2010 Ninth International Conference on Machine Learning and Applications

Heterogeneous Imitation Learning from Demonstrators of Varying Physiology and Skill

Jeff Allen and John Anderson
Department of Computer Science
University of Manitoba, Winnipeg Canada
Email: jallen,andersj@cs.umanitoba.ca

***Abstract*—Imitation learning enables a learner to improve its abilities by observing others. Most robotic imitation learning systems only learn from demonstrators that are homogeneous physiologically (i.e. the same size and mode of locomotion) and in terms of skill level. To successfully learn from physically heterogeneous robots that may also vary in ability, the imitator must be able to abstract behaviours it observes and approximate them with its own actions, which may be very different than those of the demonstrator. This paper describes an approach to imitation learning from heterogeneous demonstrators, using global vision for observations. It supports learning from physiologically different demonstrators (wheeled and legged, of various sizes), and self-adapts to demonstrators with varying levels of skill. The latter allows a bias toward demonstrators that are successful in the domain, but also allows different parts of a task to be learned from different individuals (that is, worthwhile parts of a task can still be learned from a poorly-performing demonstrator). We assume the imitator has no initial knowledge of the observable effects of its own actions, and train a set of Hidden Markov Models to map observations to actions and create an understanding of the imitator's own abilities. We then use a combination of tracking sequences of primitives and predicting future primitives from existing combinations using forward models to learn abstract behaviours from the demonstrations of others. This approach is evaluated using a group of heterogeneous robots that have been previously used in RoboCup soccer competitions.**

I. Introduction

Imitation learning - the ability to observe demonstrations of behaviour and reproduce functionally equivalent behaviour with ones own abilities - is a powerful mechanism for improving the abilities of an intelligent agent. Evidence of learning from the demonstrations of others can be seen in primates, birds, and humans [1], [2], [3]. From an AI perspective, this is attractive because of its potential for dealing with the general problem of knowledge acquisition: instead of programming a robot for each individual task, robots should ultimately be able to gather information from human demonstrations [4], [5], [6], or from one another [7], [6], [8] with the result that the robot's performance at that task improves over time. Additionally, demonstrations do not have to be active teaching exercises: the imitator can simply observe a demonstrator with no communication necessary.

To make imitation learning useful, an agent must first have an understanding of its own primitive motor skills, observe demonstrations and their outcomes, and ultimately interpret these within the context of its own primitives. In doing so, the agent develops new motor skills by creating hierarchical combinations of primitives [2], providing a deeper understanding of the imitated behaviour. In any real world setting, this will be complicated by the fact that multiple demonstrations will likely be performed by different agents. Arguably this *should* be the case, since seeing the full range of ways in which a task could be accomplished is faster than the learner discovering these itself, and different agents will likely perform a task in different ways.

Humans naturally deal with heterogeneous demonstrators: if a child's first exposure to the game of frisbee is through observing a dog catching a frisbee in its mouth, when the frisbee is thrown to the child they will likely attempt to catch it in their hand instead. This way they learn the task using the skills that are natural and available to them, even if the demonstration displayed a different set of skills. Robots have been developed for many purposes, and consequently differ in size, control programs, sensors and effectors. In order to increase the performance of a learner and allow it to learn from whatever demonstrators happen to be available (ultimately, a mixture of humans and other robots), overcoming differences in physiology is absolutely necessary [9].

In this paper, we present a framework for imitation through global vision, which models multiple demonstrators by approximating the visual outcomes of their actions with those available to the imitator, with no prior knowledge of demonstrators' abilities or physiology. This framework is able to learn from a range of heterogeneous demonstrators (different physiologies, modes of locomotion, sizes, and behavioural control systems), as well as a different range of domain-specific skills. Individually modelling its teachers enables the robot to be adaptable to heterogeneous demonstrators as well as a range of skill levels. This allows the robot to approximate differences in physiology by actions suited to its own abilities, and to leverage the power of heterogeneous demonstrators to learn portions of a task from one demonstrator that are difficult to approximate from others. It similarly allows an agent to be selective in learning from those who demonstrate better skills in the domain at hand (yet still learn useful portions of a task even from agents that that are not skilled).

The experimental domain we use to ground this work is robotic soccer. In our evaluation, an imitating robot learns to shoot the soccer ball into an open goal, from a range of demonstrators that differ in size as well as physiology (humanoid vs. wheeled).

978-0-7695-4300-0/10 $26.00 © 2010 IEEE
DOI 10.1109/ICMLA.2010.23

IEEE computer society

II. Related Work

A number of prior approaches to imitation learning have influenced this work. Demiris and Hayes [1] developed a computational model based on the phenomenon of *body babbling*, where babies practice movement through self-generated activity [10]. Demiris and Hayes [1] devised their system using *forward models* to predict the outcomes of the imitator's behaviours, in order to find the best match to an observed demonstrator's behaviour. A forward model takes as input the state of the environment and a control command that is to be applied. Demiris and Hayes [1] use one forward model for each behaviour, which is then refined based on how accurately the forward model predicts the behaviour's outcome. By using many of these forward models, Demiris and Hayes construct a repertoire of behaviours with predictive capabilities. In contrast, the forward models in our framework model the repertoire of individual demonstrators (instead of having an individual forward model for each behaviour), and contain individual behaviours learned from specific demonstrators within them (the behaviours can still predict their effects on the environment, but these effects are not refined during execution). This provides the imitator with a model that can make predictions about what behaviours a specific demonstrator might use at a given time.

