LIFELONG PERSONALIZATION FOR SOCIAL ROBOT LEARNING COMPANIONS

INTERACTIVE STUDENT MODELING ACROSS TASKS AND OVER TIME

SAMUEL LEE SPAULDING

S.M., Massachusetts Institute of Technology (2015)
B.S., Yale University (2013)

Submitted to the Program in Media Arts and Sciences, School of Architecture and Planning, in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Media Arts and Sciences

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2022

© Massachusetts Institute of Technology 2022. All rights reserved.

AUTHOR

Samuel Lee Spaulding
Program in Media Arts and Sciences
August 18, 2022

CERTIFIED BY

Dr. Cynthia Breazeal
Professor of Media Arts and Sciences
Thesis Supervisor

ACCEPTED BY

Dr. Tod Machover
Academic Head
Program in Media Arts and Sciences

LIFELONG PERSONALIZATION FOR SOCIAL ROBOT LEARNING COMPANIONS
INTERACTIVE STUDENT MODELING ACROSS TASKS AND OVER TIME

by

SAMUEL LEE SPAULDING

Submitted to the Program in Media Arts and Sciences, School of Architecture and Planning, on August 18, 2022 in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Media Arts and Sciences

Abstract:

Early language and literacy skills are important foundations for learning and form the basis of later academic success. Motivated by a growing scientific consensus that language learning requires engaging students cognitively, affectively, and socially, this thesis advances work to develop “social robot learning companions" that engage with and adapt to students across different language/literacy tasks to provide long-term, scalable, and personalized learning assistance. Personalized student modeling helps promote learning and engagement, but sophisticated modeling relies heavily on student interaction data. In order to elicit useful amounts of personalized student data, researchers have increasingly employed “long-term" interaction designs, which occur over distinct sessions at different times.

This thesis broadens the scope of single-task “long-term personalization" to “multi-task personalization" across different tasks. Both “long-term" and “multi-task" personalized interaction designs are mirrored by an associated shift in algorithm and model design: continual learning, which accounts for the temporal sequence in which data is received, and transfer learning, which accounts for the task in which data originates, using data from a ‘source’ task to learn a model in a different ‘target’ task. The combination of these paradigms, which I call “lifelong personalization" could lead to flexible personalized models that can better adapt to individuals over time and across tasks.

This thesis is a presentation and evaluation of continual and transfer learning methods, focusing on their impact on accuracy and data efficiency of personalized student models, and on student learning and engagement. To facilitate this research, I have developed a unified robotic game system for studying lifelong personalization over two different educational games, each emphasizing certain language and literacy skills. The robot’s behavior in each game is backed by 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. By evaluating new methods for flexible, adaptive student personalization within a suite of custom-designed games for promoting students’ language/literacy skills, this thesis contributes both algorithmic and human-centered insights for the future of educational human-robot interactions.

Thesis Advisor:
Cynthia Breazeal

LIFELONG PERSONALIZATION FOR SOCIAL ROBOT LEARNING COMPANIONS

INTERACTIVE STUDENT MODELING ACROSS TASKS AND OVER TIME

SAMUEL LEE SPAULDING

THESIS READER

Dr. Rosalind Picard
Professor of Media Arts & Sciences
Massachusetts Institute of Technology

LIFELONG PERSONALIZATION FOR SOCIAL ROBOT LEARNING COMPANIONS

INTERACTIVE STUDENT MODELING ACROSS TASKS AND OVER TIME

SAMUEL LEE SPAULDING

THESIS READER

Dr. Julie Shah
H.N. Slater Professor in Aeronautics and Astronautics
Massachusetts Institute of Technology

LIFELONG PERSONALIZATION FOR SOCIAL ROBOT LEARNING COMPANIONS

INTERACTIVE STUDENT MODELING ACROSS TASKS AND OVER TIME

SAMUEL LEE SPAULDING

THESIS READER

Dr. Robin Morris
Regent's Professor of Psychology
Georgia State University

LIFELONG PERSONALIZATION FOR SOCIAL ROBOT LEARNING COMPANIONS

INTERACTIVE STUDENT MODELING ACROSS TASKS AND OVER TIME

SAMUEL LEE SPAULDING

THESIS READER

Dr. Hae Won Park
Research Scientist, Personal Robots Group
Massachusetts Institute of Technology

ACKNOWLEDGMENTS

I'd especially like to thank my advisor, Cynthia Breazeal, for taking me on as a student, sharing her wisdom and vision with me and providing outstanding opportunities and resources for my research and always being a supportive advisor. It has been such a privilege and pleasure working with her for the past decade and sharing in the genuinely one-of-a-kind research environment she has created in PRG that crackles with energy powered by Cynthia's unique blend of vision, leadership, energy, taste, charisma, and perseverance.

I'd also like to thank the other members of my thesis committee:

Rosalind Picard has been a key advisor and advocate from my first semester at the Media Lab. I have always benefitted from her keen editorial sense, her expansive vision of a human-centered technological future, and her push for technical and scientific rigor. Thank you for being a supportive mentor

Julie Shah provided important technical perspective and grounding to this research. I'm grateful for her time and attention to the mathematical and algorithmic details of this modeling approach.

Robin Morris has been a patient, optimistic, and gracious partner in this research helping ensure that any evaluation of literacy robots would have real grounding in the learning sciences and the psychology of literacy. His contributions to the experimental design and assurances were incredibly helpful.

Hae Won Park has served variously as big-picture framer, cloud architect, chief motivational officer, and sounding board for this project. When I rejoined PRG for my doctorate, I wasn't counting on having such an outstanding mentor and role model to learn from, but I'm so glad that I did.

Thank you to everybody who worked on this research, the amazing team that helped out with the study and the robot station: Brayden, Ishaan, Huili, Jocelyn, Ben, Vincent. Every UROP who spent a semester or summer working with me. MIT is really such a special place, and its such a privilege to be somewhere with such a high density of very talented, energetic, generous people.

To the other members of the Personal Robots Group, past, present, and future: this group has at various times felt like a family, a home, a workplace, an asylum, a workshop, a spaceship, a civilization, a monastery. Despite existing in a department that is always looking to the future, one of PRG's secret weapons is that we build upon our past. Every generation of students looks back at approaches from the past with a mix of awe (how did they DO that!?) and horror (why did they do THAT?!). Some people contribute foundational infrastructure, others add ornamental flourishes, but every student and every project adds to whatever this crazy HRI social robot future world-vision is that we're all working on, so thank you to the whole PRG universe for inspiration, infrastructure, and insight.

A few members of the core staff deserve special thanks: Jon Ferguson, whose depth and breadth of swiss-army-knife tech skills inspires me to keep honing my craft and reminds me of the compounding benefits of lifelong learning. Meng Xi, who embodies team-first,

can-do attitude and of course.....discipline(!) in all he does for the group. Building Tegas with Meng changed my entire attitude about the importance of workplace organization.

And of course, Polly Guggenheim, who is the wise guru of grad school and eternal saint of PRG. I can't tell how many times I've gone to Polly with my problems, just to hear her throw them right back in my face and say 'HA, you think that's bad - get a load of MY life!'. We commiserate together, rile each other up, I think "oh my god, its all over, how can this go on?!?!", we both threaten to quit, and then .... I leave her office and miraculously feel better?! Polly's revolutionary approach to therapy should be taught in psychiatric colleges.

I also have to thank the MAS staff, especially the Leadership team that stepped up during the turbulence of the past few years. The connections in the MAS community are really deep and that helps us accomplish awesome things together! Thanks to Kevin Davis, Cornell, and Candido for keeping the building upright and being extremely tolerant of all the ridiculous things that go on inside it. Thanks as well to the MAS academic staff – Keira Horowitz, Linda Peterson, Monica Orta, Sarra Shubart, Becca Bly, and Mahy El-Kouedi – who were some of the most helpful people I've ever encountered for demystifying arcane academic regulations. Ryan, Mirei, Kristin, and all the Member Relations team all deserve thanks for managing the magic relationship between member companies and student researchers, somehow making demos always seem like a good opportunity for both sides!

Thanks to Brian Scassellati and the amazing ScazLab group, not only for letting an excited first-year start working in your lab, but for cultivating a community that stretched beyond Yale, beyond the SAR Expedition, to the whole field of HRI for sharing tools, sharpening ideas, and always "doing good stuff".

Thank you to my friends, near and far from MIT, who have been a constant source of fun, insight, and companionship throughout this PhD journey. And the ultimate thank you to my family: Mom, Dad, Matt, Becca, Sarah, to the Jaffe family: Howard, Elisabeth, Matt, and to Caroline who has been my constant partner and companion through the past decade. During a PhD there's so much uncertainty around it, and having people alongside you that you can trust and be certain they'll be there for you is so invaluable and there's just no way I'd be able to be standing here today without them.

This research was supported by the National Science Foundation (NSF) under a NSF Graduate Research Fellowship and Grants IIS-1734443 and IIP1717362.

LIST OF FIGURES

LIST OF TABLES

CONTENTS

LIFELONG STUDENT MODELS FOR LONG-TERM LITERACY PRACTICE: PERSONALIZATION ACROSS TASKS AND OVER TIME

1

INTRODUCTION

The ultimate goal of this research is to develop better interactive robots that can deeply personalize to individuals over long-term interactions. These robots could be invaluable resources that could foster learning in ways similar to those of the best human teachers, yet still provide the advantages of digital technology such as data fluency, always-on availability, and scale of distribution. Educational researchers have long recognized that a *personalized* approach to pedagogy is one of the best ways of promoting learning (Pane et al. [2015]; Bernacki et al. [2021]), yet in a world with increasing demand for education, the availability of qualified teachers has not kept up with the demand from students. Technology has an important role to play in realizing the vision of personalized education for all. In recognition that learning is not only a cognitive process, but also an emotional and a social process, in this thesis I propose to design *social robots learning companions* that are capable of modeling students, adapting to them, and introducing them to educational material that is best suited for each student, presented in a way that takes into account their individual learning differences.

A 2018 review (Belpaeme et al. [2018]) on the use of social robots as educational tools concluded "[social robots] have been shown to be effective at increasing cognitive and affective outcomes and have achieved outcomes similar to those of human tutoring *on restricted tasks*" (emphasis mine). What are these restricted task scenarios? "short, well-defined lessons delivered with limited adaptation to individual learners or flexibility in curriculum". Results from studies of single-session tutoring interactions with limited personalization paint an overall picture of benefits that are stable, positive, and modest. In order to improve the impact of social robot tutoring technology, researchers are looking towards educational interactions where personalization plays a larger role, and to long-term interactions to develop deeply personalized models.

Despite general recognition that long-term interactions enable a more impactful approach for the field, developing agents capable of sustaining long-term interactions is no simple feat. Some of the challenges researchers face in sustaining long-term interactions include lower student engagement (due to repetitive interactions and declining novelty), personalized models that represent only limited aspects of student mastery (narrowly focused models are more straightforward to implement and require less data to train), and early stopping (due to cold-start model learning, leading to poor model performance in early sessions).

1.1 MULTITASK PERSONALIZATION: PERSONALIZED MODELS ACROSS TASK CONTEXTS

multitask personalization

To address some of these challenges, I introduce an approach to long-term interaction design called '*multitask personalization*' in which students interact with a social robot across different task contexts throughout a long-term interaction. Within each task, the robotic interaction partner learns a task-specific personalized model of the student that is *transferrable* across tasks throughout the long-term interaction, i.e., data collected from earlier interactions with a student on a prior task can be used to improve personalized model learning in a new task.

A multitask personalization approach has potential to address many of the practical challenges associated with sustaining long-term interactions. Student engagement is likely to remain higher over time when engaged in different, varied tasks with a learning partner, compared to repeating the same task multiple times. Personalized student models can also draw on data from a wider variety of task contexts in order to learn a more multifaceted picture of a student's mastery. And transferring data from interactions on prior tasks can help speed up model learning on a new task, reducing the risk of early stopping from cold-start learning.

In prior published research (Spaulding et al. [2021b]), I outlined the theoretical benefits of a multi-task personalization paradigm and evaluated the combined-task proficiency and data efficiency of the approach in models trained to estimate simulated student mastery in two different game tasks. These games, called WordDecoder and WordBuilder, were developed in partnership with experts in children's media and early literacy learning, and were designed to help young students practice different literacy skills, namely decoding and starting-sound identification.

I developed a flexible Gaussian Process-based approach to modeling student knowledge in each game task, with an instance-weighting protocol based on task similarity that allowed for data transfer across tasks. This analysis showed that multi-task personalization improved the sample-efficiency of model training, and was particularly useful for avoiding the problem of 'cold-start' modeling. This research was conducted with the assumption that student knowledge was static, i.e., that students' level of knowledge was fixed throughout the interaction sequence.

In subsequent research (Spaulding et al. [2021a]), to further validate the potential of multitask personalization for real-world scenarios — and recognizing that in real human-robot educational interactions, a student is not a fixed target but a dynamic one — I augmented the original multitask personalization approach with a continual learning module that enabled the joint system to better support personalized modeling of dynamic/non-stationary targets.

1.2 CONTINUAL LEARNING: PERSONALIZED MODELS OF NON-STATIONARY TARGETS

Continual Learning

'*Continual Learning*' (CL) – a "learning paradigm where the data distribution and learning objective change through time, or where all the data...are never available at once" (Lesort et al. [2020]) primarily deals with the issue of distributional *shift* over time, recognizing that, in the real world, temporal data are not independent and identically distributed (IID), but rather drawn from a distribution that may change over time, but without a clear signal of such a shift (Lesort et al. [2020]). Continual Learning techniques attempt to improve model performance as this shift occurs, often with an implicit assumption that such shifts will be relatively smooth.

Multitask learning

Multitask learning, on the other hand, focuses more on learning distinct tasks with clear boundaries. In a typical multitask learning scenario, a learner knows from which tasks its training data originated, assumes that each such task is stationary and that its data is IID, and, frequently, the training data arrive in a batch, rather than over time. These broad distinctions can largely be characterized by a focus on task 'shift' versus task 'switch'. However, this boundary is not always strict, and researchers often work to address both issues simultaneously (e.g., Ruvolo and Eaton [2013]).

long-term interactions

As human-robot interaction (HRI) researchers have begun to adapt research methods towards *long-term interactions*, continual learning methods have become more popular in the algorithms and models underlying these interactions. Churamani et al. have detailed many advantages of adopting a continual learning approach in developing affect-aware interactive robots Churamani et al. [2020]. The authors highlight a number of important shifts in viewpoint when adopting this approach, recognizing that human affective response is idiosyncratic (personalized), dynamic (changes over time), and contextual (changes with task or environment). I argue that these same qualities apply more broadly, to many aspects of human interactive behavior, though in this thesis I primarily focus on student learning in educational interactions. Indeed, some of the most salient markers of learning behavior *are* affective behaviors, therefore it is only a short conceptual leap to hypothesize that the benefits of Continual Learning applied to affect recognition and response may prove similarly beneficial when applied to recognizing and responding to student learning behaviors.

CognitiveModel

Though Churamani et al. did not explicitly refer to *long-term* interactions (LTI) with users, the theoretical frameworks of continual learning and multitask personalization are natural fits for the practical goal of sustaining long-term interactions. My goal with this thesis is to demonstrate the strengths of this combined approach by emphasizing their benefits in the application domain of an agent attempting to model student knowledge in the form of a *CognitiveModel*. Students' knowledge is idiosyncratic (each student has their own private mastery model), dynamic (this model can change over the course of an interaction), and contextual (student knowledge can manifest differently in different task contexts). A modeling approach that

acknowledges and accounts for these qualities may be the key to successful, personalized long-term interactions.

1.3 LIFELONG PERSONALIZATION: PERSONALIZED MODELING ACROSS TASKS AND OVER TIME

In this thesis, I introduce computational methods that move beyond the traditional algorithmic view of modeling student knowledge as supervised learning of a fixed target, or "estimation" of mastery on a single task. Instead, I adopt a broader view of student modeling that incorporates ideas from both continual and multitask learning into an approach to long-term student modeling as a process of personalization *over time* and *across tasks*, which I refer to as 'lifelong personalization'.

To motivate the use of this term and connect the dots between various methods referred to in other literature, I outline here the relational structure of several key concepts used throughout the remainder of the paper.

Personalized student modeling

Personalized student modeling has been shown to help promote **student learning and engagement** (Lindsey et al. [2014]; Ramachandran et al. [2019]; Park et al. [2019]; Yudelson et al. [2013]). In order to advance the degree and sophistication of personalized modeling, we require **personalized interaction data** from a student. To elicit useful quantities and kinds of personalized student data, researchers have been looking towards ***long-term interaction*** designs (Leite et al. [2013]), which occur over several sessions at different *times*. After observing shortcomings of single-task longitudinal interactions, I introduced the idea of "***multitask personalization***" for interactions which occur in different *task contexts* (Spaulding et al. [2021b]). Each of these paradigm shifts in interaction design are mirrored by an associated paradigm shift in algorithm and model design: **continual learning**, which accounts for the temporal sequence in which data is received and assumes a dynamic or non-stationary modeling target, and ***transfer learning*** which accounts for the task in which training data originated and uses data from one 'source' task to more quickly learn a model in a different 'target' task. When these two algorithmic paradigms are combined, yielding flexible personalized models that can model individuals over time and across tasks, I call this ***lifelong personalization***, based on Parisi et al.'s definition of lifelong learning systems as "an adaptive algorithm capable of learning from a continuous stream of information, with such information becoming progressively available over time and where the number of tasks to be learned (e.g., membership classes in a classification task) are not predefined" (Parisi et al. [2019]). This concept structure is represented graphically in Figure 1.

long-term interaction

multitask personalization

transfer learning

lifelong personalization

Of course, these are not universal definitions of these terms, and researchers may interpret or use these terms in slightly different ways. In this thesis, I primarily apply these definitions in the context of *personalized interactive modeling*. Some works look at separate individuals as separate tasks (Jaques et al. [2017]), others consider non-stationary task learning as the primary hallmark of lifelong learning (Xie et al. [2020]). For the purposes of this thesis,

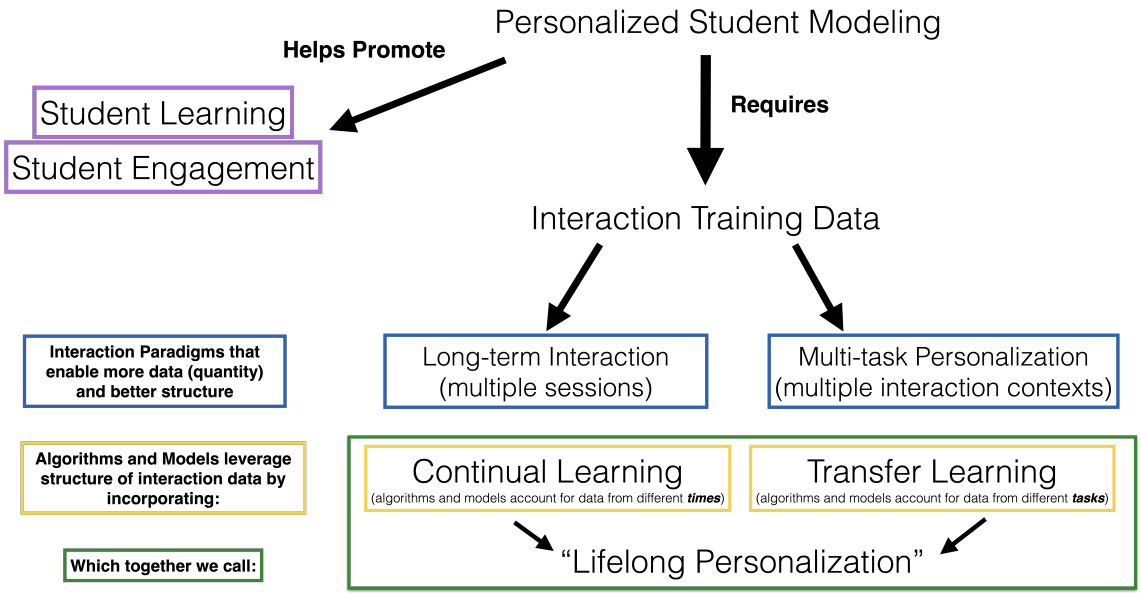


Figure 1: Conceptual structure of terms and goals.

however, we restrict our discussion to the application of these paradigms in the context of learning models of an individual over time and across tasks for adaptive personalization.

2

BACKGROUND

2.1 SUMMARY OF RESEARCH CONTRIBUTIONS AND APPROACH

While personalized social robot systems have been shown to improve student learning over long-term interactions in pre-registered, large-sample trials (Vogt et al. [2019]), 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 (Finn et al. [2017],Zhao et al. [2019]), 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 the limitations of single-task repeated interactions, I developed a "multi-task personalization" transfer learning approach in which students play multiple distinct games, with interaction data and inferred player models transferred across games. I 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 (Irfan et al. [2019]).

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 or

spelling skills, our system design connects both skills to give a more complete picture of a student's learning progress in shorter time.

In addition, I propose to incorporate continual learning methods into multitask transfer, improving the ability of the underling Gaussian Process models to adapt to nonstationary modeling targets, a combined approach I call 'lifelong personalization'.

2.1.1 *Overview of Approach*

Transfer learning (Pan and Yang [2009]) 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 as the degree of similarity between source and target task formulations.

In this thesis, the source and target tasks are 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 the 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.

Gaussian Process

Curriculum

covariance kernel

For example, in WordDecoder, the primary literacy skill the game is designed to assess and encourage is *decoding*: an observation of a student correctly identifying the sounds corresponding to the letters 'FALL' should increase the model estimate that the student is likely to also be able to decode the letters 'BALL'. This 'closeness' is reflected in the design of the covariance kernel for WordDecoder (see Section 4.1.3). Likewise, in WordBuilder, the primary literacy skills guiding the game design are *letter-sound pairing* and *starting-sound identification*. An observation of a student correctly identifying the beginning letter of 'SNAKE' should increase the model estimate that the student is likely to be able to identify the beginning letters of 'SNAIL'.

