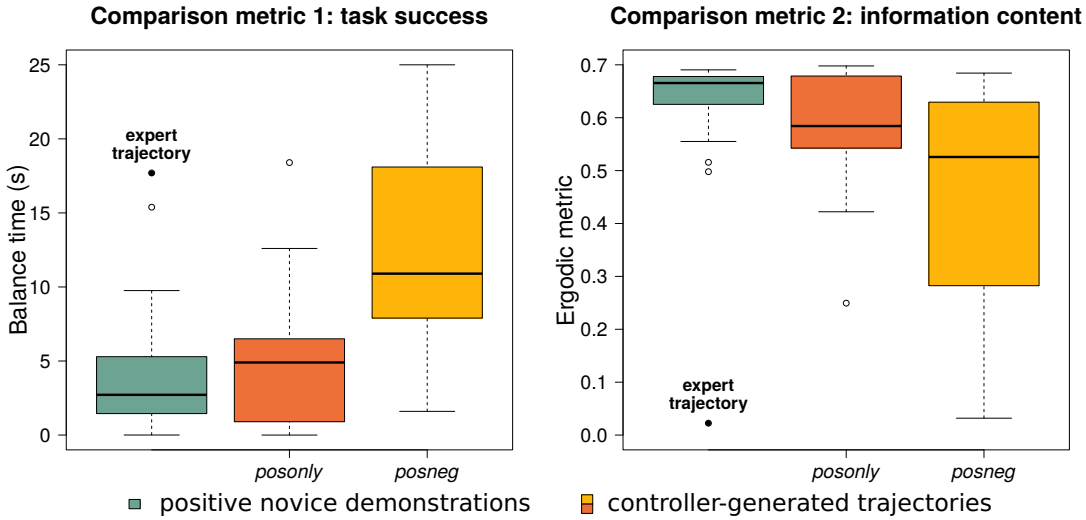

Fig. 6: Comparison of best task executions from 24 novice participants and skill reconstruction based on objectives learned from participant demonstrations, using only positive demonstrations (*posonly*, orange) and from both positive and negative demonstrations (*posneg*, yellow). We employ two performance metrics for comparison: task success time (left) and the ergodic metric [31], which measures information captured about the task through the learned distributions (right). For both metrics, *posonly* skill reconstructions achieve performance comparable to or better than the novice demonstrations. Moreover, *posneg* trajectories significantly outperform the provided novice demonstrations in both metrics: 1 (F=9.07, p=5.7e-10) and 2 (F=1.2, p=3.8e-5)—in fact they provide skill reconstructions comparable to expert task executions.

controller, we model the system as a double-integrator with state $X = [x, y, \dot{x}, \dot{y}]$ and $U = [\ddot{x}, \ddot{y}]$. We use the an inverse kinematics solver to generate joint states corresponding to the target trajectory and employ a low-level joint controller to execute the trajectory.

For the target-reaching task, we define success as reaching a target location without colliding with an obstacle. For the cleaning task, success m is evaluated as a continuous variable based on both workspace coverage and object avoidance. If the controller-generated trajectory comes too near the object, the trial is considered a failure ($m = 0$). Otherwise, the cleaning is assessed by calculating the percentage of the workspace visited by the end-effector, as approximated on a 5×5 grid.

IV. Experimental Results

A. Ergodic LfD enables learning from imperfect demos

We show that ergodic LfD can be used to infer the cart-pole inversion task from imperfect novice demonstrations and that the learned task definitions can be used to recreate the skill on average better than presented during demonstrations. This performance comparison is also visible in Fig. 6, where we see that the controller-generated trajectories using the *posonly* task definitions are on average better than the demonstration trajectories. More specifically, a t-test comparison shows that trajectories generated using ergodic LfD have higher success times (F=0.24, p=0.06) and are more ergodic w.r.t. the true task definition (F=0.68, p=0.002) than the provided demonstrations. This means that when learning from only positive demonstrations, our trained controller will on average slightly outperform the provided task demonstrations. When controller trajectories are more ergodic w.r.t. the learned task distribution than individual demonstration trajectories, it indicates that our learner can actually outperform the demonstrations used to learn the task.