Prior work in imitation learning has often used a series of demonstrations from demonstrators that are similar in skill level and physiologies [11], [5]. The approach presented in this paper is designed from the bottom up to learn from multiple demonstrators that vary physically, as well as in underlying control programs and skill levels.

Some recent work in humanoid robots imitating humans has used many demonstrations, but not necessarily different demonstrators, and very few have modeled each demonstrator separately. Those that do employ different demonstrators, such as [11], often have demonstrators of similar skills and physiologies (in this work all humans performing simple drawing tasks) that also manipulate their environment using the same parts of their physiology as the imitator (in this case the imitator was a humanoid robot learning how to draw letters, the demonstrators and imitators used the same hands to draw). Inamura et al. [12], [13] use HMMs in their mimesis architecture for imitation learning. They trained a humanoid robot to learn motions from human demonstrators, though they did not separately model or rank demonstrator skills relative to each other like we do in our work. They also only have humanoid demonstrators, unlike our work that focuses on multiple heterogeneous demonstrators.

Nicolescu and Matarić [5] motivate the desire to have robots with the ability to generalize over multiple teaching experiences. They explain that the quality of a teacher's demonstration and particularities of the environment can prevent the imitator from learning from a single trial. They also note that multiple trials help to identify important parts of a task, but point out that repeated observations of irrelevant steps can cause the imitator to learn undesirable behaviours.

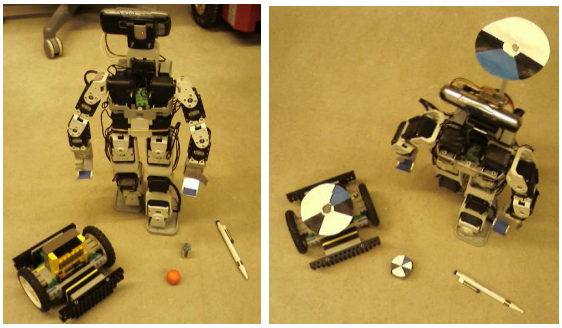

Fig. 1. Two views of the heterogeneous robots used in this work (a ballpoint pen is used to give a rough illustration of scale). The right side of the image shows the robots with visual markers in place to allow motion to be tracked by a global vision system.

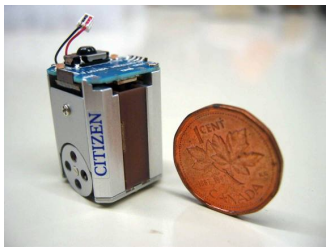

Fig. 2. Closer view of the Citizen Eco-Be Microrobot (v.1).

They do not implement any method of modeling individual demonstrators, or try to evaluate demonstrator skill levels as our work does.

III. Methodology

The robots used in this work are shown in Fig. 1. The robot imitator, a two-wheeled differential-drive robot (built from a Lego Mindstorms kit, and previously used by us in the RoboCup Small-Size league), is on the far left. One of the three robot types used for demonstrators is physically identical (i.e. homogeneous) to the imitator, in order to provide a baseline to compare how well the imitator learns from heterogeneous demonstrators. Two demonstrators that are heterogeneous along different dimensions are also employed. The first is a humanoid robot based on a Bioloid kit, using a mobile phone for vision and processing [14]. The choice of a humanoid was made because it provides an extremely different physiology from the imitator in terms of how motions made by the robot appear visually. The third demonstrator type is a two-wheeled Citizen Eco-Be robot (version I, close-up in Fig. 2), which is about $1/10$ the size of the imitator. This was chosen because the large size difference and difficulty in moving a ball due to light weight makes for a different dimension of heterogeneity.

The imitation learning robot observes one demonstrator at a time, with the demonstrated task being that of shooting a ball into an empty goal, similar to a penalty kick in soccer. This task should allow for enough variation between approaches for

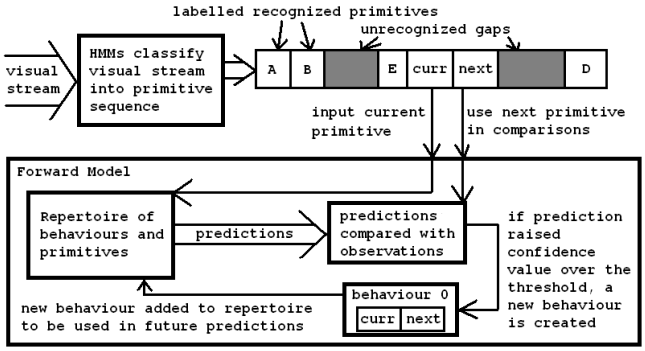

Fig. 3. Imitation Learning Architecture

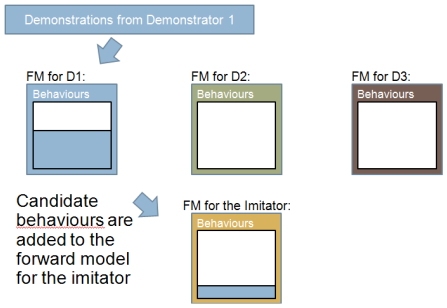

Fig. 4. Demonstrations from each demonstrator are used to train a forward model representing that demonstrator (Demonstrator 1, here). Frequently occurring behaviours in each session are are moved to the forward model representing the imitator as potential behaviours to use in its own activities.