Our approach for multitask 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 decoding 'BALL'), we look at the difference in the covariance between source and

target tasks for 'BALL' and all other words in the CURRICULUM. See Section 4.2 for greater detail on the instance-weighting transfer algorithm.

Continual Active Training Data Management

Our approach to continual learning consists of a data structure and associated algorithm for '*Continual Active Training Data Management*' or **CATDaM**. Because Gaussian Processes do not naturally assign temporal information to training data, they can be slow to adapt if the underlying distribution generating the data shifts. With **CATDaM**, we augment each Gaussian Process with a 'memory' that tracks the temporal and interactive context in which training instances (i.e., observations of student learning behavior) are received, and actively reduce the weight of 'stale' data that may no longer reflect the student's underlying knowledge. For example, if a student fails to correctly decode "DOG", but later the robot gives a demonstrative lesson showing how to decode that word, **CATDaM** allows the model to reduce the weight on the prior missed opportunity, knowing that the student's mastery may have changed after the lesson.

2.2 RELATED WORK

2.2.1 *Social Robots as Adaptive Language Learning Companions for Children*

Social robots' ability to interactively engage students has received increasing attention in the past decade (Belpaeme et al. [2018]). Prior work has shown how social robots can significantly increase children's engagement and language/literacy skills, from vocabulary acquisition (Schodde et al. [2017]) to word decoding (Spaulding et al. [2016]) and complex narrative generation (Park et al. [2019]). 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 a 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 appears to be motivated by competition or collaboration or how best to encourage students after a setback. Field research studies (Vogt et al. [2019], Gordon et al. [2016]), 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*" (Baxter et al. [2016]).

novelty effect

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.

2.2.2 *Player Modeling in Interactive Games*

Adaptive player modeling is an umbrella term for techniques using player data to make inferences that affect subsequent gameplay (Yannakakis and Togelius [2018]). Sometimes called 'Experience Management' (Thue and Bulitko [2018]), 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" (Zhu and Ontañón [2019]).

Adaptive player modeling

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 (Lika et al. [2014]). 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.

cold start learning

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 (Geden et al. [2020]). 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.

2.2.3 *Transfer Learning and Nonstationary Modeling in Gaussian Processes*

In general, rather than compiling laundry lists of related citations, I introduce and cite prior work at relevant sections throughout this thesis. However, owing to the more abstract nature of the following articles and less *direct* applicability to the following empirical content, I wish to briefly highlight some

particularly helpful articles that inspired this project in the area of transfer learning and nonstationary modeling, as applied to Gaussian Processes.

Soh et al.'s formulation of transferrable trust models using Gaussian Processes uses a similar kernelized 'task ' representation to our design of task-specific CognitiveModels (Soh et al. [2020]). Snoeke and Adams outlined an 'input-warping' method to address nonstationarity in Gaussian Processes that provided a clear exposition of theoretical capabilities of GPs to handle nonstationary functions (Snoek et al. [2014]). Cao et al. introduced us to the idea of transfer-coefficient based instance-weighting for Gaussian Processes (Cao et al. [2010]), and our evaluation measures of transfer viability, efficiency, and proficiency are based on discussion in Rosenstein et al. [2005].

2.2.4 *Perspectives on Lifelong Personalization*

Long-term or *Longitudinal* **Interaction** (LTI) is a term used to refer to interactions between a user and an artificial agent that unfold over multiple distinct encounters (Irfan et al. [2019]). In other words, "long-term interaction" describes a practical paradigm for designing and evaluating interactions between users and agents. In the context of educational interactions, long-term interactions have followed a pattern of users engaging in a single repeated interaction structure (e.g., playing a single game or answering questions) with updated content reflecting the output of increasingly personalized models trained on data from the previous interaction sessions (Leite et al. [2013]). While this type of repeated single-task interaction has formed the bulk of long-term interaction research to date, there is a recognition that we may be near the useful limit of current single-task paradigms, and that future breakthroughs in sustaining long-term interactions will come from research developing agents that can personalize to a user's changing behaviors and preferences over time and across task contexts.

Many researchers, across a wide swath of computer science and artificial intelligence have written in recent years about the pitfalls and promise of adaptive personalization, long-term interactions with intelligent agents, cross-task generalization, and the benefits that systems exhibiting these capabilities may bring to society:

Johnson and Lester, in an article reflecting on 20 years of research to predict future trends for pedagogical artificial agents wrote: "Conventional domain-specific learner models may be useful for pedagogical agents in the short term, but they will be of limited value over time as learners move between learning experiences. (Johnson and Lester [2018])"

Melanie Mitchell, weighing in on the utility of modern AI systems, wrote:

> In fact, the theoretical basis for much of machine learning requires that training and test examples are 'independently and identically distributed' (IID). In contrast, human learning — and teaching — is active, sensitive to context, driven by top-down expectations, and transferable among highly diverse tasks, whose instances may be far from IID. (Mitchell [2020])

Zhu and Ontañon have written cogently about the “personalization paradox", the tendency of adaptive systems to induce distributional shift in the subject of the model

> “The key underlying problem is that while user modeling tries to acquire a model of some aspects of interest of the user (such as their preferences), personalized adaptation changes the context the user interacts with....A special case of this problem occurs when the goal of the personalization system is to induce behavior change....the system’s explicit goal is to push the user’s preferences or behavior in a particular direction. As a result, user modeling might reflect the user at the start, rather than what she has become."(Ontañón and Zhu [2021])

Finally, in a lecture addressing future challenges for the field of Learning Analytics, Ryan Baker identified *“transferability"* as the first of a series of challenge problems for the field to tackle over the next 20 years, writing

> A modern learning system learns a great deal about a student — their knowledge at minimum, and increasingly their motivation, engagement, and self-regulated learning strategies. But then the next learning system starts from scratch...It’s like there’s a wall between our learning systems...If you seek better learning for students, tear down this wall! (Baker [2019])

Fundamentally, personalized student data remains a major practical challenge towards achieving successful interactive educational systems. Single-session educational interactions in HRI (e.g., some reviewed in Belpaeme et al. [2018]) generally do not provide enough data to learn interesting and distinct personalized models capable of sustaining extensive learning gains or engaged interaction in the long-term. Thus far, successful examples of long-term adaptive personalization tend to repeat a carefully designed interaction centered on a single task over several sessions to augment the dataset (Ramachandran et al. [2019]; Park et al. [2019]).

3

SYSTEM, METHODS, AND APPARATUS

3.1 PERSONALIZED LITERACY GAME SYSTEM

To investigate the algorithmic effects of multitask personalization and lifelong learning in students, researchers in the Personal Robots Group have developed an integrated, deployable social robot system capable of sustaining language/literacy practice between young students and a robot through game-based interactions.

We have used this system to investigate multitask personalization via player model transfer between two games, called WordDecoder and WordBuilder, which are designed to help young students practice a variety of early literacy skills through interactive co-play with an adaptive, personalized robot tutoring agent. Both games were developed for Android tablets using the Unity game engine, and receive robot action command and relay player input through ROS (Quigley et al. [2009]) to a backend system controller. The games were developed for children learning to read, approximately ages 5 to 7, and throughout the design and development process we consulted experts in children's media design and early childhood literacy to ensure that both the content and game designs would be age-appropriate and aligned with the overall educational goals of the project.

As the child and robot play each game together, the robot tutoring agent learns a Gaussian Process model, which we refer to as the CognitiveModel, that estimates the child's 'mastery' of the game. Both games share a Curriculum of words, which serves as both a list of words a student can encounter in the game as well as a unified domain space for the underlying CognitiveModels of each game. In other words, the CognitiveModel is an estimate of how likely the student is to successfully apply the primary literacy skill (rhyming, spelling) to each word in the Curriculum, based on observations of their prior gameplay. Each game has undergone playtesting validation and the Curriculum was curated by experts in early childhood learning to ensure a representative set of 74 words that are generally phonetically, orthographically, and semantically (e.g. animals, foods, household items) age-appropriate and regular.

The personalization model employed is implicitly based on a theory of 'mastery learning', in which a learner's current knowledge forms the basis of subsequent lessons, with an emphasis on in-task performance, skill mastery, and learning efficiency. While mastery learning is one of the most popular theories of learning for computational modeling, Bernacki et al. give an excellent overview of the various ways in which personalized learning systems implicitly correspond, in part or in whole, to a wider myriad of learning theories (Bernacki et al. [2021]).

Figure 2: An integrated social robot platform that supports different game “tasks".

3.1.1 *Cloud-connected Deployment Station*

As part of this thesis, I contributed to the development of cloud-based infrastructure systems to support long-term, unmanned deployments of social robots. These systems must possess an unusually high degree of reliability and robustness. They must be capable of remote monitoring, updating, and troubleshooting, even if one or more components fail, or if underlying infrastructure outside of the team’s direct control is disrupted (e.g., power or internet access at the deployment site).

Jibo ***Robot***
Intel NUC

The basic unit of the deployment architecture is the **Station**, which consists of a *Jibo* ***Robot*** set inside a plastic **Housing**. Within the **Housing** is: a Samsung S5 **Tablet**, **Front-facing Camera**, and an ***Intel NUC***, (an onboard computer that serves as the primary remote access point for the station). The NUC runs several docker containers that support the games, robot, sensor devices, and communications between them. S02-ROS-usb-cam is a container that runs on startup and interfaces with the device hardware (e.g. **Front-facing Camera**). The remaining containers are all instances of the same image, the *mitprg/ros-bundle* package. This package includes Affdex binaries, ROS, the latest versions of the game controllers, and a game launcher module that communicates with the tablet and invokes the various game controllers as necessary. Figure 2 shows a recent picture of the operational station.

3.1.2 *WordDecoder and WordBuilder: Designing Games for Early Literacy*

The gameplay and design of WordDecoder and WordBuilder centers around the family of literacy skills known as *phonological awareness* and *phonemic articulation*. In plain English, these are early literacy skills that center on familiarity with recognizing and reproducing the basic sounds of spoken language. Throughout this thesis, we have taken substantial domain guidance from the Phonemic Awareness Literacy Screening (PALS) project

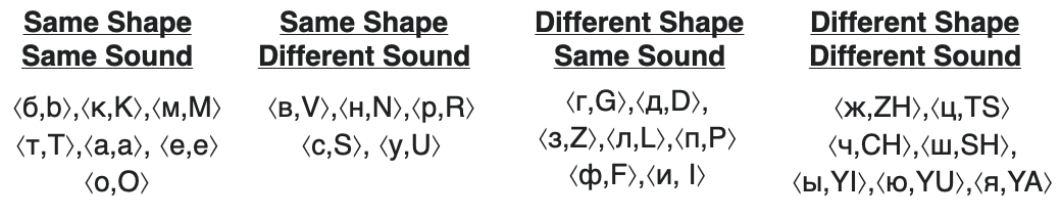


Figure 3: The Russian alphabet can be roughly divided into four categories based on their familiarity to English speakers. Some letters have a similar symbol and similar sound, others have unfamiliar symbols and make unfamiliar sounds.

(Marcia Invernizzi [2015]), a set of resources developed by the University of Virginia and the Virginia Board of Education that 'provides a measure of children's knowledge of several important literacy fundamentals: phonological awareness, alphabet recognition, concept of word, knowledge of letter sounds and spelling". PALS has been extensively researched and validated as a useful tool for assessing early literacy skills, particularly in the context of early reading interventions, in both mono- and bi-lingual populations Huang and Konold [2014]. Specifically, WordDecoder is designed to help assess and promote 'letter-sound pairing', and WordBuilder is designed to help assess and promote 'starting-sound identification' (Marcia Invernizzi [2015]).

3.1.2.1 *Russian language game redesign*

These games went through several design revisions, and due to the COVID-19 pandemic, were adapted in several keys ways to fit new experimental realities. While they were originally designed and playtested for young readers of English, in March 2020 MIT's Institutional Review Board prohibited research activity with minors. Over the course of the following two years, these prohibitions would be gradually relaxed, but research with our original target population (students from ages 5-7) would not be permitted until April 19th 2022, nearly two years after the original restrictions. In order to keep making progress, in September 2021, we decided to re-orient our final human-subjects evaluation around vaccinated MIT undergraduates. Both games were redesigned for second-language learning, specifically learning Russian.

Russian has a number of nice properties, especially because learning its alphabet preserves some of the difficulty of mapping symbols and sounds that early readers experience. The Cyrillic alphabet features letters with varying degrees of similarity to the Latin alphabet. Some letters (e.g. 'M', 'T') have the same sound and symbol in both alphabets. Others require mapping familiar symbols to new sounds, learning new symbols, or even learning new sounds altogether.

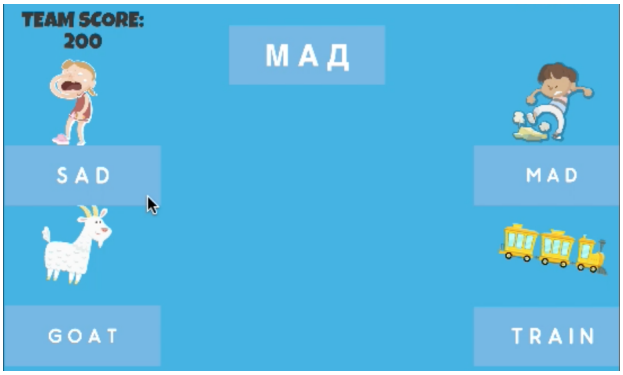


Figure 4: A round of WordDecoder, adapted for Russian language learners. MAD is the Target word, other words are SAD, GOAT, and TRAIN.

3.1.2.2 *WordDecoder*

WordDecoder is a two-player game that proceeds in a series of rounds, each of which offers a chance for either the robot tutor or the student to select a word that is 'encoded' in the letters. At the start of each round, the tablet shows the letters that spell out the 'Target' word at the top of the screen (see Figure 4), with four graphic images, smaller pictures of other words from the Curriculum, below. Exactly one of these word graphics is the 'Target' word spelled out by the letters. The first player to correctly decode the Prompt and touch 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).

3.1.2.3 *RhymeRacer - an early version of WordDecoder*

RhymeRacer was an early version of WordDecoder that was designed to help assess and promote 'rhyme awareness' . In this version of the game, the robot tutor or the student are asked to select the word that rhymes with a central 'prompt word'. At the start of each round, the tablet shows a picture of the 'Prompt' word in the center of the screen (see Figure 5), surrounded by four 'Target' word graphics, smaller pictures of other words from the Curriculum, exactly one of which rhymes with the Prompt word. The tablet also gives a recorded audio prompt, saying "What rhymes with [Prompt Word]?" as the images are displayed. The first player to correctly tap on the rhyming Target word graphic is awarded points, after which the graphics clear and the next round begins.

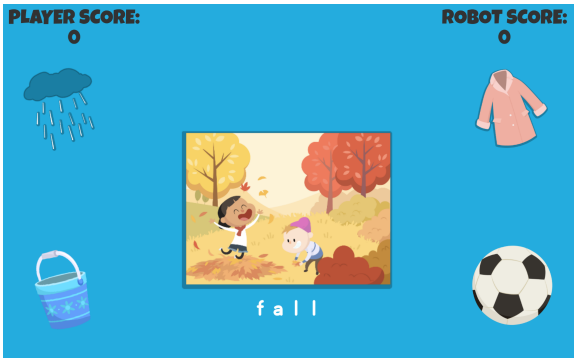


Figure 5: A round of RhymeRacer. FALL is the Target word, Prompt words are RAIN, COAT, PAIL, and BALL.

3.1.2.4 *WordBuilder*

WordBuilder is the second game developed to study multitask personalization in long-term interactions. It was specifically designed to complement WordDecoder and went through a similar design process, including playtesting, consultation with educational experts, and content and asset revision by experienced children's media designers. Most of the visual assets are shared across both games, including the graphics of the Curriculum words, both to help reinforce students' understanding, and also, practically, to help ensure that the correlation between student performance in the two games is based on students' mastery of the underlying skills, not on factors related to the game interface design.

WordBuilder serves as a counterpart to WordDecoder in two main ways: First, WordBuilder is designed to help students practice Letter-Sound pairings and starting-sound identification (phonetic skills), rather than decoding (an alphabetic skill), to broaden the curricular coverage of the unified system. Much like WordDecoder, 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.

At the start of each round, participants hear the translated Target word pronounced out loud by the tablet. The letters which spell out the translated Target word are placed in letter slots in the center, *except* for the letter that forms the "starting sound" (phoneme) of the word. Surrounding the center letters are the 'true' starting letter and 5 distractors in a random order and location (see Figure 7). Within each round, the student and the robot work together to select the correct starting letter, based on the pronunciation of the Target word and the sound of each letter. The round ends when the submit button is pressed, and the human-robot team scores points if the team placed the correct starting-sound letter of the Target word into the letter slot. The completed word is then displayed on the right side of the screen, and the next round starts.

These games were designed together to study multitask personalization in long-term interaction. They share several task design qualities that are advantageous for enabling transfer (indeed, they were designed with trans-

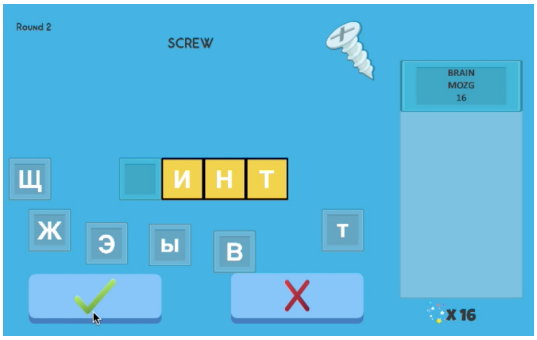


Figure 6: Screencap of a single 'round' of WordBuilder. Participants hear the translated Russian word pronounced and see all letters spelled out except for the starting letter. Participants have to use their knowledge of Russian-English letter-sound pairings to select the correct starting letter

fer specifically in mind): a common software and system architecture and, notably, the word-space Gaussian Process modeling paradigm described in Section 4.1. Transfer learning problems in which the source and target task share a domain (input and/or label space) are termed 'homogeneous' transfer problems, and are generally considered more straight-forward to address than the broader class of 'heterogeneous' transfer problems (involving different domains), which are essentially unbounded in difficulty.

Our approach to student model transfer between WordDecoder and WordBuilder represents a solution to 'partially heterogeneous' transfer problems: those which involve overlapping, but not exactly equal, domains. In this case, the input space is the Curriculum of words used in each game. Each game has its own Curriculum, designed based on the specific literacy skill emphasized and selected from a larger pool of age-appropriate word lists for early readers. For instance, WordDecoder has more words with shared rhyme endings, and WordBuilder has more words clustered around shared starting sounds. Despite the inexact overlap in task curricula, our transfer method is capable of transferring not only shared words from source task to target task, but even source task words that do not appear in the target task (and vice versa). More detail on this procedure is provided in Section 4.2.1.

3.1.3 *Strategy and Content Models: Adaptive Gameplay and Content Personalization via Cognitive Modeling*

By committing to a nonstationary model of student learning, this project puts new emphasis on the robot tutor's *demonstration actions*. At various points in both games, either the student or the tutor will have an opportunity to respond to a word presented from the Curriculum. The ContentModel determines which specific words, drawn from the Curriculum, are presented, while the *StrategyModel* determines whether the student is prompted to respond (giving a 'sample' of training data for the CognitiveModel) or whether the robot responds (providing a 'demonstration' that can potentially improve student learning). This paradigm of interwoven 'samples' and

StrategyModel

'demonstrations' that mix assessment and learning is an example of the 'stealth assessment' design pattern, commonly used to achieve educational goals in interactive games without breaking immersion and experience flow Shute and Ventura [2013].

The StrategyModel is parametrized by a single weight, ω, which incorporates the recent history of gameplay and the amount of model data to balance the robot's strategy with respect to student engagement. At each action decision point, the StrategyModel selects probabilistically from two "strategy actions" — OBSERVE and DEMONSTRATE, choosing OBSERVE with probability ω and DEMONSTRATE otherwise. When the StrategyModel selects the OBSERVE action, it gives the child an opportunity to respond, prompting a response if none is immediately forthcoming. When the StrategyModel selects the DEMONSTRATE action, the robot proactively gives its own response: a correct answer and an explanation of its reasoning. Originally, the games featured a larger robot action space, including the opportunity for the robot to take actions such as "EASY WIN", introducing a word that the model predicts a student would be likely to answer correctly, and "MAKE MISTAKE", whereby the robot intentionally makes a mistake. These actions were intended to boost students' *affective* state, by instilling confidence through correct answers or mitigating disappointment by showing that all players make mistakes. However, as the focus of this thesis sharpened to understanding the impact of transfer learning on *cognitive* modeling, we excluded these actions from the final study.

The ω parameter (Equation 1) essentially moderates the rate of 'exploration' versus 'exploitation' in the robot's behavior: exploration corresponds to the OBSERVE action, obtaining more information for the CognitiveModel, exploitation corresponding to the DEMONSTRATE action by myopically pursuing learning gains. ω naturally rises over the long term as the model gets more samples, but in later rounds, a robot that always chooses the DEMONSTRATE action might not be very encouraging for a student to interact with. In order to align the agent's StrategyModel with the short-term gameplay context, we add a penalty term for each robot demonstration in the past 5 rounds, to ensure students continually have the opportunity to fully participate in answering.

$$\omega = .25 + .05 * n - .5 * (p_r) \tag{1}$$

where n is the number of samples the model has already OBSERVEd and p_r is the percentage of DEMONSTRATE actions taken by the robot in the last 5 rounds.

ContentModel

The *ContentModel* determines what specific words from the Curriculum are presented to the players, and in what order. In this project, the ContentModel selects words via an Active Learning protocol, which selects the word that best aligns with the goals of the tutor's selected strategy, given the current estimated CognitiveModel of the student. For instance, if the current tutor strategy is OBSERVE, the ContentModel

selects the word with the maximum uncertainty under the most recent posterior CognitiveModel, i.e., the word where the agent is least confident about its estimate. If the current tutor strategy is DEMONSTRATE, the ContentModel selects the word with lowest variance of all words with negative posterior mean, i.e., the word that the agent is *most confident* that the student has *not mastered*.