B. Negative demos consistently improve learning

Furthermore, we show that negative demonstrations add more value than numerous positive demonstrations, allowing data-efficient learning. Here, for each of the 24 participants, we learn a task definition from 3 positive and 3 negative demonstrations. We use a controller to recreate the skill with respect to the learned distributions. We compare the controller-generated trajectories with the provided trajectories, again using success time and the ergodic metric. Results of a t-test show that trajectories generated using ergodic LfD have higher success times (F=9.07, p=5.7e-10) and are more ergodic with respect to the true task definition (F=1.2, p=3.8e-5) than the provided demonstrations. They also have higher success times (F=1.4, p=4.9e-7) and are more ergodic (F=0.79, p=0.003) than the trajectories generated using *posonly* demonstrations. Finally, note that the effect sizes are significantly larger than in the earlier comparison.

In the event that an end-user cannot generate any successful demonstrations, we also demonstrate the ability to define a successful task specification from just negative demonstrations. This is visible in Fig. 4, where we note that the skill reconstruction achieves inversion around $t = 20s$. Although impractical for many tasks, this interesting result illustrates that valuable information about a task can be captured in negative demonstrations.

C. Ergodic LfD with negative demos works for variable tasks

Ergodic LfD with *posneg* demonstrations extends to a variety of tasks. It is particularly useful for open-ended

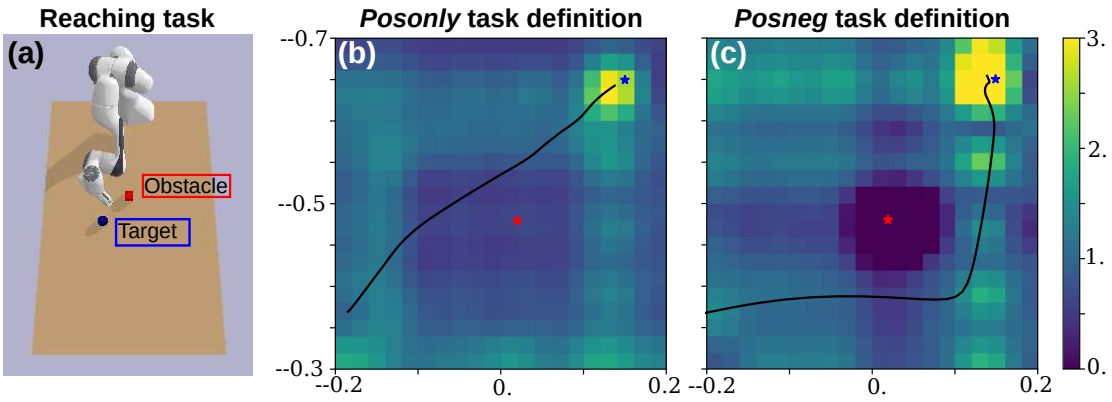

Fig. 7: Target (shown in blue) reaching around an obstacle (shown in red) with a robot arm. (b) Task definition and resulting robot end-effector trajectory generated with an ergodic controller using the positive-only demonstrations. (c) Task definition and resulting robot end-effector trajectory generated with an ergodic controller using the combined positive and negative demonstrations. Negative demonstrations more effectively reflect the region of avoidance, representing what *not* to do.