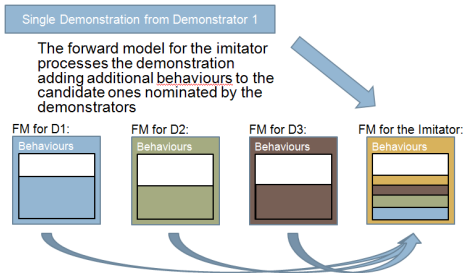

Fig. 5. All demonstrations are passed to the demonstrator models to elicit any further candidate behaviour nominations.

both different skill levels and different physiologies to have an impact. All knowledge of the task to be learned is gained by observing the demonstrators: no communication between the imitator and its demonstrators is allowed (or necessary).

Whether a robot is learning from imitation or not, it must begin with a set of motion primitives that it can use to accomplish actions. In our implementation we have defined these as the atomic motor commands available to the wheeled imitator as (*forward, backward, left, right* and *stop*). In our work, prior to any imitation learning the imitator collects visual data of the outcomes of its own primitive actions using the Ergo vision system [15], to create a basic understanding of what the imitator itself can do. These visual data are used to train a set of Hidden Markov Models (HMMs) [16], which can be used to match activity it views later to actions in the agent's repertoire.

In our approach to imitation learning, the data recorded in a demonstration (and observed during a trial of the imitator) are the x and y field coordinates of the demonstrator/imitator and the ball, as well as the orientations of the demonstrator/imitator. This data is sufficient for the imitator to learn the chosen task from the collection of demonstrators. During each observed demonstration, the imitator uses its knowledge of the visual effects of its own actions (i.e. the mapping represented by HMMs) to convert the visual stream of a demonstration into a sequence of primitive symbols (Fig. 3, top). This matching process is described in [17], and will result in some visual segments that precisely match an imitator's action, others where an action is a close approximation, and others where there will be no match at all (gaps). To attempt to learn from portions of a demonstration where a match is poor or no match at all is possible, the imitator must construct a more meaningful abstraction of the demonstration, using *behaviours*. An implementation-level description of behaviour creation and maintenance requires an understanding of all elements of this approach, and so the equations involved are presented following an abstract description.

Behaviours are learned by combining primitives to produce more complex actions based on observations [18], [3], [5]. In our implementation, a new behaviour is created from a combination of two primitives or existing behaviours when the frequency of the two occurring in sequence surpasses a threshold. For example, suppose the primitive *forward* is recognized in demonstrations, followed by the primitive *left* often enough that the frequency of their sequential occurrence surpasses the threshold. A *forward-left* behaviour is created, made from the primitive sequence *forward* followed by *left*. To keep the number of behaviours learned reasonable, each behaviour has a *permanency* attribute, which is used in conjunction with predictive forward models (described below). As the ongoing actions of a demonstrator are observed, the primitive or behaviour deemed most likely to occur next is predicted, and confirmed through future observations (which may involve a long sequence of primitives to be matched in the case of complex behaviours). A behaviour's permanency is increased if the behaviour is observed after being predicted (i.e. it is useful for modeling behaviour), and slowly decays over time otherwise, to the point where the behaviour is eventually deleted. If the behaviour is predicted and then observed frequently enough, the decay rate will slow, and if the permanency attribute surpasses a threshold, the behaviour will be marked as undeletable.

Behaviours are built and stored using a type of *forward model* (Fig. 3, bottom) which represents frequencies of primitives and behaviours occurring in sequence, and are used to explain and predict the behavior of demonstrators in terms of

the imitator's repertoire. In our approach, a unique forward model is created for each demonstrator to which the imitator is exposed, and begin with only the imitator's primitives. There is an additional forward model for the imitator itself, used to model how the given task should be performed once imitation learning is complete. Training begins by viewing demonstrations for each demonstrator in turn, training only the forward model for that demonstrator: Behaviours are proposed, promoted, and removed through decay as described above. Throughout the training of the demonstrator forward models, frequently occurring behaviours are passed on to the forward model representing the imitator, as suggestions for controlling its own actions (Fig. 4). Following this, each forward model representing a demonstrator is then used to process each demonstration from all demonstrators (Fig. 5). This step allows behaviours in one demonstrator model that may not have been the most frequently used, to be further stimulated by the demonstrations of others and passed along to the imitator forward model. That is, a particular movement combination may be useful but not be the best approach for demonstrator X, but might improve on some part of the technique demonstrated by demonstrator Y. This allows demonstrator X to make a partial contribution even if the technique ultimately followed by the imitator more closely resembles that of Y (for example, because of physiology differences). Finally, the imitator does the processing of all demonstrations using the candidate behaviours added by the forward models for the demonstrators, allowing the imitator to keep some demonstrator behaviours and discard others, while also learning new behaviours of its own.