In order to effectively teach a student, the agent must know what words the student has already mastered and which it has not. Therefore the agent faces the twin challenge of simultaneously estimating a student's individual knowledge state while using its latest estimate to teach new content, though the act of teaching itself may change the student's underlying knowledge. The StrategyModel balances these two objectives, while the ContentModel employs active learning to improve both objectives, speeding up both model learning and student learning. As the number of demonstrations increases over time, shifting the student's knowledge and, therefore, the distribution from which their observed 'samples' are drawn, the tutoring agent employs a form of 'negative' active learning to *remove* past samples from the CognitiveModel training set. We expand on the implementation of this 'continuous active training data management (**CATDaM**) in Section 4.2.2.

4

GAUSSIAN PROCESSES: FLEXIBLE IN-GAME STUDENT MODELING

4.1 GAUSSIAN PROCESSES: FLEXIBLE IN-GAME STUDENT MODELING

4.1.1 *Gaussian Processes Overview*

The fundamental modeling approach behind each game's estimate of a student's cognitive task mastery is Gaussian Process (GP) regression in a domain space of words from the Curriculum, essentially identical to the model described in Spaulding et al. [2018]. A Gaussian Process 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. Put differently, a Gaussian Process (GP) is a distribution over functions, defined over some input domain, where the joint distribution of the functions at any finite set of domain points is jointly Gaussian, i.e. at any particular domain point ($x \in X$), the GP posterior is Gaussian (i.e defined by a mean and variance), $\{\mu_x, \sigma_x\}$.

A Gaussian Process is itself parametrized by a mean function and a covariance function. In discussing GPs, we say that functions, defined over a domain X, are distributed according to a Gaussian Process with mean function μ, and covariance function k (Eq. 2). Functions are sampled (or 'realized') from the GP posterior by combining samples from the GP posterior at a set of domain 'test' points (the GP posterior at each point has a normal form). The mean and variance of the GP posterior at each test point is driven by two factors: First, a set of observed training data, $D = \{\{x_0, y_0\}...\{x_i, y_i\}\}$, and second, the covariance function, $k(x, x')$ that relates how 'close' two points in the domain are to each other – more technically, the degree to which the posterior predictions at two domain points are correlated.

When the covariance function is designed as a nonlinear distance map, the GP covariance function is referred to as a covariance *kernel*, and Gaussian Process inference is sometimes framed as a method for estimating the value of unobserved 'test' points based on observed 'training' points and a kernel that computes distances between training and test points. This view, perhaps more familiar to practicing data scientists, casts Gaussian Process inference in the framework of *supervised learning*. Gaussian Processes are widely used across a variety of real-world domains in part, for their ability to perform well in data-sparse applications Wang et al. [2005] and for the ready interpretation of their posterior as function estimates with uncertainty bounds.

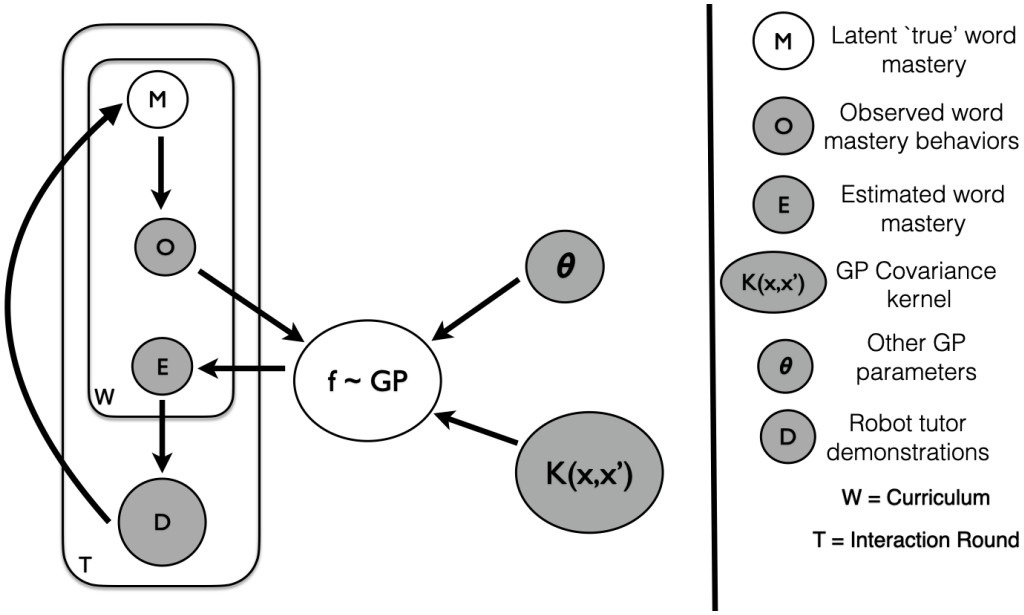


Figure 7: Plate notation model of a non-stationary Gaussian Process.

$$f(x) \sim GP(\mu(x), k(x, x')) \tag{2}$$

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 $\{x_i, y_i\}$. 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 training set contributes to posterior inference at each other output point. Once the domain and covariance kernel are defined, GP inference is fairly straightforward Rasmussen and Williams [2005]. The covariance kernel, therefore, defines the 'task' modeled by the GP output predictions.

4.1.2 *Gaussian Processes in Word Space: empirical implementation*

A Gaussian Process can flexibly represent a wide variety of domains and can be tailored by model designers to incorporate domain knowledge via the covariance kernel. In this section I discuss the implementation of this approach, applied to modeling student's literacy skills in separate game tasks, defined over a shared domain of English language *words*, called the Curriculum.

Because each game task may only be able to access a small amount of personalized data, the combined system leverages the shared domain to perform instance-weighted data *transfer* across game tasks, allowing a model targeting one particular literacy skill (e.g. decoding) to incorporate personalized data obtained from an interaction focusing on a different literacy skill (e.g. starting-sound identification). As previously discussed, this thesis also extends the approach to nonstationary environments, by augmenting

the conventional Gaussian Process with a 'continuous active training data management' protocol, that acts as a mirror to the active learning protocol pursued by the ContentModel. Instead of selecting domain points to *add* to the training set based on the GP posterior, the **CATDaM** protocol selects points already *in* the GP training set to *remove* (see Section. 4.2.2)

In the word-space domain, each input data point is a word from the Curriculum and a score from [-1, 1], where -1 represents complete lack of mastery, 1 represents full mastery, and 0 represents neutral mastery. For each point in an output set, the GP model computes a posterior mean and posterior variance $\{\mu_i, \sigma_i\}$, which, in this application, represent the posterior estimate that the student can apply the modeled skill to the output word (e.g. correctly decode the word 'FALL') and the uncertainty surrounding that estimate.

Under the framework of supervised learning, the GP prior mean function is conventionally set to 0 everywhere, leaving the covariance as the primary way for researchers to encode domain knowledge in the model 'design'. In fact, because the two task CognitiveModels differ only in their covariance kernels (they share all other hyperparameters), the covariance functions functionally *distinguish*, and therefore *define* the game task (with respect to each other). In other words, because the two CognitiveModels share an input space, mean function, and noise hyperparameters, the difference in their posterior estimates, if provided with the same training data, is *solely driven* by the differences in their covariance functions.

The training data take the form of a 'target word' from the curriculum and a score, representing an estimate of skill mastery applied to the target word, derived from gameplay. Scores range from [-1, 1], representing the student's demonstrated level of skill mastery applied to that word during gameplay, providing an intuitive scale for interpreting training data and, hence, the GP posterior.

A Gaussian Process is a regression model, and can therefore handle a continuous label space, but the design of the WordDecoder and WordBuilder game input gives only a discrete, binary signal: whether the student selected the correct decoded word (or 'starting-sound' letter) or not. To map from the binary signal of response correctness, we blend that information with continuous contextual features like timing. The final score (y_i) for a round Target word (i.e., a 'sample') (x_i), is derived by adding a timing adjustment, $p(t_d)$, to the 'correctness' binary variable (1 or 0), to correct for the possibility of guessing.

The timing adjustment is computed via a Latency Operating Characteristic (also known as Speed-Accuracy Trade-off) curve. While many mathematical models of the relationship between choice accuracy and response timing have been hypothesized including sequential sampling models (SSMs), and random walk (RW) models, we use a two-state mixture of random guesses (MRG) model (Lappin and Disch [1972]), with different parameters for each state affecting response-time and accuracy, to derive the final score.

If the selected word is correct, the timing adjustment is assessed 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 $p(t_d) = 0.5$. If the selected word is not the correct word, the penalty is assessed as $p(t_d) = 0.1 \cdot (MAX\_TIME - t_d)$, reflecting the idea that a longer time spent thinking about an incorrect answer demonstrates more potential mastery than a hastily entered guess (Heitz [2014]). The timing values are scaled differently in WordBuilder, but follow the same procedure. In both cases, the final timing adjustments are clamped to the range $[0.05, 1]$ before instance-weighted transfer.

4.1.3 *Designing WordDecoder and WordBuilder Covariance Functions: A Gaussian Process example in word-space*

The key difference between the two game CognitiveModels is their *covariance* kernels, which compute a distance metric between words in the Curriculum, bringing pairs of words 'closer' together when their task outputs (i.e., estimated student mastery) are more highly correlated. In WordDecoder, the covariance function is based on the cosine distance between the GloVe semantic word vectors (Pennington et al. [2014]) of each domain word, plus an additional term that increases the covariance between two words which share a final phonetic ending (i.e., when words are part of the same rhyme group) (Eq. 3). This combination of semantic and phonetic information has previously been validated by in-person student studies (Spaulding et al. [2018]), and was developed with input from external collaborators with expertise in early language and literacy skill development.

$$Cov_{rr}(\{w_i, w_j\}) = \nu[\alpha + \cos(GloVe(w_i), GloVe(w_j))], \tag{3}$$

where α = 1.0 iff w_i and w_j share a phonetic ending, and 0 otherwise. ν is a normalization constant.

WordBuilder's covariance function, reflecting the game's letter-based focus is based on *orthographic* information – information about the letters that make up a word's written form. The foundation of the covariance kernel is the Levenshtein distance, normalized over the combined length of the two words (Eq. 4). Levenshtein distance, also known as *minimum edit distance*, counts the number of single-letter additions, deletions, or substitutions (i.e. 'edit's) to convert one string into another. In essence, this kernel reflects the idea that words which are orthographically closer to each other are more likely to be mastered together, or not.

$$Cov_{wb}(\{w_i, w_j\}) = \nu[\alpha + Levenshtein(w_i, w_j)], \tag{4}$$

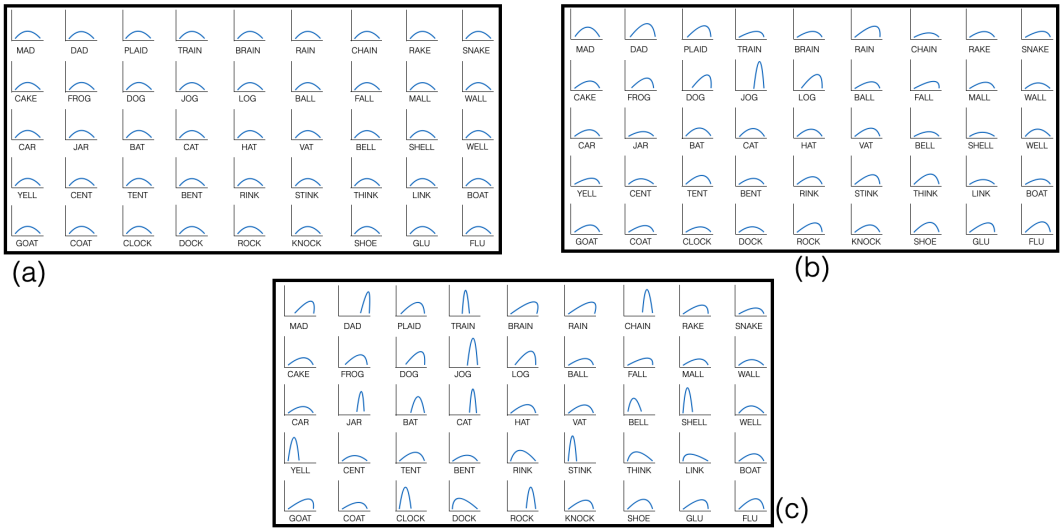


Figure 8: (a) Gaussian process prior, (b) GP posterior after one observation of student response, (c) GP posterior after several rounds of observation and inference. Mean estimates range from $[-1, 1]$ and variances from $[0, 1]$.

$\alpha = 1.0$ iff w_i and w_j share a beginning letter, and 0 otherwise. ν is a normalization constant.

In both tasks, the covariance kernels primarily function to help the GP CognitiveModels quickly generalize from observed samples to words not yet seen in the curriculum, improving the efficiency of model learning, as well as enabling the ContentModel to make personalized choices about which words from the curriculum to introduce in the games.

4.2 TRANSFERRABLE GAUSSIAN PROCESSES: INSTANCE-WEIGHTING BASED ON TASK COVARIANCE SIMILARITY

Both WordBuilder and WordDecoder models work well on their own as single-task models (see Section 5.2.1 for single-task baselines), but the broader goal of this project is to *transfer* observed training data from one game's CognitiveModel to a CognitiveModel targeting the other game, i.e., multitask personalization.

Both games' models share the same underlying Gaussian Process form, defined over a word space from the Curriculum. Unique to each game task is the *geometry* of this space, defined by the respective covariance kernels. How should we leverage this unified representation to transfer data from a source task to a target task (and back)? Because the two tasks are broadly related (i.e. both early literacy skills, and individual mastery likely correlated between them), we could consider simply adding all observed source task data to the target task training set. However, this approach ignores that some source task data points are more informative to the target task than others. In other words, the correlation of source task output with target task output *varies over the word space domain*. Moreover, we can use the definitions of the covariance kernels to compute a metric of task similarity at each domain

input point, which gives us a score of how similar the local geometries are for each task. We can interpret this instance-specific task similarity metric as a *transfer coefficient*. 'Instance-weighting' refers to a family of transfer learning methods for training a target task model on source task data, where the source-task data are re-weighted (a very simple form of task-transformation) before incorporation into the target-task training set (Pan and Yang [2009]). Thus we describe our transfer learning approach as an instance-weighting method, where each instance's transfer coefficient is derived from a similarity metric between each word's use in one game and its use in another (Eq. 5).

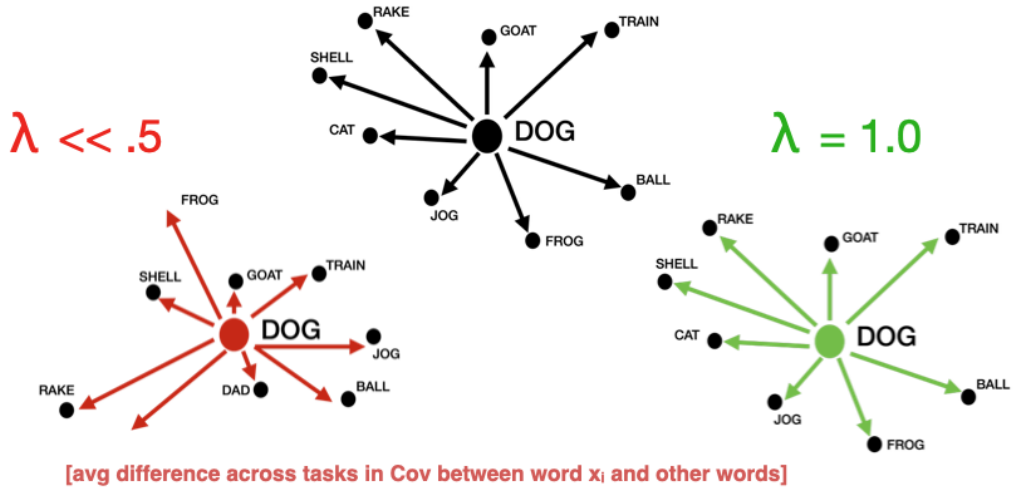
The covariance function of WordDecoder encodes the domain knowledge that words which share an ending are 'closer' to each other (i.e. if you can correctly decode DOG, you are more likely to be able to decode FROG) (Lenel and Cantor [1981]). Likewise, the covariance function of WordBuilder encodes the 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.

$$\lambda_i = \frac{\sum_{w \in W} 1 - ||Cov_s(x_i, w) - Cov_t(x_i, w)||}{|W|}. \tag{5}$$

A transfer coefficient of 1 indicates 'perfect' transfer, i.e., that instance word conveys the same information in both source and target tasks, whereas a transfer coefficient of 0 indicates that the source and target task are uninformative to each other, with respect to that instance word. To avoid undue complications in evaluating this method, we reweight data instances only once in our evaluations, from the originating source task to the target task. If the model switches tasks multiple times, previously transferred data is not re-weighted and re-transferred back to the original source-task model.

By design, the range of possible training data scores lies within [-1, 1], which, in addition to providing a natural interpretation of scores as 'mastery', also simplifies the instance-weighting transfer procedure. Because positive values are interpreted as positive mastery and negative values as lack of mastery, multiplying by the (positive) transfer coefficient λ can never change the



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

Figure 9: Instance-weighted transfer learning in theoretically ideal case of 'perfect' transfer (green) and under more realistic conditions (red).

sign of a training instance, i.e., a negative demonstration in WordDecoder remains a negative demonstration in WordBuilder.

4.2.1 *Non-overlapping Curriculum Transfer*

As discussed in Section 3.1.2, WordBuilder and WordDecoder draw their content from two separate Curriculums, hence the input spaces of their respective data instances do not exactly match. In this section, we present an expansion of the prior instance-weighting method that extends to 'partially heterogeneous' transfer scenarios in which the source and target tasks' input domain are not fully shared and have task-specific domains. Despite this challenge, our method still allows for instances of all source task words to be usefully transferred to the target task and vice versa.

Without loss of generality, we walk through the method on one-way transfer, from a single source task to a single target task (the actual game order doesn't matter). Each word in the source and target tasks' Curriculum can be labeled as belonging to one of 3 sets: S, words that only appear in the source task, O, overlapping words that appear in both, and T, words that appear only in the target task. The Curriculum for the source task, therefore, is O ∪ S and the Curriculum for the target task is O ∪ T.

Unifying the two partially overlapping domains, we construct a new "joint curriculum", composed of S ∪ O ∪ T. Over the joint curriculum, we re-compute the task-specific covariances between *all* words, leveraging the fact that the domain of the *covariance functions* is over words for which we can

compute both semantic vector representations and orthographic Levenshtein distance (see Section 4.1.3), a vastly larger set of words.

Then, we transfer as before, computing the transfer coefficient as the average difference in task covariance between the instance word and every other word in the joint curriculum, re-wighting the source instance by the coefficient, and incorporating the reweighted sample into the training data of the target task GP.

The generalization of this method introduces some new details. For instance, we call $O \cup T$ the 'core curriculum' because it represents the set of words upon which the target task will be evaluated: just the words that appear in the Target task game. Transferred source task instances in O change the model estimates directly as an observation for a specific word in the core curriculum, *just as if they had been directly observed in the target task*. Transferred source task instances in S are *not* part of the core curriculum, but still impact the model indirectly through their impact on posterior estimates of core curriculum words.

In essence, this method reduces the partially heterogeneous case to the homogeneous case, at the cost of some data (in S) having a less direct impact on the target task model, further underscoring the flexibility and extensibility of the word-space representation for adaptive cognitive modeling.

4.2.2 *Improving nonstationary GP modeling via Continual Active Training Data Management*

In this thesis, I extend multitask personalization to 'lifelong personalization' by bringing transferrable personalized models to *nonstationary* domains. This is primarily accomplished via a novel extension to the Gaussian Process modeling framework described thus far. In prior work (Spaulding et al. [2021b, 2018]), I noted that Gaussian Processes do not naturally have a sense of temporal data - if the model receives training points of (0,1) and (0,-1), the mean posterior prediction is indeed (0,0). But, counter to the intuitive interpretation of variance as uncertainty, the posterior distribution at 0 is not a *high variance* Gaussian (indicating a newly uncertain prediction), but rather a low-variance Gaussian (indicating certainty that the 'true' value lies in between the observed data points). Unlike other uncertainty-based estimation methods (e.g., Kalman filters) in which uncertainty is updated over time, Gaussian Processes lack a mechanism for increasing uncertainty around previously observed training data. The proposed solution, aimed at adapting Gaussian Processes to lifelong learning scenarios, is an active 'learning' protocol we call 'continual active training data management', or **CATDaM**.

In its simplest formulation, **CATDaM** consists of a data structure that organizes the observed training data temporally, and an active learning algorithm that marks 'stale' data points and removes them from the active training set. Much as the active learning method used by the ContentModel (described in Sec. 3.1.3) is closely tied to the tutoring agent's choice to observe

a student response, the active removal of training data followed by **CATDaM** is closely tied to the tutoring agent's *demonstrations*.

Demonstrations by the tutoring agent represent the most direct opportunity for the agent to influence *student* learning, by providing the correct response to a prompt word (as a player) and explaining out loud its reasoning to the student. As described in Sec. 3.1.3, the agent's decision to give a demonstration and the CURRICULUM word demonstrated are, in fact, coordinated by the STRATEGYMODEL, the CONTENTMODEL, and the COGNITIVEMODEL. The decision to take the DEMONSTRATE strategy action comes first, and then the CONTENTMODEL selects the word which the COGNITIVEMODEL is most confident the student has *not* mastered (i.e. has a negative posterior prediction for mastery).

A demonstration represents important contextual information for **CATDaM**! It signals that a student's mastery with respect to that domain point (i.e their mastery of the demonstrated word) may have shifted, and that prior observations of student performance may no longer reflect their current mastery. In order to address this potential distribution shift in student mastery, **CATDaM** marks prior observations of student response to that target word in the memory data structure and removes them from the active training set. Not only does the **CATDaM** protocol remove training data that may no longer reflect the current 'distribution', but it also has the additional advantage of directly increasing model uncertainty at the demonstrated domain point, signaling to the CONTENTMODEL that it is a good candidate for observing student performance at a future opportunity.