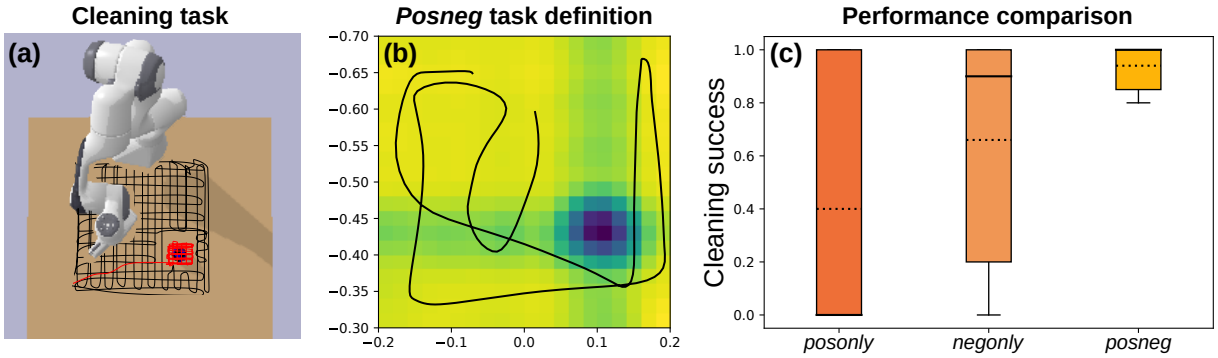

Fig. 8: Cleaning around an object with a robot arm. (a) An example positive demonstration trajectory is shown in black and a negative demonstration is shown in red. (b) Task definition and resulting robot end-effector trajectory generated using the ergodic controller with positive and negative demonstrations. (c) Comparison of success between results from task definitions generated with positive, negative, and *posneg* demonstrations. The black line represents the median result and the dotted line represents the mean. The *posneg* definition results in significantly better performance than either *posonly* or *negonly* definitions, capturing both the desired cleaning and object avoidance goals.

tasks, such as target reaching, where how you get to the destination often does not matter, and for multi-objective tasks, such as target reaching while avoiding an object, where there might be an additional safety constraint on the correct task execution. Positive-only task definitions can be limiting, particularly when trying to represent constraints in the environment.

Fig. 7 shows the results of trying to accomplish a target-reaching task while trying to simultaneously avoid another object in the environment. We present an example trajectory generated from random initial conditions based on a task definition learned from 13 positive demonstrations and combined *posneg* demonstrations (13 positive + 3 negative). Note that with positive demonstrations, the goal location is successfully reflected, but the region of obstacle avoidance is only starting to appear. When we add negative demonstrations (see Fig. 7c), the region of avoidance is more clearly defined.

Similarly, generating task definitions using only positive definitions can be inefficient for multi-objective tasks such as cleaning around an object, as depicted in Fig. 8. Here, task success requires overall coverage to sufficiently clean the workspace and object avoidance to not disturb the object on the table. We compare the results for 10 controller-generated trajectories from random initial states for each type of task definition (*posonly*, *negonly*, and *posneg*)—generated from 5 positive demonstrations, 2 negative demonstrations and 3+2 combined *posneg* demonstrations, respectively. As seen in Fig. 8b, the combination of positive and negative demonstrations offers best performance, highlighting the region to avoid while still representing the rest of the cleaning task. The *posonly* controller-generated trajectories result in many outright failures, where the robot arm collides with the object. However, when it successfully avoids the object, the cleaning results in a 100% workspace coverage. The negative-only definition result in very few failures due to object collision, but the overall workspace coverage is low. The *posneg* definition significantly outperforms both other definitions, resulting in no failures and a median 100% success rate.

V. Conclusions & Discussion

This paper introduces ergodic LfD for learning from novice robot users and illustrates the value of negative demonstrations—reflecting what *not* to do—in imitation learning. While positive-only demonstrations can result in

successful skill reproduction, the combination of positive and negative demonstrations can help to efficiently generate task definitions for difficult tasks. Moreover, we show that ergodic LfD is particularly well suited for multi-objective and open-ended tasks, where either multiple goals are equally important (i.e., moving a cup without spilling) or different motion trajectories can accomplish the same task. As such, there is potential to extend to applications with a focus on learning safety constraints and user preferences, such as in assisted driving—similarly to [34] but without the need for preference querying. Moreover, in future work, the ergodic learning framework could be further automated by implementing feature selections algorithms as in [35], [36], and [37]—the feature variance between negative and positive demonstration could provide insights into features key to a task's success. Overall, the presented results are promising and the proposed algorithmic framework and negative demonstrations have potential to enable demonstration-efficient LfD from imperfect demonstrations for a range of robotic applications.

