Towards Transferrable Personalized Student Models in Educational Games

Samuel Spaulding, Jocelyn Shen, Haewon Park and Cynthia Breazeal
MIT Media Lab
Cambridge, Massachusetts

ABSTRACT

To help facilitate play and learning, game-based educational activities often feature a computational agent as a co-player. Personalizing this agent's behavior to the student player is an active area of research, and prior work has demonstrated the benefits of personalized educational interaction across a variety of domains. A critical research challenge for personalized educational agents is real-time student modeling. Most student models are designed for and trained on only a single task, which limits the variety, flexibility, and efficiency of student player model learning.

In this paper we present a research project applying transfer learning methods to student player models over different educational tasks, studying the effects of an algorithmic "multi-task personalization" approach on the accuracy and data efficiency of student model learning. We describe a unified robotic game system for studying multi-task personalization over two different educational games, each emphasizing early language and literacy skills such as rhyming and spelling. We present a flexible Gaussian Process-based approach for rapidly learning student models from interactive play in each game, and a method for transferring each game's learned student model to the other via a novel instance-weighting protocol based on task similarity. We present results from a simulation-based investigation of the impact of multi-task personalization, establishing the core viability and benefits of transferrable student models and outlining new questions for future in-person research.

KEYWORDS

Transfer Learning; User Modeling; Human-Robot Interaction; Educational Games

ACM Reference Format:
Samuel Spaulding, Jocelyn Shen, Haewon Park and Cynthia Breazeal. 2021. Towards Transferrable Personalized Student Models in Educational Games. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), Online, May 3–7, 2021*, IFAAMAS, 9 pages.

1 INTRODUCTION

Early language and literacy skills are important foundations for learning, representative of the multifaceted human abilities of increasing importance in the 21st century. So-called 'Serious Games' are used in many scenarios that require interactive practice or training, and these games often feature computational agents that help facilitate and personalize the gameplay to make practice more efficient and effective [18]. Some games feature embodied "social" robots that co-occupy physical game space with human players. Compared to virtual counterparts, physically embodied robots have been shown to improve learning and engagement across a wide variety of tasks [6]. These benefits broadly overlap with metrics of 'resonance' [15] in learning games, suggesting that combining embodied human-robot interaction (HRI) with the context of a serious game may make for more effective learning interactions than either individual paradigm.

Following these insights, researchers are studying how social robots can be used to promote childhood education by engaging with, understanding, and adapting to students to provide scalable, personalized, interactive digital learning systems. Prior research has shown that, like physical embodiment, modeling a student player's knowledge and adapting the game to their level leads to greater learning and engagement [21]. Thus far, these modeling efforts have focused on implementations within individual games; student models remain largely ad-hoc, and almost always focused on a single educational activity, i.e., a student model or policy trained separately per game 'task'.

Research on collaborative HRI for physical tasks highlights the effectiveness of multi-task and transfer learning methods [19], [8], yet single-task personalization remains the current standard for adaptive educational games. We hypothesize that designing *transferrable* student player models (i.e., a model designed for and trained on one game task, sharing data and inference with a model designed for a separate game task) can be an effective paradigm for increased data efficiency and improved final quality of learned models, as well as support a greater variety and scope of educational gameplay over long-term interactions. To investigate these research questions, we have developed an autonomous social robot system with an associated suite of interactive educational games designed to help young students practice early language/literacy activities. In each game, the robot and student engage each other in interactive play that is designed to assess, teach, or reinforce a particular early language or literacy skill.

In this paper we present formal models and transfer approaches between two such games: **RhymeRacer**, an interactive digital rhyming game in which players group words with similar rhyme endings, and **WordBuilder**, in which players work together to create words out of letter blocks. Both games are designed to support a human student and a robotic peer as the players, and facilitate adaptive, mixed-initiative interaction between them to help young children learn and practice early alphabetic principles regarding letters, sounds, and spelling. As gameplay unfolds, the robotic player learns a personalized model of students' knowledge from gameplay.

Our contributions in this paper include details of each game and the associated social robot system; formal descriptions of their associated interactive player models and a computational approach for transferring player models between games; and results from a

Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3–7, 2021, Online.

simulation experiment that establish the theoretical potential of transferrable, multi-task student models.

2 RELATED WORK

Our work draws from two distinct but overlapping areas of research: Human-Robot Interaction, specifically that which focuses on educational game-based interaction, and research on Serious Games and AI.

2.1 Social robots as adaptive language learning companions for children

Social robots' ability to interactively engage students has received increasing attention in the past decade [4]. Prior work has shown how social robots can significantly increase children's engagement and language/literacy skills, from vocabulary acquisition [25] to word decoding [28] and complex narrative generation [21]. In many of these projects, robots model students' knowledge and adapt the educational content and robot behaviors to promote learning and engagement. These models can yield actionable insights into student's current state of knowledge, estimates of interpretable parameters like rate of learning, and information about students' learning styles and interaction preferences such as whether a student is motivated by competition or collaboration or how best to encourage students after a setback. Field research studies ([31], [10]), conducted 'in-the-wild' over several weeks at local schools have shown that personalized social robot systems can effectively improve student learning over long-term interactions.

Despite these recent advances, designing human-robot interactions that maintain student engagement over the long-term remains a challenge, in part because the basic interaction structure typically remains fixed over time. The personalized models improve as additional interaction data are incorporated, but because the models are designed for a single interaction task, the student experiences little variety in the main activity over the course of a long-term interaction. For example, students engaging in a vocabulary learning interaction with a robot over several weeks would typically follow the same pattern of hearing a lesson or playing a few rounds of a touchscreen-based game with the robot, with the main difference being new content selected by an increasingly personalized model incorporating the prior week's data. After the first few interactions, children's engagement tends to drop off, a phenomenon well-known among HRI researchers as the "novelty effect" [3].