To make this more concrete, imagine a scenario in which the tutoring agent OBSERVE's a student incorrectly decoding the word "DOG" in the first round. The GP posterior updates with a lower estimate of decoding mastery for "DOG". In a subsequent round, the STRATEGYMODEL tells the tutoring agent to take the DEMONSTRATE strategy action, and because the COGNITIVEMODEL is confident the student has *not* mastered "DOG", the CONTENTMODEL selects it for demonstration. After the robot gives a demonstration of the correct decoding to the student, **CATDaM** looks back through this student's play history and removes prior observations of their decoding mastery for "DOG". Because the robot's latest demonstration may have substantially shifted student mastery, **CATDaM** assumes that these prior observations no longer reflect the best estimate of their future ability. The COGNITIVEMODEL is retrained, and the subsequent posterior estimate is now more uncertain about the student's ability to decode "DOG" (though other word decoding observations, – "LOG", "FROG", etc. – still influence this posterior estimate). Because the uncertainty surrounding this estimate is now boosted, "DOG" once again becomes a good candidate for selection by the the CONTENTMODEL, when the STRATEGYMODEL tells the tutoring agent to take the OBSERVE strategy. This dynamic of a tutoring agent's OBSERVE-DEMONSTRATE-OBSERVE behavior pattern is key to effective tutoring and arises naturally from the interplay between the CONTENTMODEL, STRATEGYMODEL, COGNITIVEMODEL, and **CATDaM**.

5

GAUSSIAN PROCESSES IN SIMULATION

5.1 EVALUATING LIFELONG GAUSSIAN PROCESSES AND MULTITASK TRANSFER IN SIMULATION

Over the past year, I have conducted several simulation-based studies of multitask and lifelong personalization, published in detail in Spaulding et al. [2021b] and Spaulding et al. [2021a]. In this section I will briefly report on the results of these studies, evaluating the effect of adding **CATDaM** to a CognitiveModel in simulation experiments with model students.

Although uncommon, it is by no means a new idea within HRI to simulate human data to evaluate robot behavior, models or algorithms under gentler (and more repeatable) conditions. The benefits of this practice are most clearly articulated in a paper that describes the "Oz of Wizard" paradigm, inverting the better-known "Wizard of Oz" paradigm in which real humans interact with a robot whose behavior is actually produced by a human (Steinfeld et al. [2009]). Under the Oz of Wizard paradigm, real robot behavior is evaluated against humans whose behavior is actually produced by a computer, i.e. *simulations* of human behavior. "Oz of Wizard" experiments involving 'simulated' students are rarely publicized, despite the widespread use of simulators in other areas of robotics (e.g., Sim2Real motion planning or task learning). In part this is because real student behavior is not easy to simulate. Real students act unpredictably, capriciously, and in ways that even the students themselves struggle to articulate.

In many fields of engineering where the 'actual' live test of a system is expensive, overly time-consuming, or carries substantial risk, simulation studies are considered *de rigeur*. Despite a simulation fidelity gap larger than many physical environment simulations, I argue that simulated student evaluations can advance research in long-term human-robot interactions by providing a more principled starting point for systems prior to conducting long-term in-person studies. For instance, studies on simulated student data can confirm that modeling algorithms perform as expected on simplified data distributions. Simulated student data can also help algorithm designers tune hyperparameters to useful values or establish reasonable performance baselines without having to conduct pilot tests on live students. Simulation studies can also allow for many different comparisons to be made in parallel, whereas human-subjects studies are more typically tightly controlled owing to the generally small number of participants, which has the unfortunate side effect of limiting the number of hypotheses that can be evaluated. We believe that the use of simulated student tests should not be considered a *substitute* for an in-person evaluation, but rather an important and insightful part

of the system implementation and preparation before a study of in-person long-term interaction is launched.

In a 2021 review published in the Proceedings of the National Academy of Sciences, roboticists highlighted HRI as an area where simulation has great potential, but also faces many challenges.

> Development of simulation tools that better represent the psychosocial nature of HRI and enable a common operating 'picture' of possible solution sets for decision making may ... establish a baseline for more effective collaboration.....Creating [simulated human] avatars is as difficult as humans are diverse, each person a unique and complex web of intertwined physical, social, emotional, cognitive, and psychological threads....Numerous questions remain unanswered in relation to abstracting in mathematical models the psychological underpinnings that trigger in humans states of anxiety, fear, comfort, stress, etc. In this context, the ability to control and display emotions in [simulated human] avatars represents a prerequisite for endowing smart robots with a sense of empathy in their interaction with humans. (Choi et al. [2021])

5.1.1 *Simulated Students: pre-study evaluation for long-term HRI systems*

In this section I describe simulated student performance data, used to analyze the effects of multitask personalization through cross-task model transfer and 'lifelong personalization' extensions via **CATDaM**. In subsequent sections I outline implementations of two classes of simulated 'students' (referred to as 'simple' and 'dynamic' *SimStudent* s), describe the theoretical assumptions on which these simulations are based, and discuss the implications of subsequent simulation experiments. These simulation studies were conducted when the WordDecoder game was called RhymeRacer and had a slightly different task design, asking students to select a Target word that rhymed with a Prompt word shown at the start of each round (see Section 3.1.2.3).

SimStudent

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 data 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'.

5.1.1.1 *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 (Hixon et al. [2011]), 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 (Høien et al. [1995]). Each phonological mastery is initialized uniformly at random in the range $[0 - 1]$. After 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 average, $\sigma = .1$) 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.

phonological mastery
word mastery

5.1.1.2 *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 (Baker et al. [2008]), 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*. These values fall within the range of empirically observed rates of student 'guess' and 'slip' behaviors.

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 $10 - \mathtt{MasteryDecile}$, so that lower levels of mastery correspond to longer timing components.

5.1.1.3 *Dynamic Students*

The dynamic SimStudent largely keeps the same implementation as the simple SimStudent, and extends it by adding a *learning rate*, r, and a *learning gain* parameter, g. Whenever the tutoring agent gives a demonstration, the learning rate parameter determines the probability that the student's mastery increases, simulating student learning. The magnitude of the score rise in the student's underlying word mastery is set by the learning gain parameter (word mastery is capped at 1, and further student learning from tutor demonstrations has no effect). In the experiments reported here, the learning rate was set to .66 and the learning gain was set to .50 (so if mastery were at its lowest possible value, two successful lessons would be sufficient to boost mastery to halfway, and 4 successful lessons would boost mastery to its highest value). Other than the probabilistic shift in word mastery in response to tutor demonstrations, the dynamic SimStudent's word mastery and performance data are simulated identically to the simple SimStudent.

5.1.2 *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 CognitiveModel then has the task of recreating or estimating the true mastery from the derived SimStudent game performance data.

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.

The primary questions we were interested in answering with this work were fundamental measures of transfer learning systems: *viability*, *proficiency*, and *efficiency*. In other words, **(1)** Viability: does incorporating source task data improve target task performance at all, or do we find that source task data is worse than no data, i.e., negative transfer? **(2)** Proficiency: Does a target task model trained on source and target task data perform better than a target task model trained on the same amount of *total* data, exclusively from the target task? **(3)** Efficiency: Does a target task model trained on source and target task data perform better than a target task model trained on the same amount of *target task* data only?

negative transfer

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

Human-centered Hypotheses

1 Viability: (*Are the tasks compatible enough to avoid negative transfer?*).

2 Proficiency: (*Does transfer improve the performance of the final model?*).

3 Efficiency: (*Does transfer improve model performance during training?*).

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.

F-1 classification score

In these simulation studies, we used the *F-1 classification score*, which combines precision and recall, as our primary model evaluation metric. The classification task is whether the model correctly predicts the *sign* (i.e. positive or negative) of the SimStudent's true word mastery. While this may seem a coarse metric for simulated study — we could, for instance, look at L1 or L2 regression loss — the sign of the word mastery is the primary determinant of the correctness of the student's response (guesses and slips notwithstanding). In a study with real students, we do not have access to a numerical form of a student's 'true' mastery; student models are evaluated based on their ability to predict student's actual response behaviors. Therefore, in the spirit of keeping our simulation as close as possible to human subject study, we focus our evaluation on the same metric: binary classification of student mastery with respect to individual curricular components.

Each figure below shows the results of the average of 20 "rollouts" of 60 'samples' for each of three classes of model: RhymeRacer single task, WordBuilder single task, and a transfer model (color shading indicates standard error of the mean). At the start of each rollout, a new SimStudent (with newly randomized word mastery) is created to represent a unique student. Each rollout consisted of 60 samples, intended to mirror the structure of many common studies of long-term interactions – 4 interaction 'sessions' each of which provided 15 useful samples (roughly in line with the actual number of samples collected in live human-robot experiments reported in Spaulding and Breazeal [2019]). Within each rollout, the transfer model alternates tasks at the start of each 'session', i.e. after 15, 30, and 45 samples respectively. Within each rollout, 'samples' represent opportunities for the tutoring agent to OBSERVE students mastery via game performance.

In live gameplay, the StrategyModel determines whether the tutoring agent DEMONSTRATEs or OBSERVEs. For our simulation study, we adopted a simple rule-based StrategyModel: the robot chooses to DEMONSTRATE after a fixed number of samples and OBSERVE otherwise. In the case of a typical 60 sample rollout, the tutoring agent DEMONSTRATES twice after

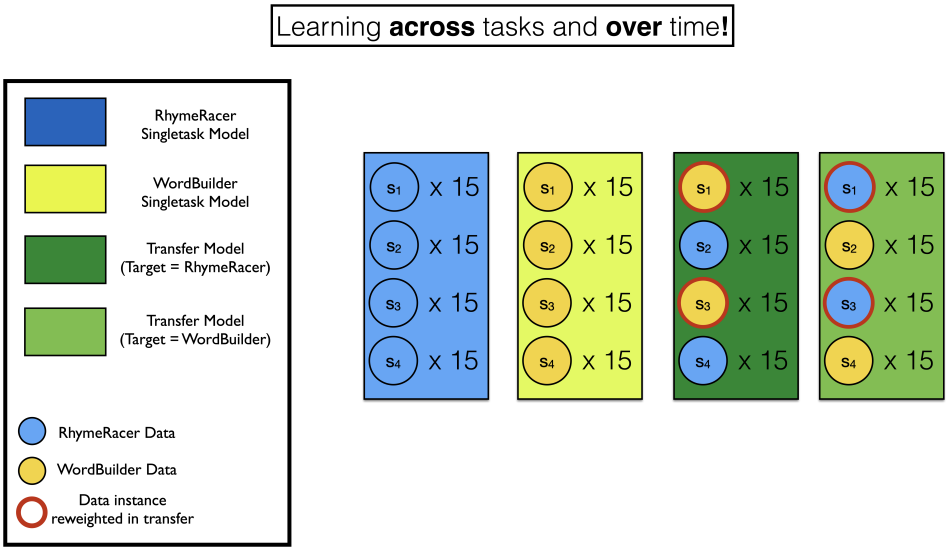


Figure 10: Visual depiction of training data for single- and multi-task student models. Blue and Yellow rectangles and circles indicate models and data instances from RhymeRacer and WordBuilder. Red rings indicate data has been re-weighted from its originating source task to a new target task

every 3 samples it OBSERVEs, starting after the first 9 samples. So in a 60 sample rollout, the student receives 34 'demonstrations' from the robot (not all of which result in successful learning), 2 each after 9,12,15... samples. In live gameplay, the StrategyModel determines whether the tutoring agent DEMONSTRATEs or OBSERVEs. For this simulation study, I adopted a simple rule-based StrategyModel: the robot chooses to DEMONSTRATE after a fixed number of samples and OBSERVE otherwise. In the case of a typical 60 sample rollout, the tutoring agent DEMONSTRATES twice after every 3 samples it OBSERVEs, starting after the first 9 samples. So in a 60 sample rollout, the student receives 34 'demonstrations' from the robot (not all of which result in successful learning), 2 each after 9,12,15... samples. Figure 10 shows the structure of the training data for each class of model graphically.

Throughout these simulation experiments, we strove 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 Spaulding et al. [2018]. 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 multitask 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 on a 2017 Macbook Pro computer with a 2.9 GHz Quad-Core Intel Core i7 processor.

5.2 GP SIMULATION RESULTS AND DISCUSSION FOR FUTURE RESEARCH

5.2.1 *Multitask Personalization with Stationary Students*

The results in this section were previously reported in (Spaulding et al. [2021b] and Spaulding et al. [2021a]). Here, we give further context for these results and provide new supporting evidence to support their conclusions, showing that the effect persists even when the task order is reversed.

Figure 11 shows the results of the rollouts when RhymeRacer is the starting task, though the same trend holds when task order is switched.

Both single-task models learn good representations of their respective game tasks over 60 samples, consistent with prior experimental results (Spaulding and Breazeal [2019]), suggesting that our simulation settings are reasonably implemented, giving confidence in further results not yet evaluated in an experimental setting with human students.

The transfer model data is depicted in two separate representations, each of which is better suited to answering different questions. The 'continuous' representation (left) shows the transfer model data as a single rollout of 60 samples, with each session segment colored to show transfer. This representation is best suited for exploring questions of final proficiency – how well do transfer models trained on a mix of source and target task data compare to single-task models trained on the same amount of data exclusively from the target task? The 'discontinuous' representation (right, both figures) shows the transfer model data split into discrete session segments, with their position on the x-axis determined by the amount of *target* task data. This representation is best suited for exploring questions of model efficiency – how well do transfer models trained on a mix of source and target task data compare to single-task models trained on the same amount of data exclusively from the target task?

Figure 11 shows that initial transfer from RhymeRacer to WordBuilder is substantial and positive, and that a WordBuilder model trained on prior data from RhymeRacer outperforms a single-task WordBuilder model, particularly during crucial early interaction rounds. Figure 12 shows that the effect remains consistent when the task order is reversed (i.e. when the task sequence starts with WordBuilder). In this case, we can see that transfer from WordBuilder to RhymeRacer boosts initial performance, but that subsequent transfer effects are less impactful as more target task data is gathered, suggesting that the benefits of task transfer may not be symmetric (i.e., the benefit of transferring RhymeRacer data to WordBuilder may not be equal to the benefit of transferring data from WordBuilder to RhymeRacer).

Overall, these results from the simplified simulation environment paint a compelling enough picture to merit further investigation of multitask personalization in the nonstationary setting. Positive transfer is evident in both directions, and there is strong evidence that multi-task personalization is most impactful in crucial early phases of an interaction, before a model

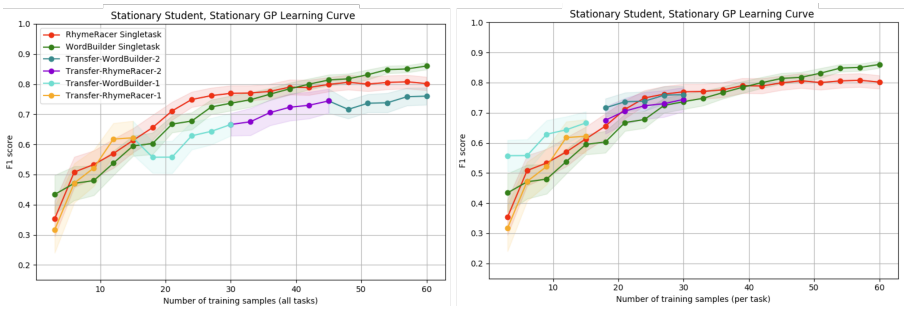


Figure 11: Simple 'Proficiency' and 'Efficiency' evaluation of multi-task vs. single-task personalized models when RhymeRacer is the first task. The transfer model trades off final classifier accuracy for multi-task generality and meets or exceeds single-task model performance with equal amounts of target task data

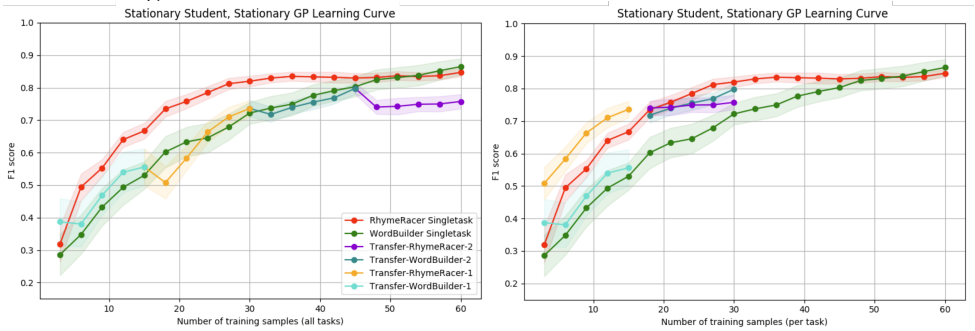


Figure 12: Simple 'Proficiency' and 'Efficiency' evaluation of multi-task vs. single-task personalized models when WordBuilder is the first task. The general trend is consistent with the results when RhymeRacer is first, indicating that the results are stable independent of task order

has an opportunity to acquire significant target-task training data. In short, simulation experiments with stationary students indicate that multitask personalization can improve the efficiency of target task model learning, and that this effect is most pronounced within the first few samples collected during an interaction. This is a critical step towards reducing the problem of cold-start learning in interactive machine learning.

5.2.2 *Lifelong Personalization with Dynamic Students: Effects on Model Proficiency and Data Efficiency*

Now we turn our attention to evaluating qualities of multitask personalization in a more complex, nonstationary simulation scenario that incorporates the effects of a tutoring agent's actions on dynamic (i.e., learning) students. In these evaluations, results for all models were derived over the same simulation timeline of 60 samples, even though in studies with real students, there is often a trade-off between opportunities for the tutoring agent to respond ('demonstrations') and the student to respond ('samples'). Because we are primarily interested in understanding data and performance trade-offs between different kinds of computational models, we chose to evaluate them over consistent data sample timelines. Even though the models evaluated with a dynamic student incorporate demonstrations and student learning and models evaluated on static students do not, we evaluate them both with

respect to the same 60 sample timeline. We also provide new results from adding 'continuous active training data management" (**CATDaM**) to the GP CognitiveModel for 'lifelong' learning, and present evidence that including **CATDaM** can improve both model performance and student learning in nonstationary scenarios.

First, we show what happens when we apply the original, static Gaussian Process model (without **CATDaM**) directly to a nonstationary simulation with agent demonstrations and dynamic students.

Figure 13 compares static single-task and transfer models evaluated in two different scenarios. On the left, we have the same experimental conditions as Figure 11, in which underlying student performance is derived from a static SimStudent and there are no demonstrations to promote student learning. On the right, the modeling GPs take the same modeling approach, but the underling student performance data is derived from dynamic SimStudents. Demonstrations from the tutoring agent slowly cause shifts in underlying student mastery. This shows the expected performance gap from modeling a dynamic target using a non-stationary modeling approach. Even under these more challenging conditions, the GP modeling framework can still learn a passable student model, but on the right, we see that the relative impact of 'stale' data and student mastery shift impede performance. Across all classes, final model proficiency stabilizes at an F1-score of [.74-.79], compared to [.84-.87] when modeling static students, a drop of 10 percentage points. The final proficiency of the multitask model also declines across both tasks, though the performance loss is less than in the single-task case. Despite this hit to overall proficiency, the most notable trends of the multitask transfer model, positive transfer and early-sample efficiency gains, remain.

Figure 14 shows the benefits of incorporating continual active training data management (**CATDaM**) into Gaussian Process student models, comparing static single-task and transfer models (on the left) to lifelong (i.e. uses **CATDaM**) single-task and transfer models on the right. For both classes of model, underlying student performance is derived from a dynamic SimStudent that receives demonstrations.

On the left, we see the same general performance trend as the right side of Figure 13. The static GP model learns a decently performant model of the dynamic student, but student learning causes both single-task and multitask models to quickly hit a lower performance ceiling than in the static-student-static-GP case. Without accounting for shift in dynamic student mastery, the learning curve for static-GP models flattens and even declines slightly. On the right, it continues to rise throughout the full 60 sample rollout, hitting basically the same level of performance as the 'static-student-static-GP' case from Figure 13.

To summarize these results: when we increase the complexity and realism of the simulation environment by adding in a non-stationary SimStudent and tutor demonstrations, stationary GP models perform about 8-10 percentage points worse (10-15%). Augmenting the GP model with **CATDaM** helps the Gaussian Process to better model the non-stationary effects of

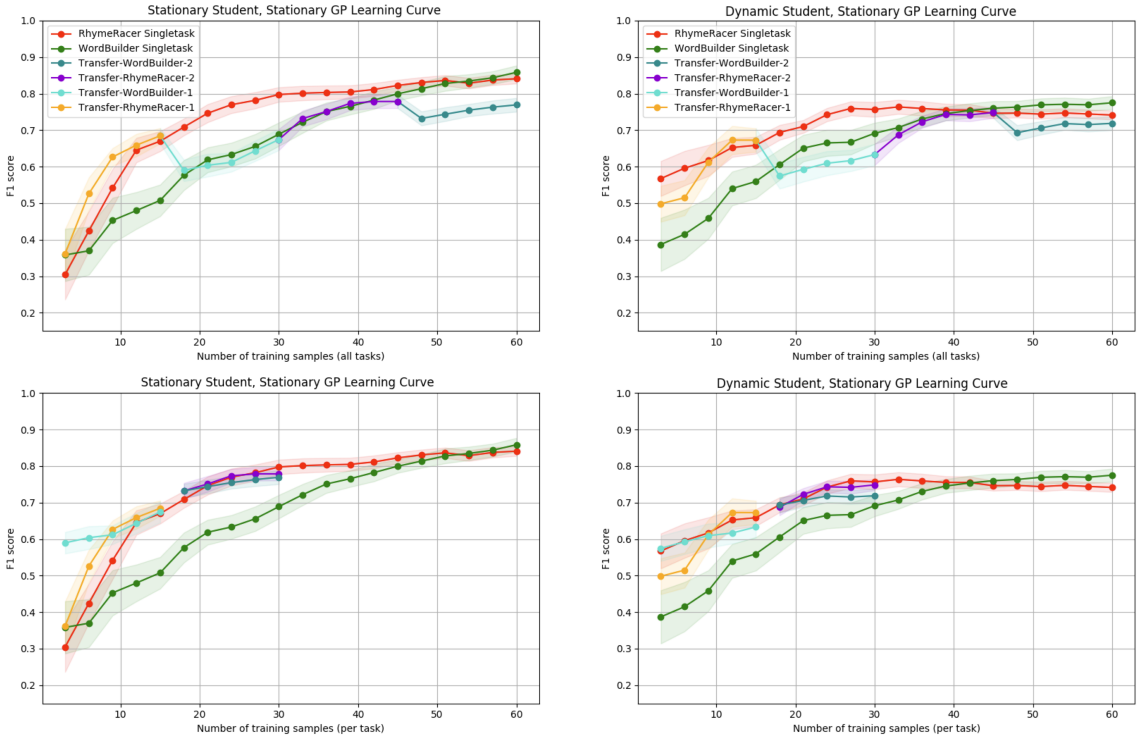


Figure 13: Static-model-static-student performance results(left) vs. static-model-dynamic-student performance results (right). Static models can learn a decent model but suffer a drop in final proficiency. Efficiency benefits of multitask model are undiminished.

tutor demonstrations, and performance performs as well as in a more complex, nonstationary environment as a stationary model does in a stationary environment.