To model the relative skill levels of the demonstrators in our system, each of the demonstrator forward models maintain a demonstrator-specific learning rate: the *learning preference* (LP). A higher LP indicates that a demonstrator is more skilled than its peers, so behaviours should be learned from it at a faster rate. The LP is used as a weight when updating the frequency of two behaviours or primitives occurring in sequence. The LP of a demonstrator begins at the half way point between the minimum (0) and maximum (1) values. When updating the frequencies (*freq*) of sequentially occurring behaviours (equation 1), a minimum increase in frequency (*minFreq* - 0.05 in our implementation) is preserved, to ensure that a forward model for a demonstrator that has an LP of 0 does not stagnate. The forward model for a given demonstrator would still update frequencies, albeit more slowly than if its LP were above 0. Equation 2 shows the decay step, which happens every time a prediction is made, and is how the permanency of all behaviors is slowly decreased. The *decay rate* is equal to $1 - LP$ and the *decayStep* is a constant (0.007 was used in our experiments). To overcome this constant decay, the permanency of a behaviour is increased when it is successfully predicted. The increase in permanency is given in Equation 3, which shows that a correctly predicted behaviour has its permanency increased by a constant *permUpdate* (0.09 in our experiments).

$$freq = freq + minFreq + minFreq \times LP \quad (1)$$

$$perm = perm - decayRate \times decayStep \quad (2)$$

$$perm = perm + permUpdate \quad (3)$$

$$LP = LP \pm lpShapeAmount \quad (4)$$

The LP of a demonstrator is increased if one of its behaviours results in the demonstrator (ordered from highest LP increase to lowest): scoring a goal, moving the ball closer to the goal, or moving closer to the ball. The LP of a demonstrator is decreased if the opposite of these criteria results from one of the demonstrator's behaviours. Equation 4 shows the update step, where *lpShapeAmount* is either a constant (0.001) if the LP is adjusted by the non-criteria factors, or plus or minus 0.01 for a behaviour that results in scoring a correct/incorrect goal, 0.005 for moving the ball closer to the goal, or 0.002 for moving the robot closer to the ball. These criteria are obviously domain-specific, and are used to shape the learning (a technique that has been shown to be effective in other domains [19]) in our system to speed up the imitator's learning. Though this may seem like pure reinforcement learning, these criteria do not directly influence which behaviours are saved, and which behaviours are deleted. The criteria merely influence the LP of a demonstrator, affecting how much the imitator will learn from that particular demonstrator. Dependence on these criteria was minimized so that future work (such as learning the criteria from demonstrators) can remove them entirely.

When the learning process is complete, the imitator is left with a final forward model that it can use as a basis for performing the tasks it has learned from the demonstrators.

IV. Experimental Results

To evaluate this approach in a heterogeneous setting, we employed the robots previously shown in Fig. 1 to gather demonstrations. Each of the robots used in these experiments was controlled using its own behaviour-based control system that was developed for robotic soccer competitions, and all would be considered expert demonstrations. The Bioloid and Lego Mindstorms robots were demonstrated on a 1020 x 810 cm field, while the Citizen was demonstrated on a 56 x 34.5 cm field (the small size of this robot made for significant battery power issues given the distances covered on the large size field). The ball used by the Bioloid and Lego Mindstorms robots was 10 centimeters in diameter, while a smaller (2.5 cm) ball was needed for the Citizen robot.

We limited the positions to the two field configurations shown in Fig. 6. In the configuration on the left, the demonstrator is positioned for a direct approach to the ball. As a more challenging scenario, we also used a more degenerate configuration (on the left).

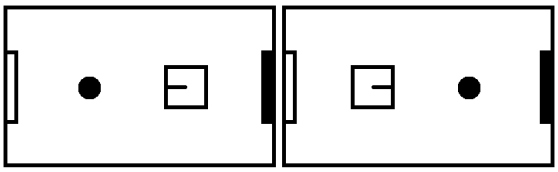

Fig. 6. Field configurations. The demonstrator is represented by a square with an orientation marker. The target goal is indicated by a black rectangle.

Demonstrator	Goals Scored	Wrong Goals Scored
RC2004	27	4
Citizen	15	3
Bioloid	12	1

TABLE I
DEMONSTRATOR PERFORMANCE, GOALS AND WRONG GOALS.

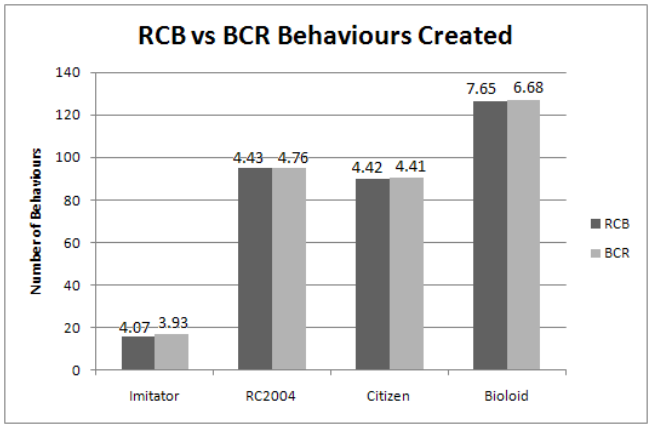

Fig. 7. The number of behaviours created, comparing RCB and BCR demonstrator orderings. Corresponding standard deviations are given at the top of each bar.