Long-term interactions are one of the few ways to effectively obtain enough data for deeply personalized models, and variety in interaction activities is crucial to maintaining engagement and mitigating the novelty effect over repeated interactions. If student models were designed to transfer across tasks, long-term interactions would benefit from more consistently high student engagement and larger and more varied player data for model personalization.

2.2 Player modeling in interactive games

Adaptive player modeling is an umbrella term for techniques using player data to make inferences that affect subsequent gameplay [32]. Sometimes called 'Experience Management' [30], adaptive player modeling is the bedrock of research on developing sophisticated interactive agents. Zhu & Ontañón highlight a number of research applications for Experience Management techniques, most relatedly, "interactive learning environments, including intelligent tutoring systems, pedagogical agents, and cognitive science/AI-based learning aids" [34].

Real-world implementations of adaptive player modeling systems face the technical problem of "cold start" learning. Analogous to the difficulty of starting a motor after it has fallen into disuse, "cold start" learning refers to the challenge of training an adaptive player model from real-time gameplay data. Personalized models require gameplay data to learn, but data-poor model instances perform poorly, so players choose not to interact with the system, thereby depriving it of future data from which to learn [17]. Transferable player models could help mitigate this problem by providing an initial baseline of data-driven model performance, derived from data collected during a prior 'source' task.

Recently, research applying multi-task learning to educational games has used data from a group of students to train a predictive model of student performance, treating each question of a game as a separate 'task' to learn [9]. In our work, each task is an entire game (comprised of multiple questions), and the task models are trained and transferred sequentially on personalized data, rather than post-hoc on group data.

Leading researchers of interactive educational games recognize both the challenge and the potential of personalized student player models designed for multi-task transfer. In a recent summary of the state-of-the-art of Learning Analytics, Ryan Baker outlined a series of challenges for the future of the field. The very first challenge on the list is the "learning system wall": the inability of student learning models to transfer outside of the environment in which they are trained [1]. Our work represents one of the first concrete implementations of such a system.

3 TRANSFERRABLE PLAYER MODELS

While personalized social robot systems have been shown to improve student learning over long-term interactions in pre-registered, large-sample trials [31], most such systems are designed around a single task and corresponding student model. Learning is a lifelong, multifaceted process, yet student observations in one task are not used to update models and policies of other relevant tasks. Machine learning systems are said to exhibit 'catastrophic forgetting' when they perform poorly on previously learned tasks after exposure to data from a new task. Yet despite substantial recent progress in meta- and multi-task learning in Deep Reinforcement Learning settings [7] [33], when students in educational interactions switch games (or even tasks within the same game), the underlying models are not designed to 'remember' student data from previous tasks at all!

To overcome this limitation, we propose a "multi-task personalization" transfer learning approach in which students play multiple distinct games, with interaction data and inferred player models transferred across games. We hypothesize three specific benefits from multi-task personalization:

First, by integrating data from multiple activities into each game interaction's unique models, multi-task personalization may lead to

more efficient use of data. Data efficiency is particularly important for applied research in real-world, personalized educational models, as data collection opportunities for novel game designs with real students tend to be scarce, compared to other application domains (e.g. player telemetry from already popular games).

Second, by enabling variety in educational tasks without (catastrophic) loss of personalization, multi-task personalization may help maintain higher levels of student engagement and mitigate the novelty effect over a long-term interaction. Currently, the inability of models to transfer or generalize over different interaction types force researchers to rely on the same interaction (or subtle variants) for several weeks, adding to the challenge of personalized long-term interactions [14].

Finally, designing a multi-task personalization system with multiple distinct tasks may also prove beneficial to educators and domain experts by increasing the variety of multimodal interaction data that can be elicited, educational skills that can be taught, and personalized models that can be learned, leading to a more holistic computational model of student players. Instead of a four-session study to evaluate a student's phonemic rhyme awareness, followed by a separate four-session study to assess student's alphabetic and spelling skills, our system design connects both skills to give a more complete picture of a student's learning progress in shorter time.

3.1 Overview of Approach

Transfer learning [20] is a class of machine learning methods involving a 'source' and 'target' task. Well-known sub-classes of transfer learning problems (e.g., domain adaptation, multitask learning) are defined based on the availability of data in source and target tasks, as well the degree of similarity between source and target task formulations.

In this paper, our source and target tasks are the CognitiveModels learned during each game, which represent estimates of a student's mastery of a literacy skill (e.g. rhyming/spelling). These CognitiveModels take the form of a Gaussian Process (GP), defined over a domain of 74 words called the Curriculum. The set of words in the Curriculum is common to each of game tasks, but the geometry of the 'word space' is unique to each task, formally defined by a covariance kernel that is the primary driver of GP inference. Each task's covariance kernel computes how 'close' a pair of words are to each other, and, therefore, how much an observation of skill mastery of a particular word affects the posterior estimate of skill mastery of an unobserved word.

For example, in RhymeRacer, the primary literacy skill the game is designed to assess and encourage is *rhyming*: an observation of a student correctly identifying a rhyme for the word 'FALL' should increase the model estimate that the student is likely to also be able find a rhyme for the word 'BALL'. This 'closeness' is reflected in the design of the covariance kernel for RhymeRacer (see Sec. 4.1.2). Likewise, in WordBuilder, the primary literacy skill guiding the game design is *spelling*. An observation of a student correctly spelling the word 'SNAKE' should increase the model estimate that the student is likely to be able to spell 'SNAIL'.

Our approach for multi-task model transfer is to instance-weight specific skill demonstrations of words, with the transfer weighting determined by the similarity of that word's use in the source and target tasks. Informally, the covariance between a given word (e.g. 'BALL') and all other words defines *what that word means* within the context of the specific task model. If, under two distinct (source and target) task covariance kernels, 'BALL' has identical covariance to all words in the Curriculum, then functionally, a positive demonstration of 'BALL' under the source task conveys the same information as a positive demonstration under the target task. To compute the transfer weighting of a training instance (e.g. a demonstration of correctly rhyming 'BALL'), we look at the difference in the covariance between source and target tasks for 'BALL' and all other words in the Curriculum. See Sec. 5 for greater detail on the instance-weighting transfer algorithm.