And, while non-stationarity lowers the final proficiency of static GP models, it does not appear to materially impact the *efficiency* results from multitask transfer. Nor are efficiency results clearly impacted when GP model proficiency rises as a result of incorporating **CATDaM**. This result indicates that the efficiency benefits of a multitask personalization approach are independent of the *proficiency* benefits of a continual learning approach.

5.2.3 *GP Modeling of Dynamic Students: Effects on Student Learning*

In addition to enabling more sophisticated evaluation of proficiency and efficiency of personalized model learning, by integrating tutoring agent actions and dynamic student learning into our simulation experiments, we can also study the effect of **CATDaM** on *student learning*, the increase in mastery due to the tutoring agent demonstrations. We quantify these results by calculating the number of 'newly mastered' words (mastery went from negative to positive) for each model type over rollouts guided by both static and dynamic GPs. Figure 15 shows that SimStudents in the dynamic GP case learned 5 more words on average, compared to students in the static GP rollouts. We hypothesize this result is due to the dynamic GP picking

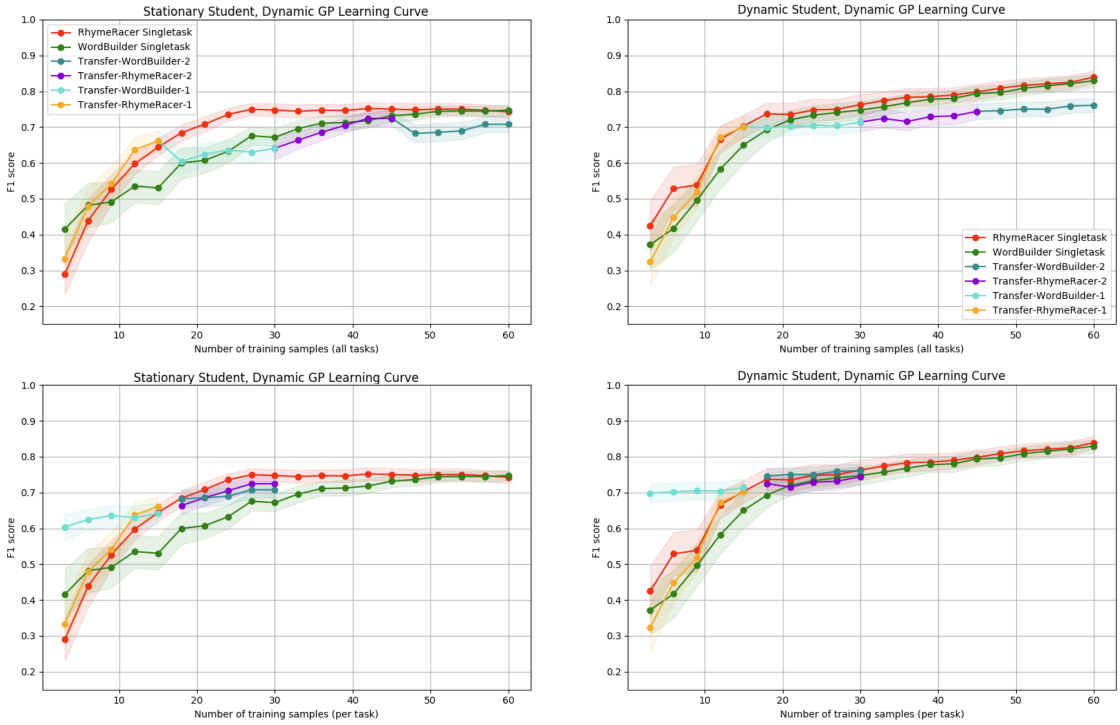


Figure 14: Static-model-dynamic-student performance results(left) vs. dynamic-model-dynamic-student performance results (right). Adding **CATDaM** to GP models improves modeling performance in nonstationary environments, while preserving efficiency benefits of multitask personalization.

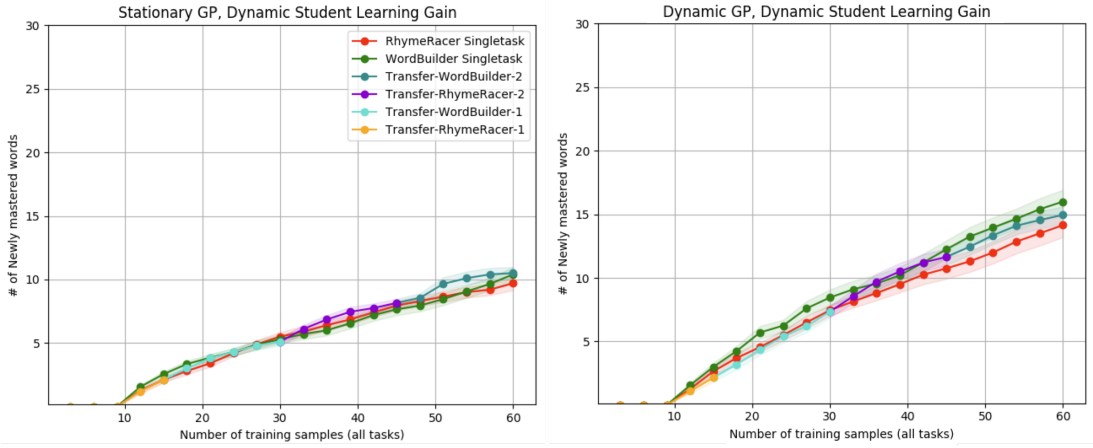


Figure 15: Student learning gains under static-model-dynamic-student (left) and dynamic-model-dynamic-student (right) simulations. Students tutored by a dynamic GP model mastered nearly 50% more words.

'better' words to demonstrate, on account of a more up-to-date estimate of word uncertainty enabled by **CATDaM**.

5.2.4 *Discussion of simulation results*

Throughout these experiments, we strove to carefully contextualize the results as supporting evidence in support of future in-person studies with human students. There is truly no substitute for actual human experimental

data. At the same time, we think that these results provide confidence to proceed with live student studies, and are demonstrative of the kinds of benefits for long-term HRI that can come from simulation analysis. In advocating for researchers to evaluate their systems in the real world, Rodney Brooks famously quipped "simulations are doomed to succeed" (Brooks and Mataric [1993]). 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) (Griffith et al. [2013]). 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.

Extending our evaluation of multitask learning to more realistic non-stationary domains lends further confidence that simulation results will extend to live long-term studies with students, but the addition of robot actions (demonstrations/observations) and stochastic student learning updates also allow us to analyze estimated student learning gains in simulation. We found that in simulated interaction with a tutoring agent using a **CATDaM**-enabled model leads to a simulated learning gain of almost 50% more new words mastered, compared to an agent using a static CognitiveModel. These results are consistent over both single-task and multi-task models, and are robust to task order in the multitask case.

We also find evidence that the extension of our modeling approach to nonstationary domains does not substantially alter the positive transfer benefits of data efficiency and cold-start avoidance previously observed in evaluating multitask personalization. In other words, adopting a continual learning approach appears to be *complementary* to a multitask personalization approach. Finally, we show that adopting a continual learning approach to dynamic student modeling also has benefits for *student learning* in addition to model learning.

These simulation results highlight a number of hypotheses of particular interest (based on the observed effects in simulation) for further evaluation via human subjects study.

First, is transfer symmetric? In simulation, we observed that the benefit of transferring RhymeRacer data to WordBuilder may not be equal to the benefit of transferring data from WordBuilder to RhymeRacer. Second, how meaningful is the effect of **CATDaM** on student word learning? Figure 15 showed a substantial difference in student word learning in simulation, but student learning is quite difficult to accurately model in simulation. Finally, does **CATDaM** improve overall model performance without changing the impact of model transfer? In discussion with other researchers, we agreed this would be a fantastic main result. But is this result merely an artifact of favorable simulation design? I.e. was this work "doomed to success"?

One of the primary benefits of simulation study prior to human subjects work is that it can highlight particular research questions for emphasis. In planning the work presented in the next chapter, we made design tweaks to insure we could appropriately answer these three questions in the course of the human subjects study.

6

LIFELONG PERSONALIZATION: STUDENT AND MODEL LEARNING IN HUMAN SUBJECTS

6.1 HUMAN SUBJECT STUDY DESIGN

Prior simulation results suggest that combining multitask personalization and continual learning into a 'lifelong personalization' approach appears to benefit both the data efficiency of model learning, the final proficiency of learned student models, and the amount of student learning gain. The simulation experiments provide useful insight as technical validation in advance of a long-term in-person study, and have also proven useful in discussions with schools and other institutional partners in preparation for research engagement as a scientific partner in long-term HRI research.

These results from simulation bring us one step closer, providing compelling evidence that combining continual learning and multitask personalization can be a successful path towards truly lifelong personalized companions, though more research is needed to confirm these effects in studies with real students.

In order to confirm these results, I conducted a human subjects study to evaluate the multitask personalization approach and the unified system. Due to the uncertainty of the ongoing COVID-19 pandemic which restricted us from using elementary school-age early readers as participants, I redesigned the games for a human subjects evaluation with MIT undergraduates, changing the task context to second-language learning, but with a similar focus on literacy skills. Changing our study population provided more reliable scheduling and participant count, but also posed new operational challenges for task and experiment design (see Section 3.1.2.1 for task translation details).

6.2 PRIMARY RESEARCH QUESTIONS

The main research goals of this study are the same as those explored in the simulation experiments: **(1)** determining the core viability of multitask personalization via student model transfer (i.e. is positive transfer possible?), **(2)** studying the impact of multitask personalization on the efficiency and proficiency of learned personalized models, and **(3)** studying the impact of combining continual learning methods (in the form of **CATDaM**) with multitask personalization.

In addition to these *system-centric* (i.e. algorithmic) research questions, we also propose to explore *human-centric* research questions: **(1)** How does student engagement change over the course of a long-term study with multitask personalization? **(2)** How do multitask personalization and lifelong personal-

ization affect student learning (as measured by posttest and within-session assessment data)?

Computational Hypotheses

C1 Lifelong Personalization does not reduce model performance with equivalent data (*No Negative Transfer*).

C2 Lifelong Personalization helps models achieve better performance with less data (*Efficiency Effect*).

C3 Lifelong Personalization improves final model performance (*Proficiency Effect*).

C4 Continual Learning improves model performance (*Continual Effect*).

Human-centered Hypotheses

H1 Student engagement is affected by Personalization condition (*Engagement Effect*).

H2 Student posttest learning is affected by Personalization condition (*Student Posttest Effect*).

H3 Student assessment learning is affected by Personalization condition (*Student Assessment Effect*).

In advance of data collection, I pre-registered the study plan on AsPredicted, a platform that allows researches to commit to investigating specific research questions *a priori* rather than generating 'hypotheses' after the results are known. Hypotheses C1, C2, C3, and C4 were included in the pre-registration form.

6.3 STUDY SCHEDULE AND EXPERIMENTAL DESIGN

6.3.1 *Experimental Conditions*

I conducted a 4-session study, with a total of 72 participants randomized to be in 1 of 4 conditions. Each participant experienced both games (WORDDECODER and WORDBUILDER) in an alternating sequence. Participants in Conditions A (n=19) and B (n=19) played the game with a robotic agent that personalizes using both multitask and continual learning (**CATDaM**), they differ in the order in which the games are played (see Figure 16). In Condition C (n=17), participants interacted with a robot using multitask personalization, but not continual learning (i.e. data is transferred across tasks, but **CATDaM** does not actively prune stale training data). Condition D (n=17) is a control condition in which students interact with the robot, without either multitask personalization *or* **CATDaM**. In other words, personalization only happens *within* each session, and the data is not used to personalize subsequent sessions at all.

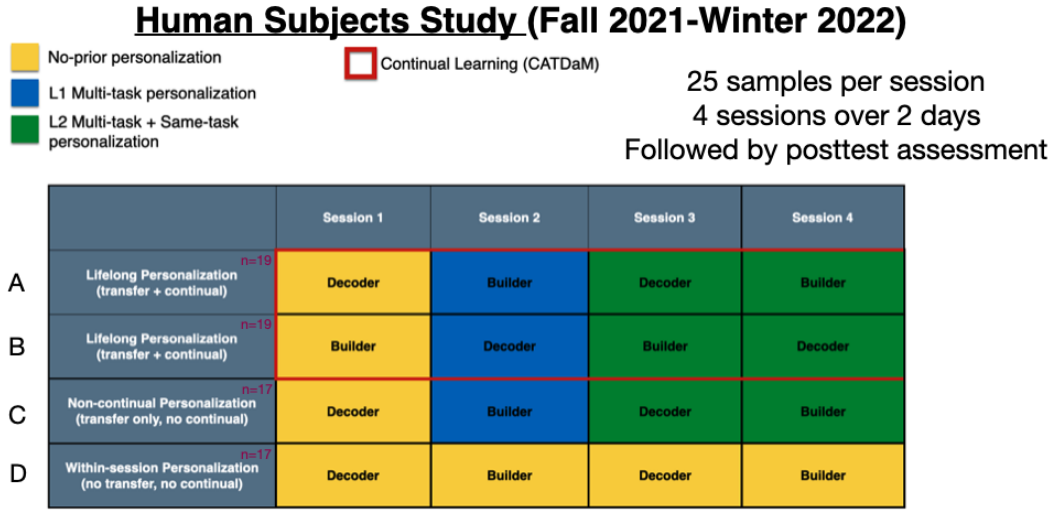


Figure 16: 4 session timeline of games, each played 2 times. Each session consists of an Interaction Phase and an Assessment Phase. Letter, word, and engagement post-test assessments followed the final session.

Comparing conditions A and B allows us to evaluate the effect of multitask personalization by comparing average model performance from each game at different levels of personalization ('No prior personalization', 'L1' Source-task transfer, 'L2' Source-task transfer and Target-task data, etc.), providing an answer to C1. For example, we can compare the Condition A model accuracy on WordDecoder (Session 1, with no prior personalization) to the Condition B model accuracy on WordDecoder (Session 2, with only WordBuilder data transferred in), to isolate the effect of multitask transfer on model performance at different checkpoints (see Figure 22).

Comparing conditions A and C allows us to isolate the effect of continual learning by comparing average model performance and student learning across conditions. Comparing Conditions A and D (and C and D) allows us to isolate the effects of lifelong personalization and multitask transfer compared to a common baseline of students who interact with a robot that adaptively personalizes only within single sessions.

6.3.2 *Study Protocol*

6.3.2.1 *Participant Recruitment*

We recruited 84 undergraduate and graduate students from MIT dorm mailing lists. These subject enrollment numbers align with our previous experience conducting similar sized studies, as well as with an independent power analysis based on estimated effect sizes and variation (Figure 17). Participants were compensated $25 for completing the study. Participants were excluded from the study if they had any prior experience studying Russian or more than incidental exposure to Russian.

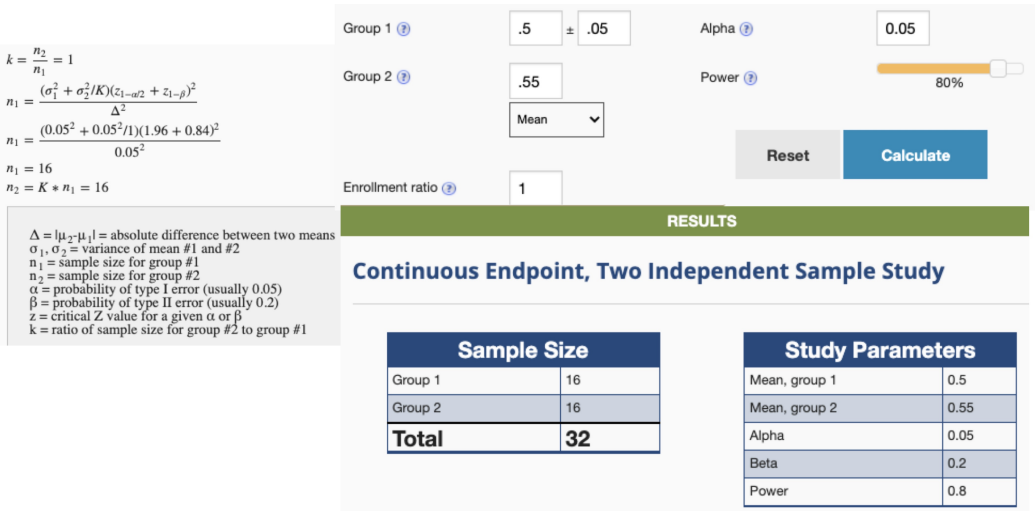


Figure 17: Independent study power analysis to determine minimum condition sample size

6.3.2.2 *Procedure*

Over the course of the study, each participant visited the lab twice within one week and during each visit engaged in two gameplay sessions with Jibo. During participants' first visit they were introduced to the robot and given a tutorial on how to play the games. After the final session, participants completed three post-test assessments: a word posttest, a letter posttest, and a survey task to report their self-assessed engagement.

Each experimental session took approximately 15 minutes to complete, with 10 minutes (15 student responses) dedicated to fully interactive gameplay and 5 minutes (10 student responses) dedicated to an 'assessment' phase. During the *Interaction Phase*, the robot played a fully active role in the game, selecting words and giving feedback to the child. During the *Assessment Phase*, the robot is a silent partner who does not comment. The ContentModel is also deactivated during the Assessment Phase and words are selected randomly from the Curriculum. Finally, the tablet does not provide the usual reinforcement signs of correct and incorrect answers. The Assessment Phase is primarily used as a way to collect session-level 'ground truth' against which we can compare model predictions as well as session-level indicators of student learning.

Interaction Phase

Assessment Phase

6.3.2.3 *Dependent Measures*

PARTICIPANT PERFORMANCE DATA When the robot agent's StrategyModel selects the OBSERVE objective, students have an opportunity to demonstrate their knowledge by providing an answer in the game. Students' responses were logged during all gameplay sessions. Computational model performance was assessed by the accuracy of predictions of future student responses (see Section 6.4). Student response

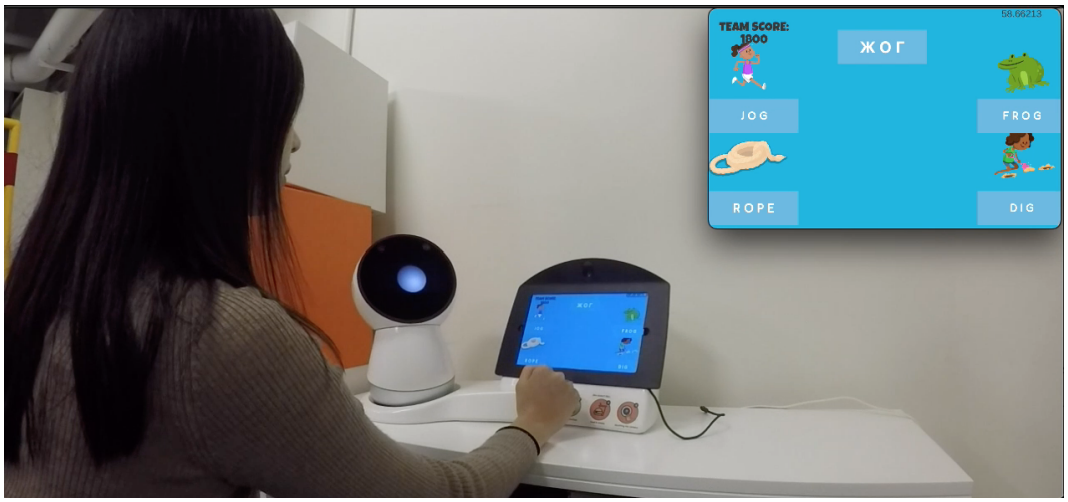


Figure 18: **GoPro camera view for human subjects study.** Setup includes front-facing camera within Station, microphones inside Jibo robot, Android tablet screen-recording, and a GoPro camera.

accuracy during the Assessment Phase was also used to assess student learning over time.

LETTER LEARNING POSTTEST After completing the final session, participants completed a worksheet showing each of the 31 letters used in the study, on which students were asked to "mark the letter or letters in English that best match the sound this Russian letter makes". Student responses were manually scored by the experimenters. Originally, students completed this worksheet as both a pre-test and a post-test (in order to compute normalized learning game and counter-balance experimental conditions), but we discontinued the pre-test component after approximately a dozen participants due to every participant claiming they had no prior experience with Russian and would have merely guessed at each letter on the pre-test.