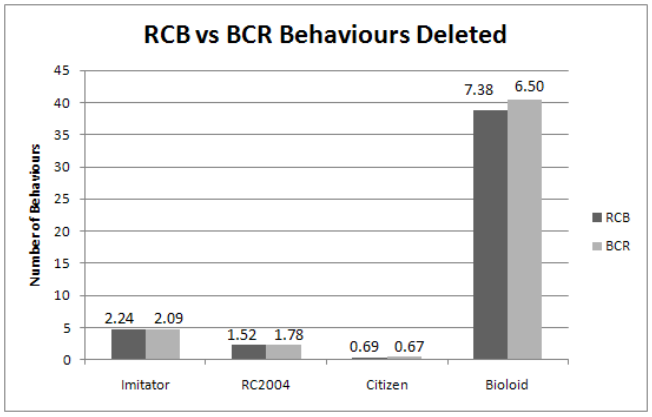

Fig. 8. The number of behaviours deleted, comparing RCB and BCR demonstrator orderings. Corresponding standard deviations are given at the top of each bar.

The individual demonstrators were recorded by the Ergo global vision system [15] while they performed 25 goal kicks for each of the two field configurations. The global vision system continually captures the x and y motion and orientation of the demonstrating robot and the ball. The demonstrations were filtered manually for simple vision problems such as when the vision server was unable to track the robot, or when the robot broke down (falls/loses power). The individual demonstrations were considered complete when the ball or robot left the field. A demonstration could result in a goal on the opposing net (*goal*), a goal on the robot's own net (*wrong goal*), or no goal at all.

One learning trial consists of each forward model representing a given demonstrator training on the full set of kick demonstrations for that particular demonstrator, presented in random order. Once the forward models representing each demonstrator are trained, the forward model representing the imitator begins training. At this point all the forward models for the demonstrators have been trained for their own data, and have provided the forward model representing the imitator with candidate behaviours. The forward model for the imitator then processes all the demonstrations for each of the two field configurations (a total of 150 attempted goal kicks) in random order. All of the forward models for each demonstrator predict and update their models at this time, one step ahead of the forward model for the imitator. This is done to allow each forward model a chance to nominate additional candidate behaviours relevant to the current demonstration instance, to the forward model for the imitator.

The total number of goals and wrong goals each demonstrator scored during all 50 of their individual demonstrations is given in Table I.

To determine if the order in which an imitator is exposed to the various demonstrators had any impact on its learning, we ordered demonstrators in two ways. The first is in order of homogeneity to the imitator. In this ordering, the Mindstorms robot demonstrator (labeled *RC2004* here because its expert-level control code was from our small-sized team at RoboCup-2004) is first, then the Citizen demonstrator (which is much smaller than the imitator, but still a differential-drive robot), and finally the Bioloid demonstrator. The shorthand we have adopted for this ordering is *RCB*. The second ordering is the reverse of the first, that is, in order of greatest heterogeneity to the imitator. The second ordering is thus Bioloid, Citizen, RC2004, or *BCR* for short.

For each of the two orderings, we ran 100 trials. The results of the forward model training processes using the RCB and BCR demonstrator orderings are presented here. All the following data has been averaged over 100 trials.

Figs. 7 and 8 show results for the number of behaviours created and deleted for each of the forward models representing the given demonstrators, with the two orderings for comparison purposes and standard deviations given above each bar. It can be seen that the RCB and BCR demonstration orderings do not affect the number of behaviours created or deleted from any of the forward models. The forward models

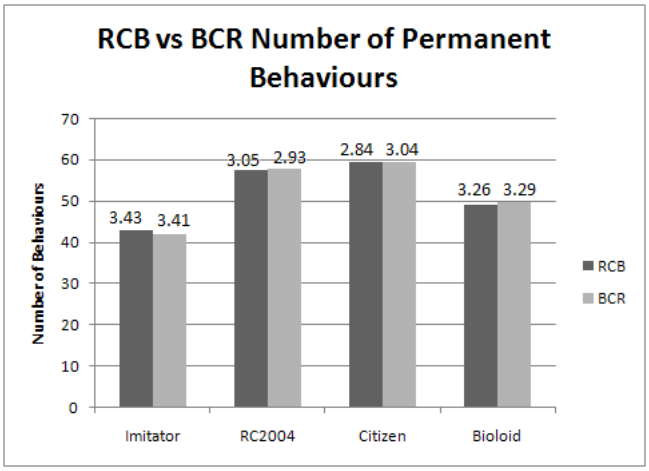

Fig. 9. The number of permanent behaviours in each forward model, comparing RCB and BCR demonstrator orderings. Corresponding standard deviations are given at the top of each bar.

representing the Bioloid demonstrator can be seen to create many more behaviours than the other forward models (and have a higher standard deviation), but they also end up deleting many more than the others. The vast difference in physiology from the other two-wheeled robots cause the forward models representing the humanoid to build many behaviours in an attempt to match the visual outcome of the Bioloid's demonstrations. When trying to use those behaviours to predict the outcome of the other two-wheeled robot demonstrators, they do not match frequently enough (i.e. they are not a useful basis for imitation), and are eventually deleted as a result.

In Fig. 9, the number of permanent behaviours for each of the forward models are shown along with standard deviations above each bar, grouped by RCB and BCR to see any effect on demonstrator orderings. It can be seen that the orderings do not affect the number of behaviours made permanent to any of the forward models, indicating that ordering does not affect the number of useful behaviours acquired by the forward models representing the demonstrators, or the imitator itself. Even though the Bioloid has a very different physiology, the forward models representing its actions still learn a relatively similar number of behaviours as the other two forward models for the other demonstrators. The forward models representing the imitator have fewer permanent behaviours, partly because the forward model for an imitator filters the candidate behaviours given to it by the forward models representing the demonstrators, but it could also be due to the fact that the imitator is only exposed to each set of demonstrations once, while the other forward models see all demonstrations once, but the demonstrations for their particular demonstrator twice.