4 PERSONALIZED LITERACY GAME SYSTEM

In this section we describe a playable system composed of two games, called RhymeRacer and WordBuilder, designed for student co-play with a physically embodied robot peer. Both games were developed with the Unity game engine, and receive commands and send input data back to a backend controller via ROS [23]. Both games were designed for children in early stages of literacy learning (approx. age 5-7). In addition to typical playtesting validation, we consulted experts in early childhood learning and children's media throughout the design process for specific content recommendations and to ensure overall fidelity to goals and practices of children's media for language/literacy learning. This paper's primary focus is on transfer learning of each game's CognitiveModel, therefore we focus our description of each game on details necessary to understand the modeling and transfer algorithms, rather than in-depth detail of each game's adaptive behavior.

The games are designed to complement each other, and feature different mechanics, scoring systems, and 'winning conditions'. RhymeRacer was designed to be more competitive, WordBuilder was designed to be collaborative. RhymeRacer's primary educational focus is on phonemic rhyme identification, WordBuilder emphasizes the alphabetic skills of spelling to reinforce the mapping between phonemes and letter combinations.

Within each game, the robotic agent presents itself as a peer and plays the game alongside the student. From student's gameplay data, the system infers a personalized CognitiveModel of the student player (an estimate of their current state of curricular knowledge, based on prior game actions). Both games use the same Curriculum of words, which forms the domain of the CognitiveModel and thus the basis for the transfer learning approach. The Curriculum is a list of 74 words hand-picked by experts in early-childhood education specifically for use in these games. They feature words which are generally phonetically, orthographically, and semantically (e.g. animals, foods, household items) age-appropriate, and a collection of words forming distinct rhyme groups.

4.1 RhymeRacer: A Game for Practicing Phonological Awareness

4.1.1 Gameplay Overview. RhymeRacer is similar to the WordRacer game (introduced in [27] and extended in [26]), a fast-paced, competitive, 2-player game that proceeds through a series of discrete game rounds.

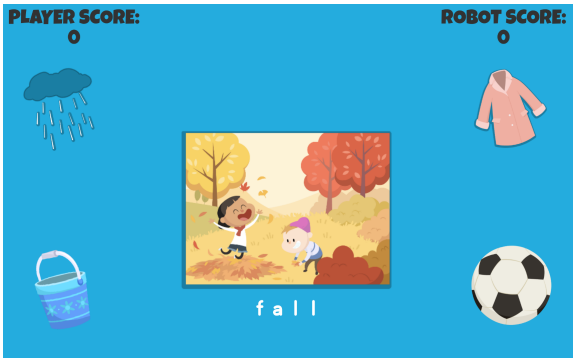

Figure 1: A round of RhymeRacer. FALL is the Target word, Prompt words are RAIN, COAT, PAIL, and BALL.

At the start of each round, the tablet shows a picture of the 'Target' word in the center of the screen (see Figure 1), surrounded by four 'Prompt' word graphics, smaller pictures of other words from the Curriculum, exactly one of which rhymes with the Target word. The tablet also gives a recorded audio prompt, saying "What rhymes with [Target Word]?" as the images are displayed. The first player to correctly tap on the rhyming Prompt word graphic is awarded points, after which the graphics clear and the next round begins.

The robot player is presented to the human player as a co-playing peer, and its outward behavior affirms this framing: the robot player selects Prompt word graphics just as the human player does, gives a mixture of correct and incorrect responses, and responds with appropriate socio-emotional behaviors to in-game events (e.g., acts excited when scoring points, disappointed when incorrect, encouraging when human player scores points).

4.1.2 Cognitive Model. RhymeRacer uses a Gaussian Process regression model in a domain space of words from the Curriculum, essentially identical to the model described in [27]. A Gaussian Process (GP) is a flexible, probabilistic model that is well-suited for regression modeling in data-sparse applications in which domain knowledge can be encoded as a covariance function. Technically, a Gaussian Process is a distribution over possible functions, where the distribution of function evaluations at a finite set of points is jointly Gaussian.

In other words, a GP is a probabilistic inference model that makes Gaussian predictions over a set of output points, based on a set of observed data points $\{x_i, y_i\}$. In the word-space domain, each input data point is a word from the Curriculum and a score from $(-1, 1)$ representing the student's demonstrated level of skill mastery applied to that word during gameplay. For each point in an output set, the GP model computes a posterior mean and posterior variance, $\{\mu_i, \sigma_i\}$, which, in our application, represent the posterior estimate that the student can apply the modeled skill to the output word (e.g. correctly identify the rhyming word for 'FALL') and the uncertainty surrounding that estimate.

The GP posterior is largely driven by the *covariance* function or kernel of the GP, which defines a pairwise distance between domain points, i.e., how much each labelled point from the input set contributes to posterior inference at each other output point. Once the domain and covariance kernel are defined, GP inference is fairly straightforward [24]. The covariance kernel, therefore, defines the 'task' modeled by the GP output predictions.

For RhymeRacer, the domain of the GP model is the set of words from the Curriculum, and the covariance is a modified form of the kernel used in [27]: the normalized cosine distance between any two words' GloVe semantic word vectors [22], plus an additional term that increases two words covariance if they share a rhyme ending (e.g. "-ALL" for FALL and BALL).

$$Cov_{rr}(\{w_i, w_j\}) = \nu[\alpha + \cos(GloVe(w_i), GloVe(w_j)], \quad (1)$$

where $\alpha = 1.0$ iff w_i and w_j share a rhyme ending, and 0 otherwise. ν is a normalization constant.