WORD LEARNING POSTTEST After completing the final session and the letter posttest, participants completed a "word" posttest activity, a modified version of WORDDECODER that shows the letters of fully translated Russian words instead of English transliterations. The tablet pronounces each word out loud, and students select one of four graphics indicating which word they think the Russian word corresponds to. As in the Assessment Phase, Jibo is a silent partner who does not play or move during this activity, words are selected in sequence to cover the entire task CURRICULUM, and the tablet does not provide overt indications of answer correctness. The posttest is designed to assess students' word learning, even though that is not the direct focus of one of the game tasks.

ENGAGEMENT AND FEEDBACK SURVEY After completing the Word Learning Posttest, participants filled out a short four-question Likert survey indicating the degree to which they agreed with statements regarding the

effect of the game design and robot behaviors on learning and the degree to which they felt engaged by each game.

6.4 DATA COLLECTION AND ANALYSIS

We collected data from 84 participants, and ended up with 72 usable data points. Participants were most commonly excluded from final analysis due to technical difficulties or not returning for their second session.

For each participant, we recorded synchronized logs of their gameplay activity, the robot's actions, and front-facing camera footage. We also collected non-synchronized over-shoulder GoPro camera footage (see Figure 18), a screen recording of the tablet game display, word and letter posttest evaluations, and the engagement posttest survey.

EVALUATING STUDENT MODEL ACCURACY: WALK-FORWARD EVALUATION In the simulation studies published so far, we were able to directly compare the predictions of the trained personalized models against the "ground truth" mastery of the simulated students. However, when evaluating personalized models trained on real students, it is not possible to directly compare against a student's "ground truth" mastery. In the past, researchers have typically adopted one of two approaches: conducting a comprehensive assessment of each student using a validated assessment tool, or collecting more interaction data from a final 'post-test' session to hold out as a test set.

To answer the most pressing research questions regarding lifelong personalization, we are not only interested in how well final models perform, but also in the incremental performance of models at various, potentially early, stages of training — as observed in simulation, benefits of personalized task transfer seem most likely to be found in early stages of task modeling. Collecting additional interaction data to use as a model test set would help evaluate the final model proficiency, but would not be a valid set to compare against earlier instantiations of the personalized models because this final test set would reflect a student's *final* mastery distribution. Again, we run into the challenges of a nonstationary "moving target" problem: the nonstationarity of students means the snapshot of student data observed in the final sessions is not a good test candidate for early instances of the student model (which might reflect earlier distributions of student mastery). The highly personalized nature of the student models also poses evaluation challenges, invalidating other techniques such as 'leave-one-out' cross-validation.

walk-forward validation

Under these circumstances, the most sound approach to model evaluation is a **Walk-forward validation** Stein [2002]. A *walk-forward validation* is a method for incremental model evaluation often used on time-series data, in which a model is trained on all data prior to some checkpoint, and then tested on a fixed-size window of data from after the training checkpoint. The results are recorded, and the checkpoint moves forward in time, adding the previous fixed-size test data to the training set, and testing on a new dataset of the same size from beyond the updated checkpoint. Walk-forward validation is

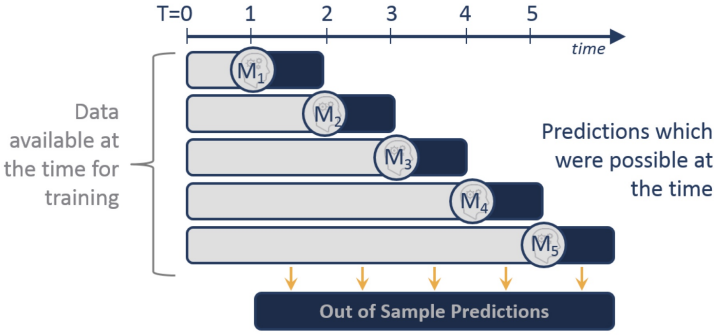


Figure 19: Visualization of Walk-forward Analysis procedure

common in financial analysis, especially for 'backtesting' active models on non-stationary time-series data.

EVALUATING STUDENT LEARNING: ASSESSMENT PHASE ACCURACY AND POSTTEST ASSESSMENT While walk-forward analysis helps overcome the primary practical challenge of evaluating different personalized models, evaluating student learning is relatively more straightforward. As previously discussed, I conducted a posttest assessment of students' letter and word learning. Students filled out a fill-in-the-blank worksheet asking them to write the letter(s) that corresponded to each of the 31 Russian letters used across the games. They also played a matching game similar to WORDDECODER to assess their knowledge of the Russian translation of each word in the CURRICULUM. These metrics provide a form of "*summative assessment*" that attempts to directly assess letter and word knowledge through common classroom formats .

summative assessment

The Assessment Phase provides a measure of student learning based directly on the students' task performance, a type of *stealth assessment*. The Target Words in the Assessment Phase are selected uniformly at random from the CURRICULUM and the robot does not provide any assistance. Students' response accuracy during this phase therefore represents a measure of *task proficiency* i.e., how well a student is mastering the specific task (as opposed to a broader assessment of the underlying literacy skills). The Assessment Phase data is particularly important because it allows us to assess student learning *throughout the course of the study.*

stealth assessment

ENGAGEMENT SURVEY After completing the word and letter posttests, we asked students to complete a short Likert scale survey reporting their engagement during each game and the degree to which they felt both the game design and Jibo's behaviors helped them learn. The survey was self-administered through a Google Form interface, depicted in Figure 20.

6.5 RESULTS

In this section we report on the results of our human subjects study, and provide commentary on the measurement and interpretation of the data, as

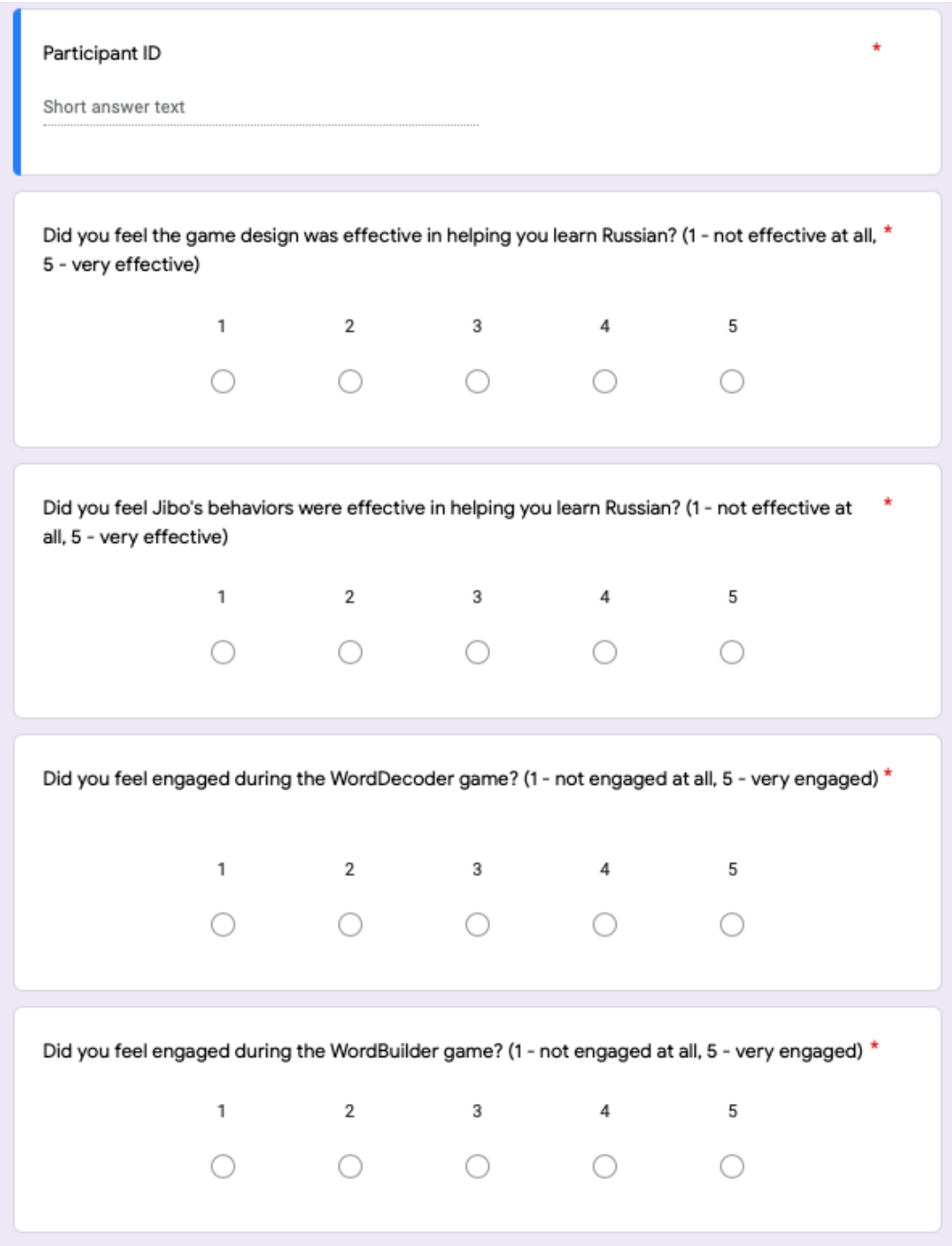


Figure 20: Posttest Survey to gauge student engagement

well as the implications of the results for each of our Computational and Human-centered hypotheses.

6.5.1 *Model Learning Results*

We start by comparing Model Accuracy results across conditions. Figure 21 shows a continuous representation of model accuracy, across each condition.

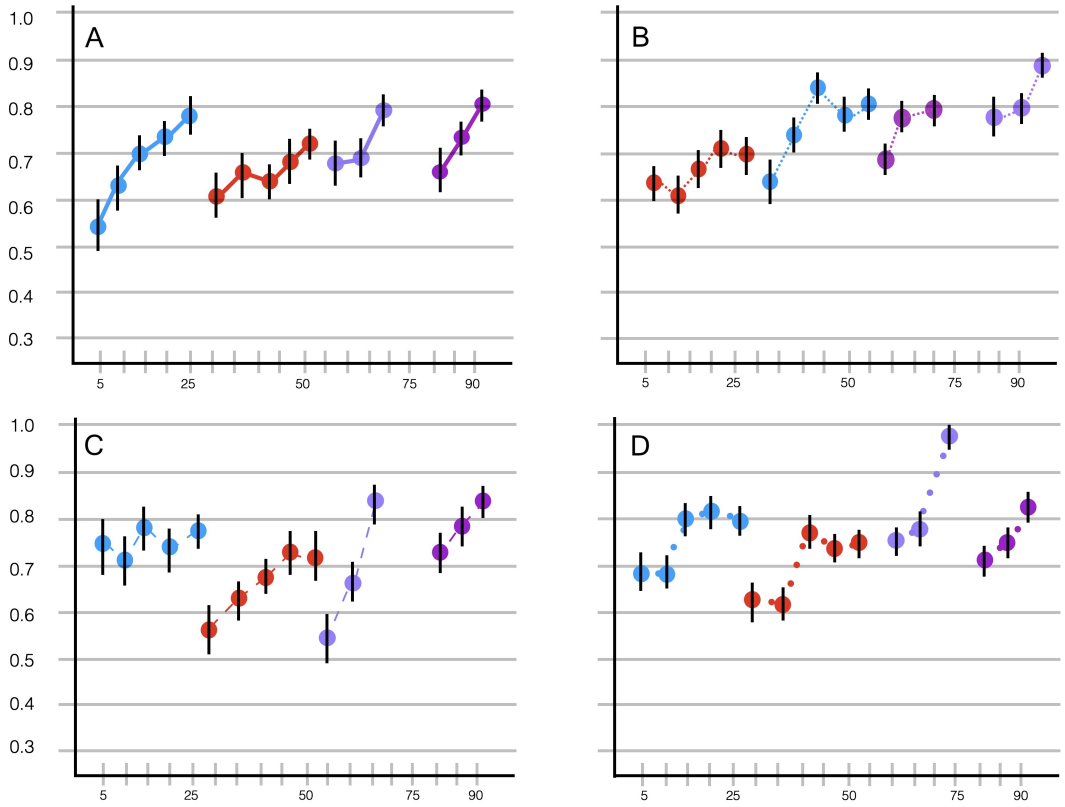


Figure 21: Continuous representation of model accuracy at checkpoints for all conditions. See Table 4 for precise numbers.

The Y-axis indicates the mean model accuracy (+/- standard error of the mean) of the student model at each checkpoint, predicting over the checkpoint test set whether a student's response would be correct. The X-axis indicates total number of 'samples' (i.e. student observations) that each model was trained on (ranging from 5 to 90). Because of the walk-forward testing procedure, the last session of each game has only 3 accuracy checkpoints (because the last 10 data points are the final test set).

The continuous representation across conditions shows a pattern broadly similar to the simulation results: within each session, model accuracy increases, and at each task switch, model performance declines a small amount before increasing again. The primary impact of this result is as an indication that the model is capable of learning correctly, and that our simulation results are at least directionally correct.

6.5.2 *A-B Comparative Model Learning Results*

Figure 22 shows a cross-condition comparison of model accuracy within each task, comparing model performance within each game across Condition A and B participants to isolate the effect of source-task transfer. Participants in Conditions A and B both interacted with a robot that personalized using multitask transfer and continual learning data management. The dif-

Table 4: Mean model accuracy ± standard error of the mean, for all conditions, tasks, and checkpoints

Condition	Task 1, Ckpt 1	Task 1, Ckpt 2	Task 1, Ckpt 3	Task 1, Ckpt 4	Task 1, Ckpt 5	Task 2, Ckpt 1	Task 2, Ckpt 2	Task 2, Ckpt 3	Task 2, Ckpt 4	Task 2, Ckpt 5	Task 3, Ckpt 1	Task 3, Ckpt 2	Task 3, Ckpt 3	Task 4, Ckpt 1	Task 4, Ckpt 2	Task 4, Ckpt 3
A	.54 ± .061	.62 ± .053	.70 ± .039	.72 ± .039	.76 ± .040	.61 ± .046	.64 ± .048	.64 ± .042	.64 ± .053	.74 ± .039	.67 ± .055	.70 ± .044	.82 ± .042	.65 ± .056	.71 ± .049	.77 ± .034
B	.63 ± .047	.60 ± .037	.66 ± .043	.70 ± .039	.69 ± .046	.62 ± .053	.73 ± .051	.84 ± .040	.77 ± .047	.81 ± .033	.69 ± .042	.76 ± .038	.79 ± .035	.77 ± .049	.80 ± .039	.89 ± .024
C	.72 ± .064	.70 ± .059	.78 ± .052	.72 ± .054	.77 ± .051	.55 ± .051	.62 ± .044	.67 ± .042	.72 ± .053	.71 ± .061	.54 ± .053	.67 ± .044	.83 ± .045	.72 ± .042	.78 ± .043	.84 ± .040
D	.69 ± .049	.69 ± .044	.80 ± .038	.82 ± .035	.81 ± .035	.62 ± .041	.61 ± .033	.76 ± .035	.75 ± .038	.74 ± .036	.76 ± .065	.77 ± .034	.95 ± .015	.70 ± .036	.74 ± .044	.81 ± .034

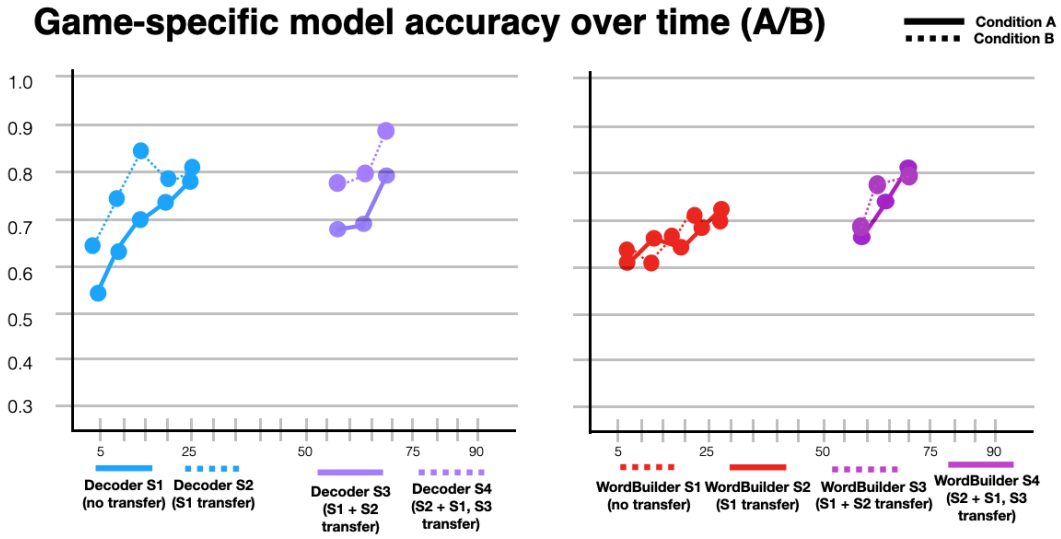


Figure 22: A-B Cross-condition Results. Transfer has a strong positive effect on model accuracy in WordDecoder, but not in WordBuilder

ference between these conditions is the task sequence (starting with either WordDecoder or WordBuilder).

These results reveal two important trends. First, this comparison shows that model accuracy on the WordDecoder task benefits substantially from transferred in WordBuilder data, with improved average model accuracy by 10 percentage points or more in early training. However, the WordBuilder model does not appear to benefit appreciably from transferred in WordDecoder data. In fact, this trend is observed across all of the follow cross-condition comparisons, suggesting that it is driven by some more general relation between the two tasks. The reasons for this phenomenon are discussed further in Section 7.1.1.

Comparing Conditions A and B provides answers to hypotheses C1 and C2 – there is no evidence of negative transfer, affirming C1, and there is evidence of increased learning efficiency in WordDecoder but not WordBuilder. These results are essentially similar to what was expected from simulation study, including the trend of task transfer benefits primarily seen in WordDecoder but not WordBuilder .

Comparing conditions C and D provides yet another view on the impact of multitask transfer on student model predictions. Condition C uses only transfer to personalize across sessions, and does not use **CATDaM** to prune older data. Condition D uses neither transfer nor continual learning to manage training data across sessions, it simply personalizes each session anew. Comparing the two gives us a view into the effect of multitask transfer *without* continual learning.

In this case, we see that there is a significant negative impact of transfer on model performance in WordDecoder. In fact, Condition C S3, is the only session over all conditions in the entire study in which the same-task model accuracy declines by more than 5 percentage points. The classifier accuracy goes from almost 80% to 55%, an enormous drop. The model accuracy recovers

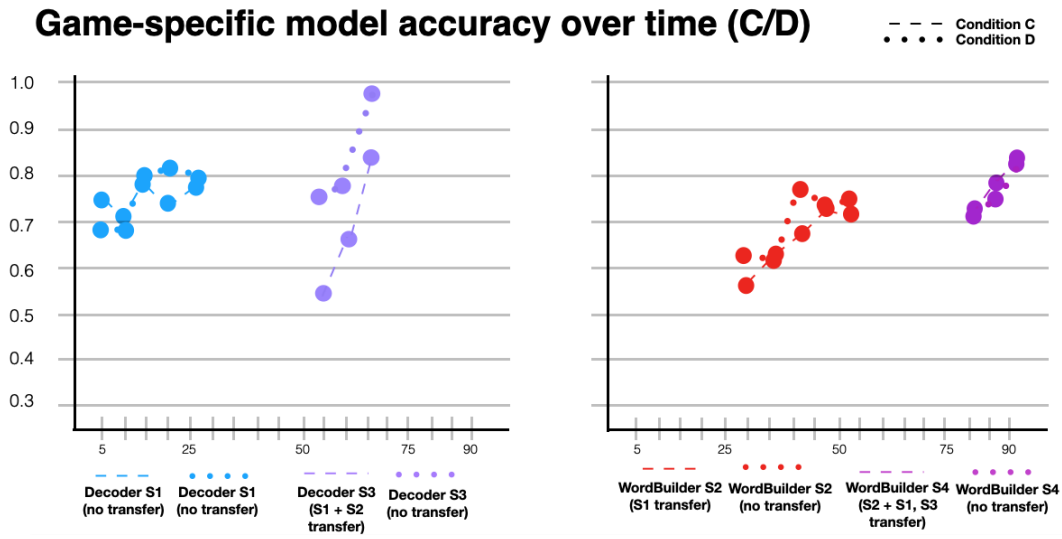


Figure 23: C-D Cross-condition Results. Without continual learning, Task shift causes model performance to degrade in Condition C

after training on new target-task data, but, unlike Conditions A and B, the Condition C model takes a large performance hit from task switch, suggesting that continual learning (**CATDaM** in particular) helps mitigate the potential for negative transfer in a nonstationary domain.

Comparing conditions A and C provides yet more insight on the impact of continual learning directly on model accuracy. In both conditions, data is transferred across tasks, but only in Condition A does **CATDaM** actively prune old training data. Within WordDecoder we see that the Condition C model initially outperforms the Condition A model, but falls behind it after task switch and data transfer, before reaching roughly equivalent performance at the end.

What can we conclude? In early phases of training, continual learning methods like **CATDaM** which remove data from the training set may be counterproductive. This result is likely an artifact of the specific implementation of **CATDaM** as a form of continual learning, in the sense that it removes past data from the training set in response to robot demonstrations independently of the total amount of data already observed (or the potential impact on model performance). Therefore, in early phases of training, continual learning methods like **CATDaM** might *lower* model accuracy (by unnecessarily removing data). However, when a model needs to account for a significant shift in task distribution (e.g. at the start of S_3), these same methods may prove advantageous.

Finally, comparing conditions A and D sheds light on the impact of combined lifelong personalization on model performance. Condition A both incorporates multitask data across tasks and uses **CATDaM** to actively prune old training data. The Condition D agent did neither, and personalizes only within sessions.