ACKNOWLEDGMENT

This material is based upon work supported by the NSF under Grant CNS 1837515. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the aforementioned institutions.

REFERENCES

[1] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: learning attractor models for motor behaviors," Neural Computation, vol. 25, no. 2, pp. 328–373, 2013.

[2] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," ACM, p. 1, 2004.

[3] A. Mavrommati, E. Tzorakoleftherakis, I. Abraham, and T. D. Murphey, "Real-time area coverage and target localization using receding-horizon ergodic exploration," IEEE Trans. Robotics, vol. 34, no. 1, pp. 62–80, 2018.

[4] G. Neu and C. Szepesvári, "Apprenticeship learning using inverse reinforcement learning and gradient methods," in Conf. on Uncertainty in Artificial Intelligence, 2007, pp. 295–302.

[5] S. Levine, Z. Popovic, and V. Koltun, "Nonlinear inverse reinforcement learning with gaussian processes," in Advances in Neural Information Processing Systems, 2011, pp. 19–27.

[6] J. Ho and S. Ermon, "Generative adversarial imitation learning," in Advances in Neural Information Processing Systems. Curran Associates, Inc., 2016, pp. 4565–4573.

[7] S. E. Scott, T. C. Redd, L. Kuznetsov, I. Mezić, and C. K. Jones, "Capturing deviation from ergodicity at different scales," Physica D: Nonlinear Phenomena, vol. 238, no. 16, pp. 1668–1679, 2009.

[8] S. E. Scott, "Different perspectives and formulas for capturing deviation from ergodicity," J. Applied Dynamical Systems, vol. 12, no. 4, pp. 1948–1967, 2013.

[9] G. Mathew and I. Mezić, "Metrics for ergodicity and design of ergodic dynamics for multi-agent systems," Physica D: Nonlinear Phenomena, vol. 240, no. 4, pp. 432–442, 2011.

[10] C.-A. Cheng, X. Yan, N. Wagener, and B. Boots., "Fast policy learning through imitation and reinforcement," in Conf. on Uncertanty in Artificial Intelligence, 2018.

[11] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," in Int. Conf. Robotics and Automation. IEEE, 2009, pp. 763–768.

[12] A. Paraschos, C. Daniel, J. R. Peters, and G. Neumann, "Probabilistic movement primitives," in Advances in Neural Information Processing Systems, 2013, pp. 2616–2624.

[13] T. Kulak, J. Silvério, and S. Calinon, "Fourier movement primitives: an approach for learning rhythmic robot skills from demonstrations," in Robotics: Science and Systems (RSS), 2020.

[14] S. Calinon, "A tutorial on task-parameterized movement learning and retrieval," Intelligent Service Robotics, vol. 9, no. 1, pp. 1–29, 2016.

[15] M. Schneider and W. Ertel, "Robot learning by demonstration with local gaussian process regression," in Int. Conf. Intelligent Robots and Systems. IEEE, 2010, pp. 255–260.

[16] N. Jaquier, D. Ginsbourger, and S. Calinon, "Learning from demonstration with model-based gaussian process," in Conf. Robot Learning, 2020, pp. 247–257.

[17] C. Basu, E. Biyik, Z. He, M. Singhal, and D. Sadigh, "Active learning of reward dynamics from hierarchical queries," in Int. Conf. Intelligent Robots and Systems. IEEE, 2019.

[18] E. Bıyık, M. Palan, N. C. Landolfi, D. P. Losey, and D. Sadigh, "Asking easy questions: A user-friendly approach to active reward learning," in Conf. Robot Learning, 2019, pp. 1177–1190.

[19] C. Basu, M. Singhal, and A. D. Dragan, "Learning from richer human guidance: Augmenting comparison-based learning with feature queries," in Int. Conf. on Human-Robot Interaction (HRI). IEEE, 2018, pp. 132–140.