A GP is a regression model, and can therefore handle a continuous label space, but the design of the RhymeRacer game input gives only a discrete, binary signal: whether the student selected the correct rhyming word or not. To determine the final model observation value (y_i) for a round Target word (x_i), we apply a timing adjustment, $p(t_d)$, to correct for the possibility of guessing. The timing adjustment is applied as a discrete, step-wise penalty of .1 based on the number of seconds it takes to give an answer, i.e, $p(t_d) = 0.1 \cdot t_d$ where t_d is the time of delay in seconds. For example, if a student selects the correct Prompt word for a round within the first second, they receive no penalty, but if they selected the correct Prompt word after 5 seconds, they receive a penalty of $p(t_d) = 0.5$.

4.2 WordBuilder: A Constructionist Game for Practicing Early Alphabetic Skills

WordBuilder is a brand-new game developed to complement RhymeRacer. The two games use a similar design process and share some game assets to maintain a consistent visual style, most notably the Target word graphics that depict the words from the Curriculum. To facilitate model transfer, the overall technical architecture and, notably, the word-based Gaussian Process modeling paradigm are also common to both games. However, the individual implementations of each model, like the tasks each game is designed to assess and reinforce, differ considerably.

WordBuilder serves as a counterpart to RhymeRacer in two main ways: First, WordBuilder is designed to help students practice spelling (an alphabetic skill), rather than rhyming (a phonetic skill), to broaden the curricular coverage of the unified system. Second, WordBuilder features collaborative, rather than competitive, gameplay; the robot and child work together to solve a spelling puzzle posed by the tablet, as opposed to the 'first-to-answer-wins' style of RhymeRacer.

4.2.1 Game Play Overview. Much like RhymeRacer, gameplay proceeds through a discrete series of rounds, each associated with a round 'Target' word whose graphic is displayed at the top of the screen. The letters which make up the Target word are randomly placed into letter blocks surrounding a set of (initially empty) letter slots in the center. For example, if the round Target word is SNAKE, the tablet shows an image of a snake and the letters S-N-A-K-E in letter blocks in a random order and location, surrounding 5 empty letter slots (see Figure 2). Within each round, the student and the

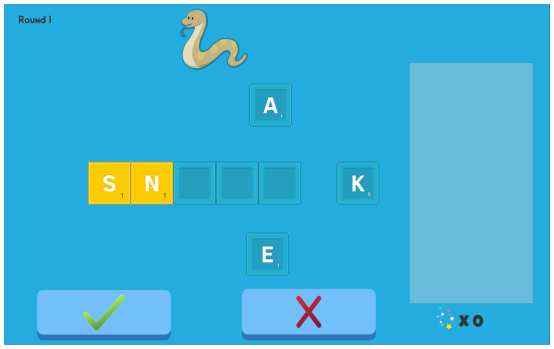

Figure 2: Screencap of a single 'round' of WordBuilder.

robot can each freely place letter blocks into the center squares to spell words; the round ends when the submit button is pressed, and the human-robot team scores points if the team placed all the letters of the Target word into the correct letter slots. The completed word is then displayed on the right side of the screen, and the next round starts.

4.2.2 Cognitive Model. By design, to facilitate multi-task transfer, WordBuilder uses the same Gaussian Process model structure as RhymeRacer, but with a different covariance kernel, better suited to modeling spelling ability. The covariance kernel for the WordBuilder GP is based on a weighted Levenshtein (minimum edit) distance between two words:

$$Cov_{wb}(\{w_i, w_j\}) = v[Levenshtein(w_i, w_j)], \quad (2)$$

where v is a normalization constant. Word distance is defined as the minimum edit distance between two words, normalized to the range $\{0, 1\}$.

When the student correctly spells the target word, that is considered a positive model observation. The final model observation is again adjusted based on the amount of time a player spends spelling each word. If the student cannot spell the target word after an upper time limit is exceeded, the robot spells the complete word and the model updates with an observation score of -1.

5 MODEL TRANSFER APPROACH

Each game stands on its own as an example of an adaptive player modeling system, but our primary goal in this paper is to advance an approach for student model *transfer*. Each game's CognitiveModel takes the form of a Gaussian Process defined over the words in the Curriculum, with the task-dependent 'word-space' geometry defined by the covariance kernel. Let us assume one source game is played without any prior personalization, and several rounds of model observations are collected and used to compute the source task GP posterior. How can we use this data to help the target task model learn more quickly or proficiently?

The most basic approach would be to simply use the source task data as-is, alongside any new target task data, using the target task covariance kernel to compute the posterior. This might be better than nothing, as proficiency in one task is likely to positively correlate with proficiency in another. But this approach fails to consider that the two tasks differ, predictably and quantifiably.

Instead, we propose an instance-weighting approach, whereby we include source task data in the target task training set, but weight each transferred data point differently based on the similarity of the source and target task with respect to that data point. The source and target tasks are defined by their respective GP covariance functions, thus our re-weighting similarity metric is based on the differences in source/target covariance for a particular domain point (i.e. a single word).

The covariance function of RhymeRacer encodes the domain knowledge that words which share a rhyme ending are 'closer' to each other (i.e. if you can correctly identify the rhyming word for DOG, you are more likely to be able to identify the rhyming word for FROG) [16]. Likewise, the covariance function of WordBuilder encodes the domain knowledge that words which share similar letters are 'closer' to each other (i.e. if you can correctly spell CAT, you are more likely to be able to spell CAR). When computing the instance weight of '(DOG, .85)', if knowing DOG impacts the inference of other words in the source task in a way similar to how knowing DOG impacts inference in the target task, then DOG should be weighted roughly equally (i.e. close to 1) in the target task. More concisely, the greater the source-target similarity in word-space geometry around a domain point, the higher the transfer weighting of any source task data at that domain point.

To formalize this intuition, we take the average (over all words in the curriculum) difference between source and target task covariances of the instance word and each other word, giving a measure of how similarly instance word data impacts inference overall in the source and target tasks. Transfer weight, λ_i, of a source task data instance $\{x_i, y_i\}$ is determined by the average difference in source and target task covariance at that point, across all words w in the Curriculum, W:

$$\lambda_i = \frac{\sum_{w \in W} 1 - ||Cov_s(x_i, w) - Cov_t(x_i, w)||}{|W|}. \quad (3)$$

Source data instances with identical target task covariances would get a weight of 1 (indicating the data instance plays an identical role in the target task), whereas data instances with highly dissimilar target task covariances (indicating a very different impact on inference) get a weight of 0 (i.e. removing that instance from the target data entirely). In this paper, data instances are only transferred and reweighted from their originating source task to the alternate target task. However, future research could explore the effect of multiple transfers and reweightings of a single data instance.