To evaluate the performance of the imitators trained using this approach, we selected two imitators from the learning trials evaluated in this section at random (one from the RCB training order, and one from the BCR order). We used the forward models to control the Lego Mindstorms robots and

Demonstrator Ordering	Goals Scored	Wrong Goals Scored
RCB	11	9
BCR	7	13

TABLE II
GOALS AND WRONG GOALS SCORED BY IMITATORS TRAINED WITH DIFFERENT DEMONSTRATOR ORDERINGS.

recorded them in exactly the same way that we recorded the demonstrators, for 25 shots on goal in each of the two field configurations (Fig. 6) for a total of 50 trials. Table II shows the results of these penalty kick attempts by the two imitators trained using our framework. We believe the poor performance is related to the rough statistics used when a forward model is controlling the imitator. The LP shaping criteria are used during the control process for selecting a behaviour to execute. The statistical methods used to calculate preconditions were not robust enough given the task at hand, and had small sample sizes to work with. This resulted in the criteria of the robot driving closer to the ball overriding the other LP criteria in most cases. This could be avoided if future work explored methods of gathering more precondition statistics, possibly in simulation for initial training, moving to physical robots later.

A. Learning from Demonstrators of Varying Skill

We also examined the ability of this approach to train an imitator through the observation of demonstrators of varying skill but identical physiology. The physiology chosen was the differential-drive Mindstorms robot. Three demonstrators were employed. The *ExpertDemonstrator* runs international competition-level code previously used at RoboCup, while the *PoorDemonstrator* simply turns until it has a minimum angle threshold to the ball and then moves on that heading. Since it will normally take more than one bump with the robot to get the ball to the goal, the latter approach will cause significant wandering over the field and a greater likelihood of scoring on its own net even from the favourable configuration. Finally, there is also an *AverageDemonstrator*, chosen randomly from the imitators trained in the heterogeneity experiments described above. This was done because their performance fell between the two extremes of the other demonstrators, and to illustrate the potential for generational learning using this approach. The actual performance of these demonstrators (in terms of the number of goals and wrong goals scored by each) is shown in Table III. To avoid any influence of demonstrator ordering on these experiments, during the phase where the forward models representing the demonstrators are trained, each demonstration is chosen randomly.

Figs. 10 and 11 show the number of behaviours created and deleted for the various forward models. The forward models for the ExpertDemonstrator have fewer behaviours created than the others, though they also have far fewer of them deleted. This indicates that the behaviours learned by the forward models for the ExpertDemonstrator are more useful than those learned by the other models. There is not a large difference between the models representing the PoorDemon-

Demonstrator	Goals Scored	Wrong Goals Scored
PoorDemonstrator	13	23
AverageDemonstrator	11	9
ExpertDemonstrator	27	4

TABLE III
THE NUMBER OF GOALS AND WRONG GOALS SCORED FOR EACH DEMONSTRATOR.

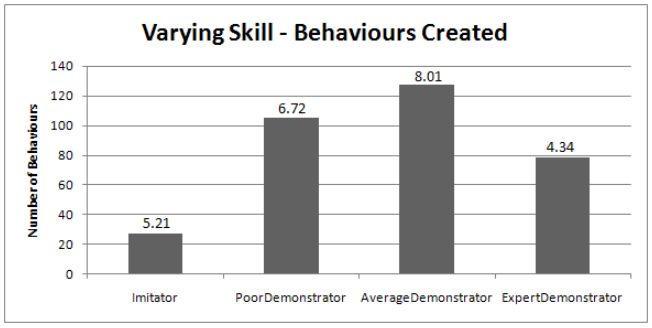

Fig. 10. The number of behaviours created. Corresponding standard deviations are given at the top of each bar.

strator or AverageDemonstrator. We believe this is due to the control system of the imitator (the AverageDemonstrator) relying too heavily on the LP criteria of its behaviours, which cause it to favour driving toward the ball. As mentioned previously, a larger set of training data would aid in proper pruning of behaviours based on preconditions.

Fig. 12 shows that the forward models for the ExpertDemonstrator retain more of the behaviours they create (make them permanent) than the other forward models. This validates our approach to behaviour permanencies that decay over time. The less skilled demonstrators have lower LPs, and therefore higher decay rates. Since the forward models representing the ExpertDemonstrator have a higher LP than the others (shown in Figs. 13:15), the forward models learn behaviours more quickly, and have their behaviours decay more slowly. The number of behaviours retained by each model is thus strongly related to the LP, which was our intention when employing

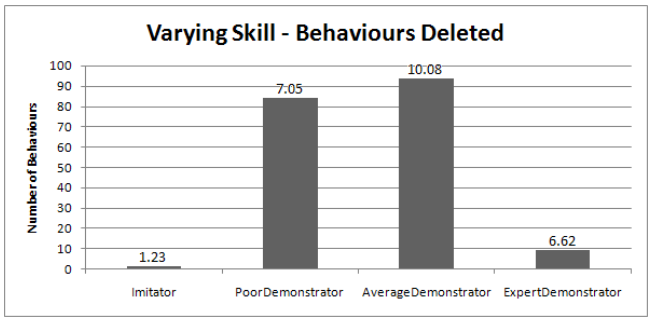

Fig. 11. The number of behaviours deleted. Corresponding standard deviations are given at the top of each bar.