[20] J. Spencer, S. Choudhury, M. Barnes, M. Schmitte, M. Chiang, P. R. Ramadge, and S. Srinivasa, "Learning from interventions: Human-robot interaction as both explicit and implicit feedback," in Robotics: Science and Systems, 2020.

[21] S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza, "Power to the people: The role of humans in interactive machine learning," AI Magazine, vol. 35, no. 4, pp. 105–120, 2014.

[22] M. Cakmak, C. Chao, and A. L. Thomaz, "Designing interactions for robot active learners," Trans. Autonomous Mental Development, vol. 2, no. 2, pp. 108–118, 2010.

[23] A. Komanduru and J. Honorio, "On the correctness and sample complexity of inverse reinforcement learning," in Advances in Neural Information Processing Systems, 2019, pp. 7112–7121.

[24] F. Torabi, G. Warnell, and P. Stone, "Behavioral cloning from observation," in Int. Joint Conf. on Artificial Intelligence, 2018.

[25] T. Brys, A. Harutyunyan, H. B. Suay, S. Chernova, M. E. Taylor, and A. Nowé, "Reinforcement learning from demonstration through shaping," in Int. Joint Conf. on Artificial Intelligence, 2015.

[26] C. Yang, X. Ma, W. Huang, F. Sun, H. Liu, J. Huang, and C. Gan, "Imitation learning from observations by minimizing inverse dynamics disagreement," in Advances in Neural Information Processing Systems, 2019, pp. 239–249.

[27] M. Palan, N. C. Landolfi, G. Shevchuk, and D. Sadigh, "Learning reward functions by integrating human demonstrations and preferences," in Robotics: Science and Systems, 2019.

[28] E. Bıyık, D. P. Losey, M. Palan, N. C. Landolfi, G. Shevchuk, and D. Sadigh, "Learning reward functions from diverse sources of human feedback: Optimally integrating demonstrations and preferences," preprint arXiv:2006.14091, 2020.

[29] S. Elliott, Z. Xu, and M. Cakmak, "Learning generalizable surface cleaning actions from demonstration," in Int. Sym. Robot and Human Interactive Communication. IEEE, 2017, pp. 993–999.

[30] T. Alizadeh, S. Calinon, and D. G. Caldwell, "Learning from demonstrations with partially observable task parameters," in Int. Conf. Robotics and Automation. IEEE, 2014, pp. 3309–3314.

[31] K. Fitzsimons, A. M. Acosta, J. P. Dewald, and T. D. Murphey, "Ergodicity reveals assistance and learning from physical human-robot interaction," Science Robotics, vol. 4, no. 29, 2019.

[32] A. Kalinowska, K. Fitzsimons, J. Dewald, and T. D. Murphey, "Online user assessment for minimal intervention during task-based robotic assistance," in Robotics: Science and Systems, 2018.

[33] K. Fitzsimons, A. Kalinowska, J. Dewald, and T. D. Murphey, "Task-based hybrid shared control for training through forceful interaction," Int. J. Robotics Reserach, vol. 39, no. 9, pp. 1138–1154, 2020.

[34] D. Sadigh, A. D. Dragan, S. Sastry, and S. A. Seshia, "Active preference-based learning of reward functions." in Robotics: Science and Systems, 2017.

[35] O. Kroemer and G. S. Sukhatme, "Feature selection for learning versatile manipulation skills based on observed and desired trajectories," in Int. Conf. Robotics and Automation. IEEE, 2017, pp. 4713–4720.

[36] L. Pais, K. Umezawa, Y. Nakamura, and A. Billard, "Learning robot skills through motion segmentation and constraints extraction," in HRI Workshop on Collaborative Manipulation, 2013.

[37] S. Niekum, S. Osentoski, G. Konidaris, and A. G. Barto, "Learning and generalization of complex tasks from unstructured demonstrations," in Int. Conf. Intelligent Robots and Systems, 2012, pp. 5239–5246.