6 RESEARCH EXPERIMENTS AND IMPACT

In this section, we describe experimental evaluations of the effect of transfer on student models, using simulation-based student data, establishing theoretical footing for our "multi-task personalization" transfer approach. We close by discussing the implications of these findings for research in multi-task player modeling.

6.1 Simulated Student Data

In this paper we provide experimental simulation results to ground our implementation of multi-task personalization. We derive our

Figure 3: An integrated social robot platform that supports different game "tasks".

student performance data from a *simulated* student, SimStudent, based on a minimal set of modeling assumptions.

Each SimStudent has an internal "true mastery" ($m_w \in [-1, 1]$) for each word in the Curriculum, per game. The SimStudent's true mastery of a word in a game can be interpreted as the student's likelihood of correctly applying the literacy skill to the word (e.g. identify "SNAIL" as the rhyme for "WHALE" or correctly spell "SNAIL" with the letter blocks). The process for generating true mastery values varies by game, and is used to simulate a student's gameplay actions during the game via a noisy sampling process.

Each SimStudent's "performance data" for a word consists of a binary 'correctness' variable corresponding to whether they successfully applied the primary literacy skill of the game to the word (e.g., selected the correct rhyme or correctly spelled the Target word), plus a scalar 'timing' variable corresponding to the amount of (simulated) time taken to answer. Each word-performance pair ($word_i$, $\{correct_i, timing_i\}$) constitutes a single 'sample'.

6.1.1 Simulating True Mastery. Although each game supports the practice of different fundamental literacy skills (rhyming and spelling), both skills are indicators of a meta-linguistic skillset known as *phonological awareness*. To generate the SimStudent's true mastery of each word in each game, we first generate a theoretical "phonological" mastery for each of the 39 ARPAbet phonemes [12], uniformly at random ($m_p \in [-1, 1]$). The phonological mastery that underlies the word-mastery of both games is an implicit modeling assumption, based on decades of research in early childhood literacy development, that there exists a link between a student's rhyming and spelling ability with respect to specific words and phonemes [13]. After random initialization, these phonological mastery values are then further transformed to derive the mastery of each Curriculum word in each game. For RhymeRacer, the mastery of the phonemes that comprise each rhyme-ending (e.g. 'AY'-'N' for 'RAIN' , 'BRAIN', and 'TRAIN') are averaged, and Gaussian noise (centered on the phoneme-mastery mean, $\sigma = .1$) is independently added to compute the SimStudent's true mastery of each word with that rhyme-ending. For WordBuilder, the phonological mastery of all phonemes that constitute a word are averaged to give the SimStudent's true mastery of that word.

6.1.2 Simulating Performance Data from Mastery. The 'correctness' component of student performance is determined by whether the student's true mastery of that word is greater or less than 0 (corresponding to correct/incorrect). However, the value of this component is randomly flipped at a rate equal to 'guess' and 'slip' binomial variables. 'Guess' and 'slip' parameters are common formulations in educational student modeling research[2], which we use here to make our simulated student data more realistic. Respectively, guess and slip parameters correspond to the probability of *correctly* answering a question *without* true mastery or *incorrectly* answering a question *despite* true mastery. For RhymeRacer, we set guess and slip rates at *.25* and *.1*, based on the multiple-choice nature of the round gameplay. For WordBuilder, due to a game design less conducive to successful guessing, the guess and slip rates are set at *.1* and *.1*.

The 'timing' component of student performance is determined by the numerical value of the SimStudent's true mastery, mixed with Gaussian noise. For these experiments, we capped the maximum timing at 10s. The student's true mastery score is binned into deciles, and the final score is calculated by sampling from a Gaussian centered on $10 - MasteryDecile$, so that lower levels of mastery correspond to longer timing components.

6.1.3 Inferring and Evaluating Models of Simulated Students. In our simulation experiments, we create a new SimStudent with a distinct, simulated 'true mastery' of each word in the curriculum per game. Each Gaussian Process student model then has the task of recreating or estimating the true mastery from the derived SimStudent game performance data. As we described in Sec 4, both task models share a domain and basic computational structure, with the primary difference being the covariance functions used to drive inference in each game task. These covariance functions encode task domain knowledge; indeed, as discussed in Sec 5, the covariance functions essentially define the task itself.

From the perspective of a simulation experiment, the underlying domain information (e.g., rhyme-ending equivalence or Levenshtein distance) is encoded in both the CognitiveModel covariance and the sampling process used to generate the SimStudent's 'true mastery'. The true mastery data is further transformed by an unknown (from the perspective of the GP student model), noisy process into student performance data, and the 'task' of the CognitiveModel is to estimate the most likely true mastery distribution.

For our evaluation metric, we use F-1 classification score, the harmonic mean of precision and recall metrics, where the true class label for each word is whether the true mastery of the SimStudent is positive or negative. This classification task may seem coarse compared to numerical regression, but because the sign of a word's true mastery largely determines the correctness component of the SimStudent's performance data (modulo guess and slip factors), this is an important function for the CognitiveModel to accurately determine. Moreover, while we could calculate a model's numerical loss with respect to the SimStudent true mastery, this type of evaluation is not possible with real student data, limiting the utility of any conclusions when generalizing to real human students.

6.2 Simulated Evaluation of Transferrable Models for Multi-task Personalization

Our research questions in this evaluation focus on three main points regarding the effect of multi-task student model transfer on efficiency and quality of student modeling: **(1)** Viability: Does source

task data objectively improve target task performance at all? **(2)** Proficiency: How does the final model for a target task, using both re-weighted source task data and new target task data, perform relative to a model trained with only the new target task data? **(3)** Efficiency: How does a model trained with multi-task data perform compared to a model trained on the same amount of single-task data? How does the learning curve differ between these paradigms?