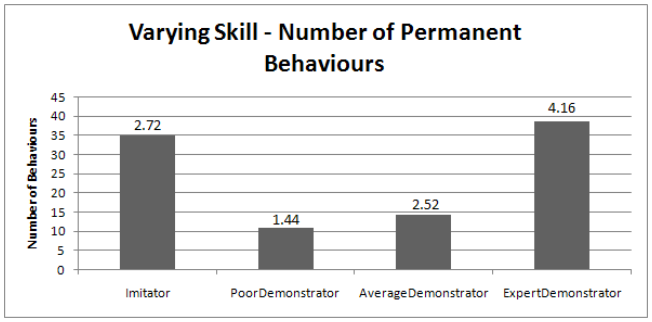

Fig. 12. The number of permanent behaviours. Corresponding standard deviations are given at the top of each bar.

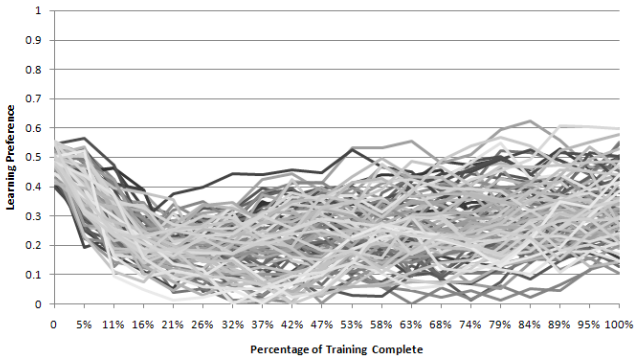

Fig. 13. The change in LP over time for the PoorDemonstrator.

demonstrator specific learning rates. These results show that our imitation learning architecture adaptively weights its learning toward demonstrators that are highly skilled. At the same time, our approach still allows less-preferred demonstrators to supply behaviours that support portions of behavior that preferred demonstrators cannot (for reasons of physiology difference, for example).

The PoorDemonstrator in these trials is recognized as poorly skilled by the imitation learning architecture fairly quickly, as the forward models representing it have their LP decrease below the average LP value (0.5), and then fluctuate around 0.3. The trend is downwards for most of the PoorDemonstrator's LP over time, but trends slightly upward as training progresses. We believe that the few behaviours the PoorDemonstrator acquires later in the training phase aid in generating predictions that match the demonstration, which in turn increases the LP of the PoorDemonstrator.

To evaluate the performance of the imitators trained using

Imitator	**Goals Scored**	**Wrong Goals Scored**
VaryingSkillTrained	11	13

TABLE IV
GOALS AND WRONG GOALS SCORED FOR AN IMITATOR TRAINED BY DEMONSTRATORS OF VARYING SKILL LEVELS.

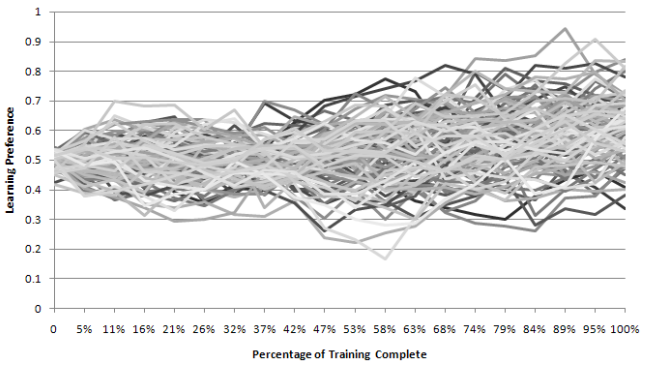

Fig. 14. The change in LP over time for the AverageDemonstrator.

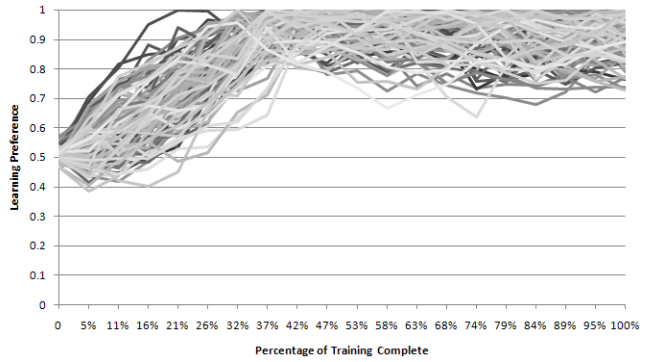

Fig. 15. The change in LP over time for the ExpertDemonstrator.

this approach, we selected an imitator from these trials at random. We used the forward model to control the Lego Mindstorms robot and recorded it in exactly the same way that we recorded the demonstrators, for 25 shots on goal in each of the two field configurations (Fig. 6) for a total of 50 trials. Table IV shows the results of these penalty kick attempts by the imitator trained from demonstrators of varying skill. Though somewhat disappointing in an absolute sense, the performance of a robot using the imitator as a control program still showed that the imitator can learn behaviours from demonstrators and perform the same tasks as the demonstrators. Moreover, this imitator achieves roughly the same results as that trained only with expert demonstrators in the previous experiment, despite having average and poor demonstrators working with it.