Initially, we expected that Condition A would outperform Condition D, due to many studies indicating the benefits of long-term personalization

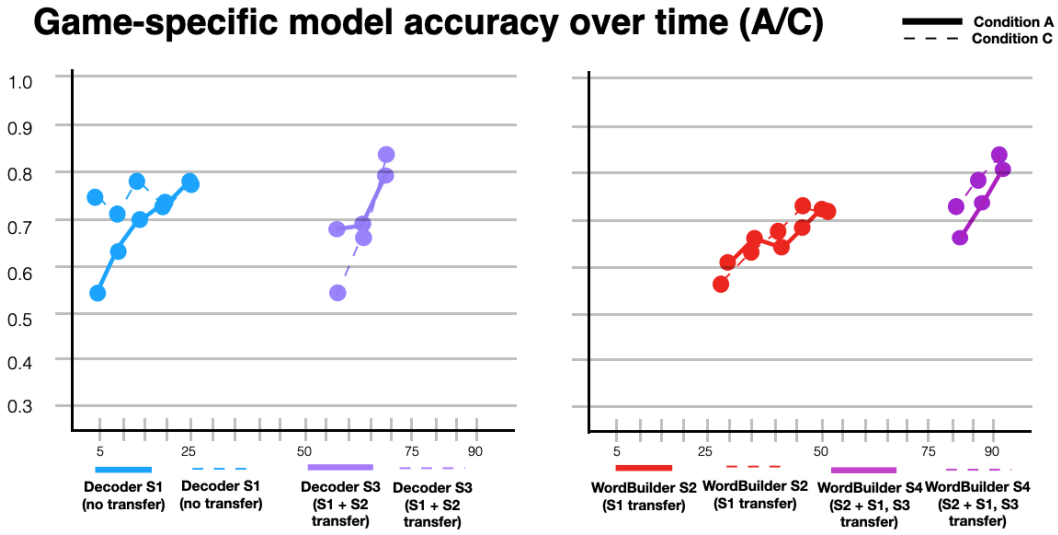


Figure 24: A-C Cross-condition Results. **CATDaM** reduces model accuracy during early phases of training in Condition A

(multiple sessions, without transfer or continual learning). But what we see instead, is that Condition D is substantially more accurate than Condition A over all sessions (in WordBuilder).

In S1, D is doing better than A because it does not implement **CATDaM** and is not actively pruning training data from the early rounds. In S3 however, D does a lot better than A, which itself was markedly better than C. In discussing the A-C cross-condition comparison results, we attributed A's superior performance to the use of continual learning to improve adjusting to the task switch and student nonstationarity.

If the lack of a continual learning learning mechanism in Condition C was clearly detrimental to model accuracy after task switch, why weren't we seeing that in Condition D? From a certain point of view, the within-session-only personalization scheme used by models in Condition D could be viewed as achieving the same goals as a hyper-aggressive variant of **CATDaM**. In other words, **CATDaM** helps a model forget old data, but not as fast as never 'remembering' that data in the first place.

This interpretation makes sense in the context of a (highly) nonstationary domain like estimating student knowledge – the students' learning is proceeding so rapidly that only the most recent data (whether transferred from a source task or directly observed in the target task) is useful for prediction, but it loses predictive power more quickly than expected. We attribute Condition D's especially strong performance to the fact that its personalization algorithm is well-suited for highly nonstationary, rapidly changing targets like student learning.

Under such conditions, more personalized data may not always lead to improved model performance, and simple continual learning methods (like **CATDaM**) may not adjust the model's training data distribution as quickly as the generating distribution (i.e. the student's cognitive state) changes.

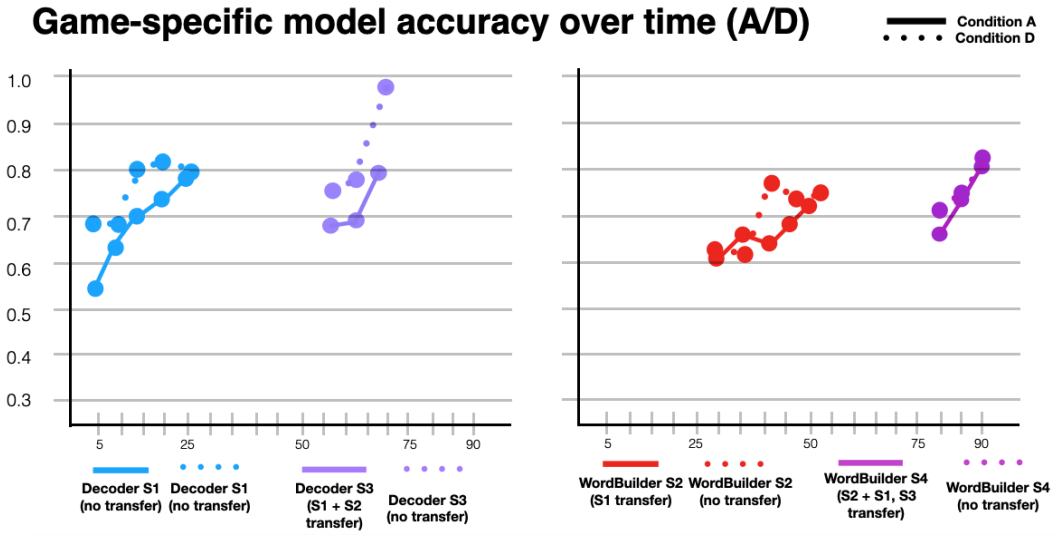


Figure 25: A-D Cross-condition Results. For a rapidly changing modeling target, incorporating long-term personalized data may not improve model performance.

These comparisons provide answers to hypotheses C3 and C4: In general, we do *not* find that Lifelong Personalization improves *final* model performance. We do find a mixed benefit of Continual Learning methods – beneficial after task switch, and slightly detrimental in early phase of training (before task switch).

6.5.3 *Student Learning Results*

Complementing the computational hypotheses, we also explored three "human-centered" hypotheses, focused on the differences in student learning across conditions. These evaluations both support the conclusions of the Computational hypotheses, and reveal additional interaction effects between transfer learning, continual learning, and student learning.

One of the major findings detailed in the previous section was that the model accuracy improvements attributable to cross-task transferred data were not bi-directional, which we attributed to WordBuilder being a more challenging literacy task than WordDecoder, limiting the amount of information that WordDecoder conveys.

Figure 26 shows that Assessment Phase student accuracy was lower for WordBuilder than the corresponding session of WordDecoder in all conditions, affirming that the WordBuilder task was more challenging for students than WordDecoder. Comparing the within-task learning rate (i.e. difference in same-task Assessment Phase student accuracy), we can compute an average measure of learning for each task, compared across personalization conditions.

Participants in Condition A learned at both a lower rate and achieved a lower final proficiency. By this metric, Condition D had the best learning rate, and Condition C was somewhere in the middle. Condition D had notably

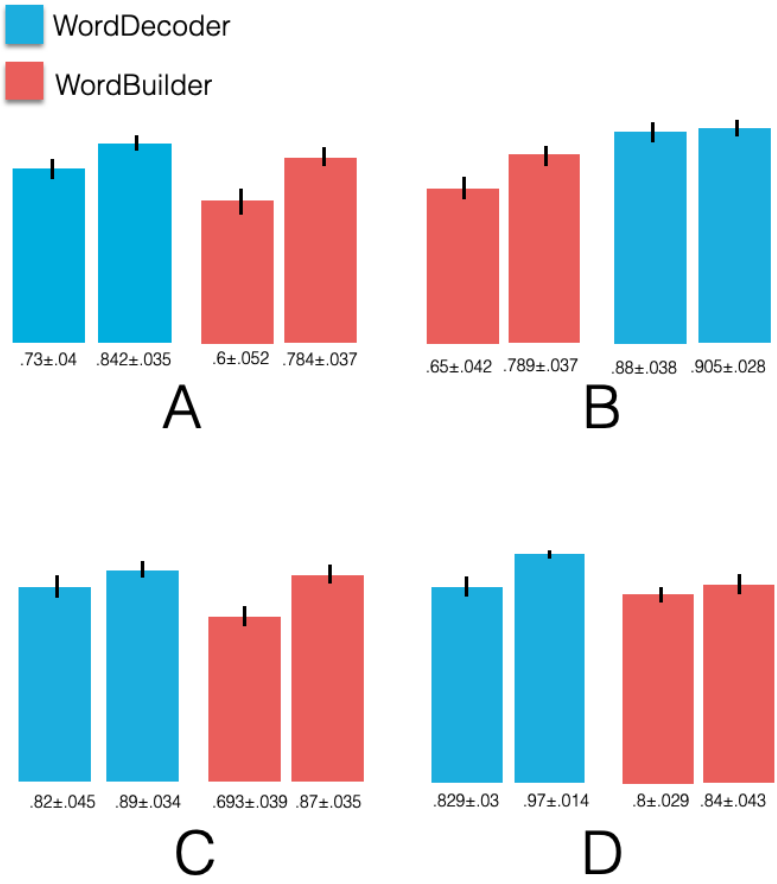


Figure 26: Student Performance during Assessment phase, separated by game. Students performance improved over time in both games, with lower performance overall on WordBuilder. Bars indicate standard error of the mean (SEM)

lower learning gain in WordBuilder from Session 1 to Session 2, but that is largely attributable to a higher base accuracy in Session 1. Condition B provides order-balanced confirmation of these learning dynamics, as the task order was reversed from Conditions A,C, and D, yet the same patterns hold. These results imply that, contrary to the findings of the simulation study, student learning in these games is *hindered* by the addition of multitask transfer and continual learning!

We hypothesize, this discrepancy is largely due to the changes in task design and study population, affecting the underlying dynamics of learning. With young students, principles of spaced repetition are important for learning, and one of the effects of **CATDaM** is to induce the model to bring back words that have been demonstrated or observed before. Older students (especially students of MIT caliber) may be quicker to master words and generalize principles through single examples, therefore their total learning may be driven more by exposure to the greater quantity and variety of words presented in Conditions C and D. These results underscore the importance of a close understanding of learning dynamics within the target

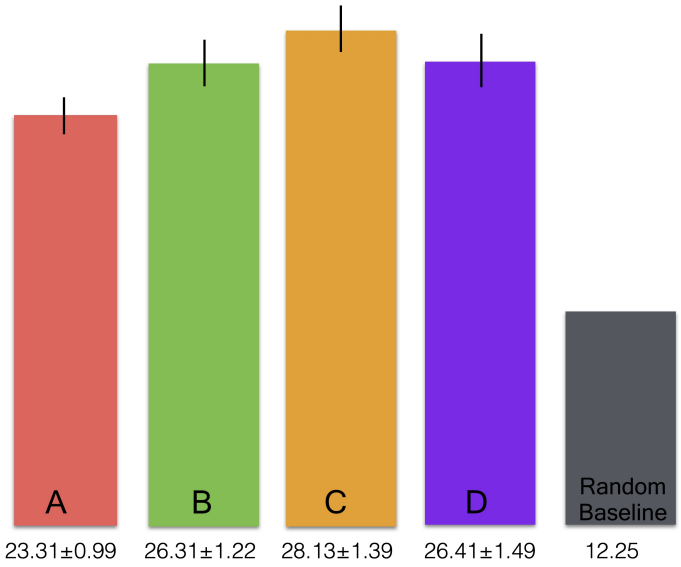


Figure 27: Results from student word learning posttest, by condition. Grey indicates chance level.

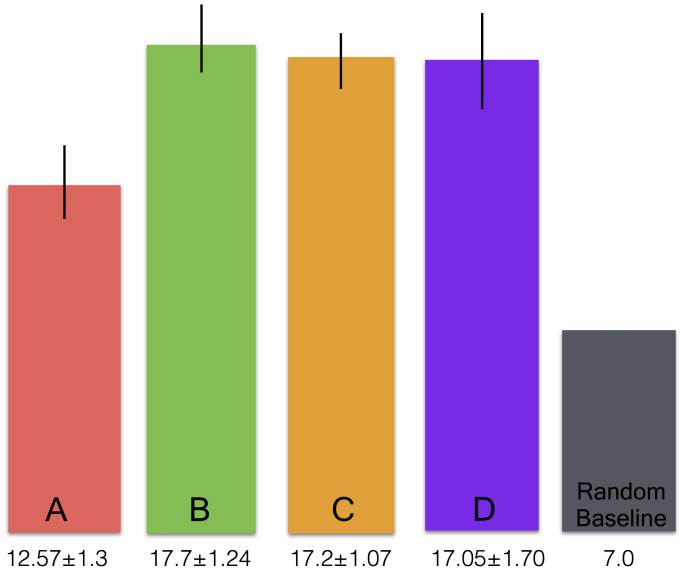


Figure 28: Results from student letter learning posttest, by condition. Grey indicates chance level.

population and task, which should be approximated by the design of each game's personalization model, ContentModel and StrategyModel.

These results are essentially supported by the word and letter posttest learning results, shown in Figures 27 and 28. Condition A students performed significantly worse than students in all other conditions (B,C, and D). The significant difference between Conditions A and B suggests that personalization condition may not be the primary driver of the learning differences (Condition A and B use the same personalization system) and some other factor (perhaps sample variance in language acquisition aptitude, see Section 7.1) may be responsible for lower student learning in Condition A.

Essentially, these results show a robust finding of learning gains over all students for both games, but that there are major learning differences across condition. In terms of the specific hypotheses outlined before study launch, these results suggest that student assessment learning and student posttest learning *is* affected by Personalization condition (H2, H3), but that, in light of the differences between Condition A and B, it may not be the largest or most important factor.

6.5.4 *Student Engagement Results*

Results from the posttest engagement survey (Figure 29) do not indicate any significant differences in the self-reported effectiveness of game design and robot behaviors across conditions. However, participants in Condition A reported lower engagement during WORDDECODER compared to all other Conditions, and lower engagement during WORDBUILDER compared to Conditions B and D. This suggests that some factor other than personalization impacted Condition A participants, perhaps causing both lower engagement and lower student learning than other Conditions. In terms of the original hypotheses, we conclude that student engagement is largely *not* affected by personalization condition (H1).

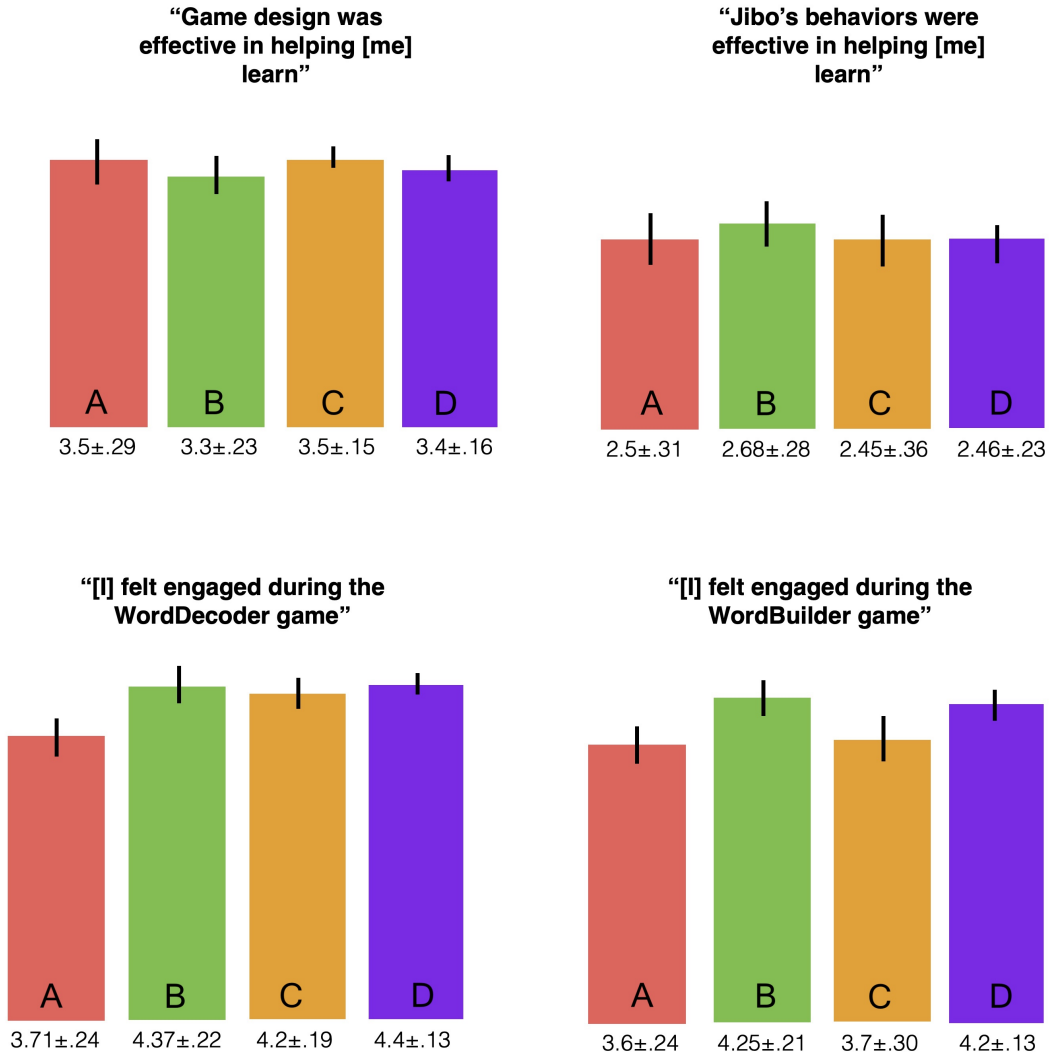


Figure 29: Engagement Survey results

7

CONCLUSION

7.1 SUMMARY OF RESULTS AND ADDITIONAL DISCUSSION

In this thesis, I have introduced and developed the idea of 'lifelong personalization' as a combination of multi-task personalization (via transferrable student models) and continual learning. These concepts were first developed and evaluated in simulation, and then rigorously evaluated in the context of a multi-session human subjects study, generating a wealth of data providing insight into the impact of transfer learning and continual learning on student model accuracy and student learning.

To summarize these results: We do see evidence that transferred task data benefits model accuracy in early phases of task training, especially when combined with continual learning, *in the WordDecoder task*. The task design of WordBuilder may have been too complex or poorly suited for student modeling and does not show any reliable condition differences. We also see that continual learning (implemented in the form of **CATDaM**) negatively affects model accuracy in early phase of task, but that without it, models are more susceptible to negative transfer when task distribution shifts. Another surprising finding is that in a nonstationary environment, more personalized data is not always better. For example, Condition D is 'forgetful' (i.e. does not retain personalized data across tasks or sessions) but does best at predicting future student performance, likely because student learning proceeds very quickly, thereby rapidly changing the students' knowledge state (the cognitive modeling target). Finally, we do observe significant student learning across multiple dimensions over the course of the study. However, it is not clear that the personalization condition has much of an impact on student learning metrics.

Below, I discuss some of the main questions and issues inspired by the conclusions as well as additional context regarding limitations of this research.

7.1.1 *Asymmetry of Task Transfer Benefit*

Why was transfer only impactful from WordBuilder to WordDecoder and not *vice versa*? This is one of the most striking and consistent trends in the data regarding transfer learning in both simulation and human subjects study.

The student learning data provides an illuminating lens, showing that WordBuilder was more difficult for students overall than WordDecoder, measured by assessment-phase student accuracy. From a linguistic perspective, WordBuilder requires two skills: 'letter-sound pairing' *and* 'starting-sound identification'. Moreover, these skills need to be applied in a less familiar linguistic context (i.e., Russian words are presented instead of En-

Restatement of Hypotheses with Conclusions

C1 Lifelong Personalization does not reduce model performance with equivalent data (*Confirmed, comparing Condition A and B*).

C2 Lifelong Personalization helps models achieve better performance with less data (*Weakly confirmed, in early phase*).

C3 Lifelong Personalization improves final model performance (*Not confirmed*).

C4 Continual Learning improves model performance (*Weakly confirmed.* ***CATDaM*** *appears to hurt model performance in early phases, but help model adapt more quickly to task switch*).

H1 Student engagement is affected by Personalization condition (*Not confirmed*).

H2 Student posttest learning is affected by Personalization condition (*Weakly confirmed, Condition A is significantly lower than C and D*).

H3 Student assessment learning is affected by Personalization condition (*Weakly confirmed, Conditions C and D outperform A and B*).

glish). Based on these aspects of task design, it seems likely that a correct answer in WordBuilder implies a student could probably correctly answer the same word in WordDecoder, but not vice versa. Therefore, data from WordDecoder would be less predictive (i.e. provide less information) with respect to the WordBuilder task performance.

7.1.2 *Condition Counterbalancing: Determining Appropriate Pre-test Metrics*

In designing the assessments and measures for this study, we spent a substantial effort designing pre- and post-test assessments for student learning. In our original study design, targeted at young readers of English, we planned to conduct a partial Phonemic Awareness Literacy Screening (PALS) assessment (see Section 3.1.2) before an after the study intervention. When we changed the target population and task design to focus on the Russian language, we changed one of the pre- and post-test assessments to a letter matching worksheets, planning to counterbalance the assignment of students to experimental conditions, based on their pre-test scores.

After the first few participants, we discontinued the letter-matching pre-test because every student so far told us they did not have any prior experience with Russian and would just guess at every letter if pressed. We assigned all participants a "pre-test" score of 0, to use in calculating normalized learning gain. In hindsight, however, this did not mean that all experimental conditions were equally balanced. Despite starting with "no prior knowledge", some conditions might have had a larger population of students who were more likely to acquire the relevant language skills, more quickly. To better balance the groups, we could have instead tried to assess *language acquisition aptitude* – a measure of how quickly of efficiently students learn a new language from

limited practice. This could have been done by asking students to quickly learn or memorize a made-up language mapping, or through other more easily assessed proxy variables, such as amount of second-language study (of any language, not just Russian) or bilingual exposure.