These questions represent the fundamental measures of success for multi-task personalization. So-called 'negative transfer' occurs when a target task model trained with a mix of source and target task data performs worse than a target task model trained with just the subset of target task data, implying that training on source task data is worse than no data and therefore transfer learning is not viable. A more proficient multi-task model supports the idea that diverse sources of data could lead to models that perform better overall in a complex target task. Finally, in data-sparse domains such as personalized human-robot interaction, more efficient learning implies that multi-task personalization helps overcome some challenges of long-term, personalized agent interaction. Despite their essential simplicity, no student modeling system, to our knowledge, has yet answered these questions.

In each of the results presented, we simulated 30 'rollouts' (each of which represents a single SimStudent) of 60 samples of student performance data. The specific word chosen for each sample is determined by an active learning procedure that selects the word with the greatest posterior variance under the current model, i.e., the word which the model is most uncertain about. After 15, 30, and 45 samples, the multi-task model switches tasks - roughly analogous to a four-session study in which each session constitutes 15 samples and the task alternates every session. The single-task models are analogous to the typical long-term study in which all data (over four, 15 sample sessions) comes from the same task.

Throughout these simulation experiments, we strived to explore test scenarios that mimic realistic operating conditions as closely as possible. In prior work, collecting even 20 good samples from a young student during a single interaction session was considered highly successful [27]. In fact, the relatively low number of personalized data samples in real-world HRI deployments was a major impetus for our investigation of transfer learning for multi-task personalization. Our simulations are computationally efficient enough to support real-time interaction. The average run-time for a complete simulation of 30 rollouts for 3 models (2 single-task, 1 multi-task), each with 60 samples was 210 seconds.

Each of the graphs shown represents a different run of 30 rollouts, with 60 samples in each. Solid colored lines represent the mean performance of each model type for a given number of samples and color shading represents the standard error of the mean model performance across rollouts.

6.2.1 Single-task Learning Curves. Figure 4 show the learning curves of models trained exclusively on single-task data. As the number of training samples increases, the CognitiveModels of both games rapidly increase in F1-score and rollout variance decreases. Around 70 samples, learning plateaus, with both models achieving F1 scores near .9 (1 represents a perfect classifier), which we attribute to having sampled almost every word in the Curriculum. These results are in line with empirical results from human students presented in

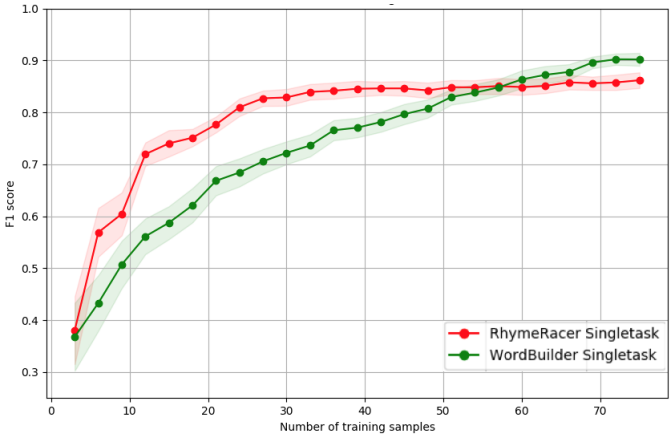

Figure 4: Learning curve for single-task models (F1-score). Both task models learn good classifiers with 60 samples

[27] and [26], which gives us confidence that our methodology for evaluating model performance on simulated student data is realistic enough to effectively characterize the effect of model transfer and multi-task personalized models.

6.2.2 Multi-task Learning Curves. Figures 5 and 6 show learning curves that address the primary questions of proficiency and efficiency of transfer models. Each shows learning curves from separate 30-rollout runs of single-task models for both tasks, plus a transfer model trained first on 15 samples from RhymeRacer, with data then transferred to a WordBuilder model, which is trained further on 15 samples from WordBuilder. The WordBuilder data is transferred back to the original RhymeRacer model, and the process repeats once more, for a total of 60 samples.

Figure 5 depicts this process as a single, extended model trained on 60 total samples split across two tasks and compares the transfer model to single-task models trained on 60 samples from a single task. This presentation makes it easier to compare the complete multi-task personalization model pipeline to models trained completely on single-task data, answering questions about the relative final proficiency of single- and multi-task personalized models.

Figure 6 depicts this process as training two task models, trained on 30 samples each. WordBuilder-1 and WordBuilder-2 benefit from pre-training on the RhymeRacer task, and RhymeRacer-2 benefits from pre-training on the WordBuilder task. This presentation makes it easier to directly compare each component (RhymeRacer-1, WordBuilder-1, RhymeRacer-2, and WordBuilder-2) of the multi-task personalization model to the segment of the single-task model trained on equivalent task data, answering questions about the relative efficiency of single- and multi-task personalized models.

Results shown in Figure 5 suggest that training on an equal mix of reweighted-source and target-task data does *not* lead to a more proficient model in this scenario, compared to training on only target-task data. The single-task models monotonically increase in classification F-score and plateau between .8-.9, whereas the final multi-task transfer model classifier averages around .79 for RhymeRacer and WordBuilder. It is, however, a positive sign that the multi-task model reaches competitive scores on *both* target