V. Conclusion

We have presented the results and analysis of the experiments used to evaluate our approach to developing an imitation learning architecture that can learn from multiple demonstrators of varying physiologies and skill levels. The complete set of experiments and all results are found in [17]. The results for the performance of our forward models when used as control systems did not perform as well as the expert demonstrators, but they still were able to control the imitator adequately. The main focus on our research was in developing an imitation learning architecture that could learn from multiple demonstrators of varying physiologies and skill levels. The results in Section IV indicate that the learning architecture we have devised is capable of properly modeling relative demonstrator skill levels and can learn from physiologically distinct demonstrators. A stronger focus on the refinement of behaviour preconditions and control (possibly through simulation) similar to the work of Demiris and Hayes [1] could make our entire system more robust.

References

[1] J. Demiris and G. Hayes, "Imitation as a dual-route process featuring predictive and learning components: A biologically plausible computational model," in *Imitation in Animals and Artifacts*, K. Dautenhahn and C. Nehaniv, Eds. MIT Press, 2002, pp. 327–361.

[2] M. J. Matarić, "Sensory-motor primitives as a basis for imitation: linking perception to action and biology to robotics," in *Imitation in Animals and Artifacts*, K. Dautenhahn and C. Nehaniv, Eds. MIT Press, 2002, pp. 391–422.

[3] A. Billard and M. J. Matarić, "A biologically inspired robotic model for learning by imitation," in *Proceedings, Autonomous Agents 2000*, Barcelona, Spain, June 2000, pp. 373–380.

[4] M. J. Matarić, "Getting humanoids to move and imitate," *IEEE Intelligent Systems*, pp. 18–24, July 2000.

[5] M. Nicolescu and M. J. Matarić, "Natural methods for robot task learning: Instructive demonstration, generalization and practice," in *Proceedings of the 2nd IJCAA*, Melbourne, July 2003, pp. 241–248.

[6] C. Breazeal and B. Scassellati, "Challenges in building robots that imitate people," in *Imitation in Animals and Artifacts*, K. Dautenhahn and C. Nehaniv, Eds. MIT Press, 2002, pp. 363–390.

[7] J. Anderson, B. Tanner, and J. Baltes, "Reinforcement learning from teammates of varying skill in robotic soccer," in *Proceedings of the 2004 FIRA Robot World Congress*. Busan, Korea: FIRA, October 2004.

[8] P. Riley and M. Veloso, "Coaching a simulated soccer team by opponent model recognition," in *Proceedings of the Fifth International Conference on Autonomous Agents*, May 2001, pp. 155–156.

[9] C. L. Nehaniv and K. Dautenhahn, "Of hummingbirds and helicopters: An algebraic framework for interdisciplinary studies of imitation and its applications," in *Interdisciplinary Approaches to Robot Learning*, J. Demiris and A. Birk, Eds. World Scientific Press, 2000.

[10] A. N. Meltzoff and M. K. Moore, "Explaining facial imitation: A theoretical model," in *Early Development and Parenting*. John Wiley and Sons, Ltd., 1997, vol. 6, pp. 179–192.

[11] S. Calinon and A. Billard, "Learning of Gestures by Imitation in a Humanoid Robot," in *Imitation and Social Learning in Robots, Humans and Animals*, K. Dautenhahn and C. N. (Eds), Eds. Cambridge University Press, 2007, pp. 153–177.

[12] T. Inamura, I. Toshima, and Y. Nakamura, "Acquiring motion elements for bidirectional computation of motion recognition and generation," in *Proceedings of the International Symposium On Experimental Robotics (ISER)*, Sant'Angelo d'Ischia, Italy, July 2002, pp. 357–366.

[13] T. Inamura, Y. Nakamura, I. Toshima, and H. Tanie, "Embodied symbol emergence based on mimesis theory," *International Journal of Robotics Research*, vol. 23, no. 4, pp. 363–377, 2004.

[14] J. Baltes and J. Anderson, "Complex AI on small embedded systems: Humanoid robotics using mobile phones," in *Proceedings of the AAAI 2010 Spring Symposium on Embedded Reasoning: Intelligence in Embedded Systems*, Stanford, CA, March 2010.

[15] ——, "Intelligent global vision for teams of mobile robots," in *Mobile Robots: Perception & Navigation*, S. Kolski, Ed. Vienna: Advanced Robotic Systems International, 2007, ch. 9, pp. 165–186.

[16] L. Rabiner and B. Juang, "An introduction to Hidden Markov Models," *IEEE ASSP Magazine*, pp. 4–16, 1986.

[17] J. Allen, "Imitation learning from multiple demonstrators using global vision," Master's thesis, Department of Computer Science, University of Manitoba, Winnipeg, Canada, August 2009.

[18] C. A. Calderon and H. Hu, "Goal and actions: Learning by imitation," in *Proceedings of the AISB 03 International Symposium on Imitation in Animals and Artifacts*, Aberystwyth, Wales, 2003, pp. 179–182.

[19] M. J. Matarić, "Reinforcement learning in the multi-robot domain," *Autonomous Robots*, vol. 4, no. 1, pp. 73–83, 1997.