7.1.3 *Close Estimation of Learning Rate*

Perhaps the most surprising finding of all from this study was that Condition D, which "forgot" students' personalized data after each session had the best forward predictions of student performance. Contrary to expectations, it seemed that long-term personalized data was not benefiting the model. Why? If the modeling target is changing (e.g. a student is learning, and therefore their knowledge state is changing) very quickly, and the model does not take that into account or misestimates the rate of change, then stale data holds back the model performance. This is the classic case of domain shift over time, but a unique aspect is that in a tutoring interaction, the modeling agent is deliberately taking action to induce that shift (i.e. teaching the student). The takeaway, then, is not that "more personalized data isn't useful" but rather that it is important to understand how stationary or dynamic the modeling target is, and have models accurately reflect that rate appropriately through active management of training data. The implementation of **CATDaM** reflects one way in which interactive tutoring agents might achieve this, albeit crudely. **CATDaM** uses contextual interaction data to inform the rate at which it manages its training data (e.g. pruning data from the training set after demonstrations or lessons). Even so, clearly this is an imprecise estimate of student learning rate. More rigorous simulation studies could prove helpful in this regard. For instance, we could have evaluated how well a single set of parameters governing **CATDaM**, the ContentModel, and StrategyModel worked across a range of simulated 'learning rates'.

7.1.4 *Student Learning and Model Learning Results Contextualize Each Other*

As we have seen, computationally modeling student learning is a complex and multifaceted problem. Designing a rigorous human-subjects experiment limits the amount of data one can feasibly collect and the number of hypotheses one can reliably evaluate. Nevertheless, looking at both student learning data and model learning data was tremendously helpful in coming to a more complete picture of the dynamics of a learning interaction. Looking at one or two metrics in isolation gives an incomplete picture of the interaction. For instance, looking at student accuracy data in WordBuilder helped elucidate why we did not observe cross-condition differences in WordBuilder model accuracy.

7.2 FUTURE WORK

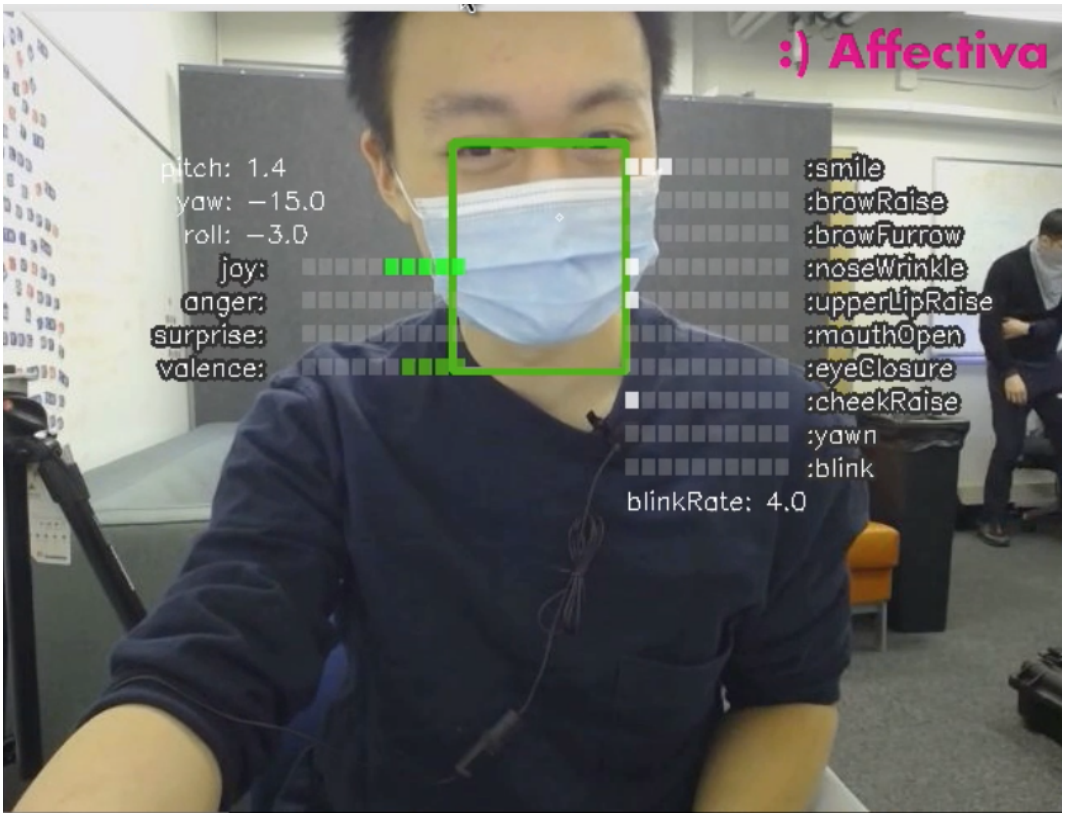


Figure 30: We collected a unique dataset of front-facing gameplay footage from mask-wearing participants.This dataset could help investigate questions regarding the utility of facial affect detection.

FUTURE WORK 1: ANALYZING AND MODELING STUDENT AFFECT In prior research, I have extensively studied how to incorporate student affective response into personalized student modeling Spaulding et al. [2016]; Gordon et al. [2016]; Spaulding and Breazeal [2019]; Spaulding [2020]. It may be somewhat surprising, therefore, that this thesis has not yet introduced any research questions that focus on sensing, interpreting, and adapting to student affective response.

The primary reason why the specific and substantive research questions of this thesis focus on student *cognitive* models rather than *affective* models is that models of student cognition have been studied longer, are more advanced, and have been better validated in field studies with students. By comparison, models of student affect remain quite rudimentary: there is still substantial debate about the reliability of automated affect recognition methods. Barrett et al. [2019], how to interpret sensed affect data D'Mello et al. [2018], and what sorts of conceptual modeling metaphors Yannakakis et al. [2021] are useful when trying to "close the affective loop" (i.e, recognize and respond to student affective displays).

This is not to downplay the deeply challenging and interesting research of modeling student affect — it is a long-held research goal of mine to develop a validated framework for combining affective and cognitive models to enable social robot learning companions to address individual children's specific

educational needs (based on the cognitive model), while also doing so in a way that engages them according to their unique socio-emotional interaction preferences (based on the affective model). And in fact, I believe the study proposed here represents an opportunity to collect a set of high-quality, varied student affect data, synchronized with other contextual interaction and assessment data. This dataset would support a variety of interesting post-hoc analyses particularly if released as an academic resource for future research (subject to ethics and privacy considerations).

In the end, however, a study design has to trade off generality for rigor – too many research hypotheses and variables will reduce the ability of the study to answer the specific research questions regarding multitask and lifelong personalization. Moreover, making student affect response a focal point of our study carries some notable risks (see next section), as we expect that masks will continue to be required in indoor group settings, particularly when young children are present. Therefore, collecting data to support post-hoc affective analysis (if possible), while focusing on direct evaluation and comparison of different personalized *cognitive* models seems to strike the best balance for the goals of this thesis.

FUTURE WORK 2: MORE ADVANCED LANGUAGE TASKS WITH OLDER STUDENTS As discussed, one limitation of this study is that the inconclusive data from WordBuilder suggests a mismatch between the task complexity and students' ability. In addition, our model of personalized learning (designed to promote practice and spaced repetition) may not have been a good match for older students learning patterns (who treated the task more like fast-mapping or few-shot learning).

Through repeated rounds of iterative design and testing with young students, the game actions were repeatedly simplified and slowed down to accommodate the needs of these very young students, many of whom are still developing fine motor skills, in addition to language and comprehension skills. In fact, the original design of WordBuilder allowed students to spell any number of words, using any or all of the provided letter blocks. In future work, I would be interested in modeling language and literacy tasks more appropriate for older students, e.g., tasks designed to help students practice sentence structure, word choice, and other aspects of writing and composition.

FUTURE WORK 3: PERSONALIZED SOCIAL AND EMOTIONAL SKILL TRAINING Socio-emotional skills are another domain in which social robots have a lot of potential to contribute as practice partners. While previous work in this area has focused on using social robots to promote social skills in children with autism, I think a broader population of students could benefit as well. I envision a set of activities themed around sharing, turn-taking, emotional articulation, and perspective-taking could be used to help student develop empathy and socio-emotional maturity. A social robot could

help structure and sequence these activities, reinforce successful interactions, and encourage beneficial patterns of self-reflection.

7.2.1 *Contributions to Long-term Vision*

This thesis spans and synthesizes many different research areas: interactive robotic agents, educational games, long-term student modeling, personalized, multi-task, and continual machine learning models, and on-site educational intervention studies.

I have presented the motivation for the paradigm of 'multitask personalization', grounded in research experience from the past decade, and extended it (in combination with continual learning) to 'lifelong personalization'. These innovations have been thoroughly analyzed in simulation, and results justified launching an in-person human subjects study. The study structure, interaction design, and data collection and evaluation plan of this study were developed to answer both computational and human-centered research questions regarding the impact of multitask and lifelong personalization in a rigorous, controlled study.

In addition to the scientific contributions of this thesis, I also plan to release the game code, model training code, and artistic assets as resources for others to build on. Building *quality*, age-appropriate educational games for young students represents a tremendous amount of work from a team of diverse talents: artists, engineers, game designers, educational researchers, student pilot testers, teachers, and many more.

It is my sincere hope that this thesis will aid and inspire others to continue researching deeply personalized interactive agents that can flexibly personalize to individuals over time and across tasks to promote well-being, personal achievement, and learning.

BIBLIOGRAPHY

Ryan S Baker. Challenges for the future of educational data mining: The baker learning analytics prizes. *JEDM| Journal of Educational Data Mining*, 11(1):1–17, 2019. (Cited on page 36.)

Ryan SJD Baker, Albert T Corbett, and Vincent Aleven. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In *Intelligent Tutoring Systems*, pages 406–415. Springer, 2008. (Cited on page 57.)

Lisa Feldman Barrett, Ralph Adolphs, Stacy Marsella, Aleix M Martinez, and Seth D Pollak. Emotional expressions reconsidered: Challenges to inferring emotion from human facial movements. *Psychological science in the public interest*, 20(1):1–68, 2019. (Cited on page 90.)

Paul Baxter, James Kennedy, Emmanuel Senft, Severin Lemaignan, and Tony Belpaeme. From characterising three years of hri to methodology and reporting recommendations. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 391–398. IEEE, 2016. (Cited on page 34.)

Tony Belpaeme, James Kennedy, Aditi Ramachandran, Brian Scassellati, and Fumihide Tanaka. Social robots for education: A review. *Science Robotics*, 3(21), 2018. (Cited on pages 25, 33, and 36.)

Matthew L Bernacki, Meghan J Greene, and Nikki G Lobczowski. A Systematic Review of Research on Personalized Learning: Personalized by Whom, to What, How, and for What Purpose(s)? *Educational Psychology Review*, 2021. ISSN 1573-336X. doi: 10.1007/s10648-021-09615-8. URL https://doi.org/10.1007/s10648-021-09615-8. (Cited on pages 25 and 37.)

Rodney A Brooks and Maja J Mataric. Real robots, real learning problems. In *Robot learning*, pages 193–213. Springer, 1993. (Cited on page 66.)

Bin Cao, Sinno Jialin Pan, Yu Zhang, Dit-Yan Yeung, and Qiang Yang. Adaptive transfer learning. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010. (Cited on page 35.)

HeeSun Choi, Cindy Crump, Christian Duriez, Asher Elmquist, Gregory Hager, David Han, Frank Hearl, Jessica Hodgins, Abhinandan Jain, Frederick Leve, et al. On the use of simulation in robotics: Opportunities, challenges, and suggestions for moving forward. *Proceedings of the National Academy of Sciences*, 118(1), 2021. (Cited on page 56.)

Nikhil Churamani, Sinan Kalkan, and Hatice Gunes. Continual learning for affective robotics: Why, what and how? In *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 425–431. IEEE, 2020. (Cited on page 27.)

Sidney D'Mello, Arvid Kappas, and Jonathan Gratch. The Affective Computing Approach to Affect Measurement. *Emotion Review*, 10(2):174–183, 2018. ISSN 17540739. doi: 10.1177/1754073917696583. (Cited on page 90.)

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017. (Cited on page 31.)

Michael Geden, Andrew Emerson, Jonathan Rowe, Roger Azevedo, and James Lester. Predictive student modeling in educational games with multi-task learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 2020. (Cited on page 34.)

Goren Gordon, Samuel Spaulding, Jacqueline Kory Westlund, Jin Joo Lee, Luke Plummer, Marayna Martinez, Madhurima Das, and Cynthia Breazeal. Affective personalization of a social robot tutor for children's second language skills. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016. (Cited on pages 33 and 90.)

Shane Griffith, Kaushik Subramanian, Jonathan Scholz, Charles L Isbell, and Andrea L Thomaz. Policy shaping: Integrating human feedback with reinforcement learning. In *Advances in neural information processing systems*, pages 2625–2633, 2013. (Cited on page 66.)

Richard P. Heitz. The speed-accuracy tradeoff: history, physiology, methodology, and behavior. *Frontiers in Neuroscience*, 8:150, 2014. ISSN 1662-453X. doi: 10.3389/fnins.2014.00150. URL https://www.frontiersin.org/article/10.3389/fnins.2014.00150. (Cited on page 48.)

Ben Hixon, Eric Schneider, and Susan L Epstein. Phonemic similarity metrics to compare pronunciation methods. In *Twelfth Annual Conference of the International Speech Communication Association*, 2011. (Cited on page 57.)

Torleiv Høien, Ingvar Lundberg, Keith E Stanovich, and Inger-Kristin Bjaalid. Components of phonological awareness. *Reading and writing*, 7(2):171–188, 1995. (Cited on page 57.)

Francis L. Huang and Timothy R. Konold. A latent variable investigation of the phonological awareness literacy screening-kindergarten assessment: Construct identification and multigroup comparisons between spanish-speaking english-language learners (ells) and non-ell students. *Language Testing*, 31(2):205–221, 2014. doi: 10.1177/0265532213496773. (Cited on page 39.)

Bahar Irfan, Aditi Ramachandran, Samuel Spaulding, Dylan F Glas, Iolanda Leite, and Kheng Lee Koay. Personalization in long-term human-robot interaction. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 685–686. IEEE, 2019. (Cited on pages 31 and 35.)

Natasha Jaques, Sara Taylor, Akane Sano, and Rosalind Picard. Predicting tomorrow?s mood, health, and stress level using personalized multitask learning and domain adaptation. In *IJCAI 2017 Workshop on Artificial Intelligence in Affective Computing*, pages 17–33, 2017. (Cited on page 28.)

W Lewis Johnson and James C Lester. Pedagogical agents: back to the future. *AI Magazine*, 39(2):33–44, 2018. (Cited on page 35.)

Joseph S Lappin and Kenneth Disch. The latency operating characteristic: I. effects of stimulus probability on choice reaction time. *Journal of Experimental Psychology*, 92(3):419, 1972. (Cited on page 47.)

Iolanda Leite, Carlos Martinho, and Ana Paiva. Social robots for long-term interaction: a survey. *International Journal of Social Robotics*, 5(2):291–308, 2013. (Cited on pages 28 and 35.)

Julia C Lenel and Joan H Cantor. Rhyme recognition and phonemic perception in young children. *Journal of Psycholinguistic Research*, 10(1):57–67, 1981. (Cited on page 50.)

Timothée Lesort, Vincenzo Lomonaco, Andrei Stoian, Davide Maltoni, David Filliat, and Natalia Díaz-Rodríguez. Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. *Information fusion*, 58:52–68, 2020. (Cited on page 27.)

Blerina Lika, Kostas Kolomvatsos, and Stathes Hadjiefthymiades. Facing the cold start problem in recommender systems. *Expert Systems with Applications*, 41(4):2065–2073, 2014. (Cited on page 34.)

Robert V Lindsey, Jeffery D Shroyer, Harold Pashler, and Michael C Mozer. Improving students? long-term knowledge retention through personalized review. *Psychological science*, page 0956797613504302, 2014. (Cited on page 28.)

Linda Swank Joanne Meier Marcia Invernizzi, Connie Juel. Pals-k: Phonological awareness literacy screening - kindergarten technical reference. Technical report, Virginia State Department of Education, University of Virginia, January 2015. URL https://palsresource.info/wp-content/uploads/2015/03/ktechnical_ref.pdf. (Cited on page 39.)

Melanie Mitchell. On crashing the barrier of meaning in artificial intelligence. *AI Magazine*, 41(2):86–92, 2020. (Cited on page 35.)

Santiago Ontañón and Jichen Zhu. The personalization paradox: the conflict between accurate user models and personalized adaptive systems. In *26th*

International Conference on Intelligent User Interfaces, pages 64–66, 2021. (Cited on page 36.)

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009. (Cited on pages 32 and 50.)

John F Pane, Elizabeth D Steiner, Matthew D Baird, and Laura S Hamilton. Continued progress: Promising evidence on personalized learning. *Rand Corporation*, 2015. (Cited on page 25.)

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019. (Cited on page 28.)

Hae Won Park, Ishaan Grover, Samuel Spaulding, Louis Gomez, and Cynthia Breazeal. 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*, volume 33, pages 687–694, 2019. (Cited on pages 28, 33, and 36.)

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL http://www.aclweb.org/anthology/D14-1162. (Cited on page 48.)

Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, page 5, 2009. (Cited on page 37.)

Aditi Ramachandran, Sarah Strohkorb Sebo, and Brian Scassellati. Personalized robot tutoring using the assistive tutor pomdp (at-pomdp). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8050–8057, 2019. (Cited on pages 28 and 36.)

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005. ISBN 026218253X. (Cited on page 46.)

Michael T Rosenstein, Zvika Marx, Leslie Pack Kaelbling, and Thomas G Dietterich. To transfer or not to transfer. In *NIPS 2005 workshop on transfer learning*, volume 898, pages 1–4, 2005. (Cited on page 35.)

Paul Ruvolo and Eric Eaton. Active task selection for lifelong machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 27, 2013. (Cited on page 27.)

Thorsten Schodde, Kirsten Bergmann, and Stefan Kopp. 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*, pages 128–136. ACM, 2017. (Cited on page 33.)

Valerie Shute and Matthew Ventura. *Stealth assessment: Measuring and supporting learning in video games*. The mit press, 2013. (Cited on page 43.)

Jasper Snoek, Kevin Swersky, Rich Zemel, and Ryan Adams. Input warping for bayesian optimization of non-stationary functions. In *International Conference on Machine Learning*, pages 1674–1682. PMLR, 2014. (Cited on page 35.)

Harold Soh, Yaqi Xie, Min Chen, and David Hsu. Multi-task trust transfer for human–robot interaction. *The International Journal of Robotics Research*, 39(2-3):233–249, 2020. (Cited on page 35.)

S. Spaulding and C. Breazeal. Pronunciation-based child-robot game interactions to promote literacy skills. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 554–555, 2019. (Cited on pages 59 and 61.)

Samuel Spaulding. Towards transferrable affective models for educational play. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 16(1):340–342, Oct. 2020. (Cited on page 90.)

Samuel Spaulding and Cynthia Breazeal. Frustratingly easy personalization for real-time affect interpretation of facial expression. In *2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 531–537, 2019. doi: 10.1109/ACII.2019.8925515. (Cited on page 90.)

Samuel Spaulding, Goren Gordon, and Cynthia Breazeal. Affect-aware student models for robot tutors. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pages 864–872. International Foundation for Autonomous Agents and Multiagent Systems, 2016. (Cited on pages 33 and 90.)

Samuel Spaulding, Huili Chen, Safinah Ali, Michael Kulinski, and Cynthia Breazeal. 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*, pages 1658–1666. International Foundation for Autonomous Agents and Multiagent Systems, 2018. (Cited on pages 45, 48, 52, and 60.)

Samuel Spaulding, Jocelyn Shen, Hae Won Park, and Cynthia Breazeal. Lifelong personalization via gaussian process modeling for long-term hri. *Frontiers in Robotics and AI*, 8:152, 2021a. (Cited on pages 26, 55, and 61.)

Samuel Spaulding, Jocelyn Shen, HaeWon Park, and Cynthia Breazeal. Towards transferrable personalized student models in educational games. In *Proceedings of the 20th International Conference on Autonomous Agents and*

MultiAgent Systems. International Foundation for Autonomous Agents and Multiagent Systems, 2021b. (Cited on pages 26, 28, 52, 55, and 61.)

Roger M Stein. Benchmarking default prediction models: Pitfalls and remedies in model validation. *Moody's KMV, New York*, 20305, 2002. (Cited on page 74.)

Aaron Steinfeld, Odest Chadwicke Jenkins, and Brian Scassellati. The oz of wizard: simulating the human for interaction research. In *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, pages 101–108, 2009. (Cited on page 55.)

David Thue and Vadim Bulitko. Toward a unified understanding of experience management. In *Fourteenth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2018. (Cited on page 34.)

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. Second language tutoring using social robots: A large-scale study. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 497–505. Ieee, 2019. (Cited on pages 31 and 33.)

Jack M Wang, David J Fleet, and Aaron Hertzmann. Gaussian process dynamical models. In *NIPS*, volume 18, page 3. Citeseer, 2005. (Cited on page 45.)

Annie Xie, James Harrison, and Chelsea Finn. Deep reinforcement learning amidst lifelong non-stationarity. *arXiv preprint arXiv:2006.10701*, 2020. (Cited on page 28.)

Georgios N Yannakakis and Julian Togelius. Modeling players. In *Artificial intelligence and games*, volume 2, pages 203–255. Springer, 2018. (Cited on page 34.)

Georgios N. Yannakakis, Roddy Cowie, and Carlos Busso. The ordinal nature of emotions: An emerging approach. *IEEE Transactions on Affective Computing*, 12(1):16–35, 2021. doi: 10.1109/TAFFC.2018.2879512. (Cited on page 90.)

Michael V Yudelson, Kenneth R Koedinger, and Geoffrey J Gordon. Individualized bayesian knowledge tracing models. In *International Conference on Artificial Intelligence in Education*, pages 171–180. Springer, 2013. (Cited on page 28.)

Zhe Zhao, Lichan Hong, Li Wei, Jilin Chen, Aniruddh Nath, Shawn Andrews, Aditee Kumthekar, Maheswaran Sathiamoorthy, Xinyang Yi, and Ed Chi. Recommending what video to watch next: a multitask ranking system. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 43–51, 2019. (Cited on page 31.)

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

Fin