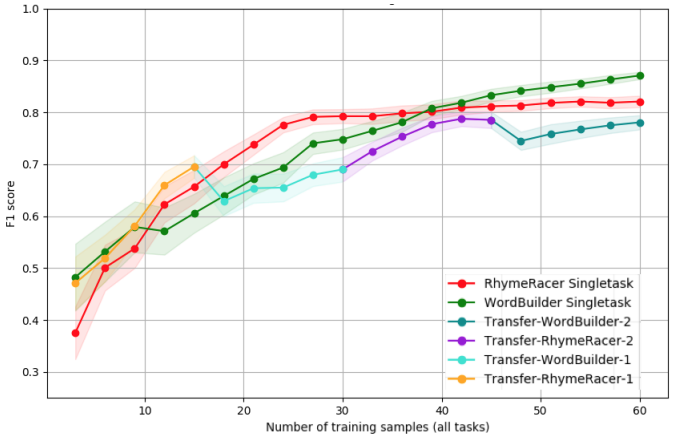

Figure 5: 'Proficiency' of transfer learning compared to single-task models. Transfer model trades off final classifier accuracy for multi-task generality

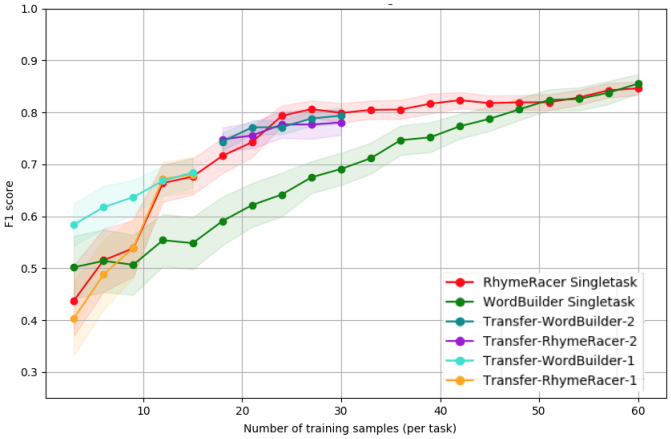

Figure 6: 'Efficiency' of transfer learning model compared to single-task models. Transfer model is equivalent to or better than single-task models with equal amounts of source data

tasks, and therefore conveys more information (at a cost of slightly lower F1-score) of a student's mastery, than a single-task model can. Whether this trade-off is ultimately worthwhile depends on the specific application constraints of such a system.

Figure 6 more clearly shows a stronger beneficial effect of transferrable student models. The RhymeRacer-1 segment of the transfer model tracks the single-task RhymeRacer model almost exactly (as expected, because at this point they are identical models). When the multi-task model transitions to the WordBuilder-1 segment, we clearly see a positive effect of model transfer – the WordBuilder-1 segment far outperforms the single-task WordBuilder model in the early stages of training. This is precisely the beneficial avoidance of 'cold-start' learning that we discussed in Sec. 2.2 and a clear demonstration that negative transfer is not occurring in this phase of learning. After the Transfer model transitions back to the RhymeRacer-2 segment, model performance tracks closely with the RhymeRacer single-task model, which suggests that negative transfer also does not occur when data from WordBuilder are transferred to RhymeRacer. We do not see clear evidence for positive transfer from WordBuilder-1 to RhymeRacer-2 at this point, but upon transferring back to WordBuilder-2, the transfer model again outperforms the single-task WordBuilder model.

7 CONCLUSIONS, LIMITATIONS, AND FUTURE WORK

To summarize our findings in terms of the original research questions: there is strong evidence of positive transfer from RhymeRacer to WordBuilder, which leads to more efficient learning and avoidance of the cold-start problem. There is also clear evidence that negative transfer does not occur from WordBuilder to RhymeRacer, however there is not clear evidence for positive transfer in this direction. There is also weak evidence that multi-task training does not lead to greater final task proficiency in this scenario, though we note that the hypothesized proficiency benefits of multi-task personalization – that learning from multiple, varied data sources can improve prediction performance on complex tasks such as modeling human knowledge – are less likely to be found in a simulation-based environment which, by necessity, cannot fully replicate the complexity and difficulty of the real-world task. This claim, moreso than others, should be further investigated in the context of real human performance data.

In advocating for researchers to evaluate their systems in the real world, Rodney Brooks famously quipped "simulations are doomed to succeed" [5]. We find this philosophy generally laudable, if not always practical. Simulated human data has an accepted role in Human-Robot and Human-Agent Interaction research (with notable examples in human-interactive machine learning systems) [11, 29]. While this project meets the criteria for such a design, we wish to state that this project constitutes *an* evaluation of the proposed transfer method, it is not a *definitive* evaluation. Further research with human subjects will be necessary, not least, because one of the major hypothesized benefits of the multi-task personalization paradigm – increased student engagement – could not be realistically evaluated by simulation experiments.

Whether due to engineering constraints, myopic design, or inflexible modeling frameworks, multi-task player modeling is not yet established as a research area in interactive AI. This paper's core contributions include: a motivation and definition of the multi-task personalization paradigm; two playable game environments; a novel student modeling approach for multi-task personalization based on Gaussian Processes; and a series of experimental simulations that establish the theoretical viability and benefits to learning efficiency from multi-task personalization. Transferrable player models are a clear and important step towards more flexible and general player models. By presenting the first detailed system implementation and empirical results from multi-task personalized models on simulated players, we provide clarity, theoretical grounding, and justification for future in-person evaluations, contributing to research on more efficient and effective personalized models.

8 ACKNOWLEDGMENTS

Thanks to all teammates who have contributed code, content, or constructive comments to our system. This work was supported by NSF Grant IIS-1734443 and IIP-1717362 .

REFERENCES

[1] Ryan S Baker. 2019. Challenges for the future of educational data mining: The Baker learning analytics prizes. *JEDM| Journal of Educational Data Mining* 11, 1 (2019), 1–17.

[2] Ryan SJD Baker, Albert T Corbett, and Vincent Aleven. 2008. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In *Intelligent Tutoring Systems*. Springer, 406–415.

[3] Paul Baxter, James Kennedy, Emmanuel Senft, Severin Lemaignan, and Tony Belpaeme. 2016. From characterising three years of HRI to methodology and reporting recommendations. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 391–398.

[4] Tony Belpaeme, James Kennedy, Aditi Ramachandran, Brian Scassellati, and Fumihide Tanaka. 2018. Social robots for education: A review. *Science Robotics* 3, 21 (2018).

[5] Rodney A Brooks and Maja J Mataric. 1993. Real robots, real learning problems. In *Robot learning*. Springer, 193–213.

[6] Eric Deng, Bilge Mutlu, and Maja J Mataric. 2019. Embodiment in Socially Interactive Robots. *Foundations and TrendsÂő in Robotics* 7, 4 (2019), 251–356. https://doi.org/10.1561/2300000056

[7] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 1126–1135.

[8] Tesca Fitzgerald, Ashok Goel, and Andrea Thomaz. 2018. Human-guided object mapping for task transfer. *ACM Transactions on Human-Robot Interaction (THRI)* 7, 2 (2018), 1–24.

[9] Michael Geden, Andrew Emerson, Jonathan Rowe, Roger Azevedo, and James Lester. 2020. Predictive Student Modeling in Educational Games with Multi-Task Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34.

[10] Goren Gordon, Samuel Spaulding, Jacqueline Kory Westlund, Jin Joo Lee, Luke Plummer, Marayna Martinez, Madhurima Das, and Cynthia Breazeal. 2016. Affective Personalization of a Social Robot Tutor for Children's Second Language Skills. In *Thirtieth AAAI Conference on Artificial Intelligence*.

[11] Shane Griffith, Kaushik Subramanian, Jonathan Scholz, Charles L Isbell, and Andrea L Thomaz. 2013. Policy shaping: Integrating human feedback with reinforcement learning. In *Advances in neural information processing systems*. 2625–2633.

[12] Ben Hixon, Eric Schneider, and Susan L Epstein. 2011. Phonemic similarity metrics to compare pronunciation methods. In *Twelfth Annual Conference of the International Speech Communication Association*.

[13] Torleiv Høien, Ingvar Lundberg, Keith E Stanovich, and Inger-Kristin Bjaalid. 1995. Components of phonological awareness. *Reading and writing* 7, 2 (1995), 171–188.

[14] Bahar Irfan, Aditi Ramachandran, Samuel Spaulding, Dylan F Glas, Iolanda Leite, and Kheng Lee Koay. 2019. Personalization in long-term human-robot interaction. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 685–686.

[15] Eric Klopfer, Jason Haas, Scot Osterweil, and Louisa Rosenheck. 2018. *Resonant games: Design principles for learning games that connect hearts, minds, and the everyday*. MIT Press.

[16] Julia C Lenel and Joan H Cantor. 1981. Rhyme recognition and phonemic perception in young children. *Journal of Psycholinguistic Research* 10, 1 (1981), 57–67.

[17] Blerina Lika, Kostas Kolomvatsos, and Stathes Hadjiefthymiades. 2014. Facing the cold start problem in recommender systems. *Expert Systems with Applications* 41, 4 (2014), 2065–2073.

[18] Francisco S Melo, Samuel Mascarenhas, and Ana Paiva. 2018. A tutorial on machine learning for interactive pedagogical systems. *International Journal of Serious Games* 5, 3 (2018), 79–112.

[19] Stefanos Nikolaidis, Przemyslaw Lasota, Ramya Ramakrishnan, and Julie Shah. 2015. Improved human–robot team performance through cross-training, an approach inspired by human team training practices. *The International Journal of Robotics Research* 34, 14 (2015), 1711–1730.

[20] Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22, 10 (2009), 1345–1359.

[21] Hae Won Park, Ishaan Grover, Samuel Spaulding, Louis Gomez, and Cynthia Breazeal. 2019. A model-free affective reinforcement learning approach to personalization of an autonomous social robot companion for early literacy education. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 687–694.

[22] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. 1532–1543. http://www.aclweb.org/anthology/D14-1162

[23] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. 2009. ROS: an open-source Robot Operating System. In *ICRA workshop on open source software*. 5.

[24] Carl Edward Rasmussen and Christopher K. I. Williams. 2005. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.

[25] Thorsten Schodde, Kirsten Bergmann, and Stefan Kopp. 2017. Adaptive robot language tutoring based on bayesian knowledge tracing and predictive decision-making. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 128–136.

[26] Samuel Spaulding and Cynthia Breazeal. 2019. Pronunciation-Based Child-Robot Game Interactions to Promote Literacy Skills. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 554–555.

[27] Samuel Spaulding, Huili Chen, Safinah Ali, Michael Kulinski, and Cynthia Breazeal. 2018. A Social Robot System for Modeling Children's Word Pronunciation: Socially Interactive Agents Track. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1658–1666.

[28] Samuel Spaulding, Goren Gordon, and Cynthia Breazeal. 2016. Affect-aware student models for robot tutors. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 864–872.

[29] Aaron Steinfeld, Odest Chadwicke Jenkins, and Brian Scassellati. 2009. The oz of wizard: simulating the human for interaction research. In *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*. 101–108.

[30] David Thue and Vadim Bulitko. 2018. Toward a unified understanding of experience management. In *Fourteenth Artificial Intelligence and Interactive Digital Entertainment Conference*.

[31] Paul Vogt, Rianne van den Berghe, Mirjam de Haas, Laura Hoffman, Junko Kanero, Ezgi Mamus, Jean-Marc Montanier, Cansu Oranç, Ora Oudgenoeg-Paz, Daniel Hernández García, et al. 2019. Second language tutoring using social robots: A large-scale study. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. Ieee, 497–505.

[32] Georgios N Yannakakis and Julian Togelius. 2018. Modeling Players. In *Artificial intelligence and games*. Vol. 2. Springer, 203–255.

[33] Zhe Zhao, Lichan Hong, Li Wei, Jilin Chen, Aniruddh Nath, Shawn Andrews, Aditee Kumthekar, Maheswaran Sathiamoorthy, Xinyang Yi, and Ed Chi. 2019. Recommending what video to watch next: a multitask ranking system. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 43–51.

[34] Jichen Zhu and Santiago Ontañón. 2019. Experience management in multi-player games. In *2019 IEEE Conference on Games (CoG)*. IEEE, 1–6.