An Intelligent Tutoring System to Train Psychomotor Skills in Virtual Reality with Programmatic Generation of Training Exercises

ANONYMOUS AUTHOR(S)

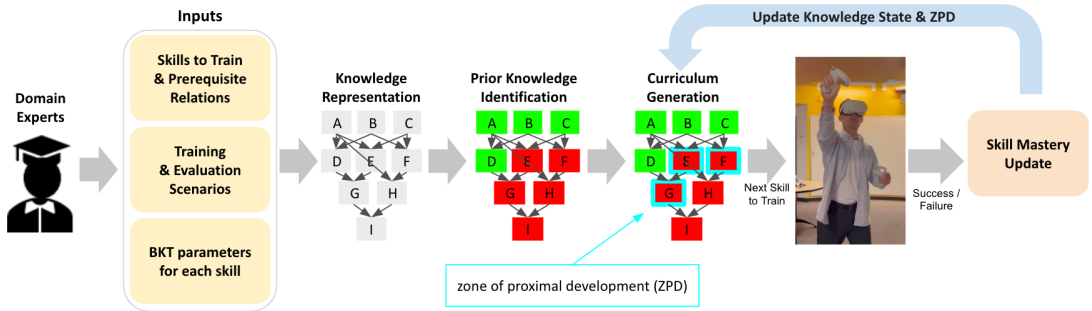

Fig. 1. An overview of our proposed intelligent tutoring system architecture for training psychomotor skills in virtual reality

Intelligent tutoring systems (ITS) have proven successful in academic settings to personalize education to students' varying learning speeds and background knowledge especially when students have limited access to instructors. Similar issues arise when learning psychomotor skills, which consist of spatial perception, cognitive planning, and physical execution of the desired plan via coordination of multiple body joints. Therefore, ITS can potentially be helpful in this setting as well. However, ITS are traditionally designed for purely cognitive skills without any physical motor skills. Most of the literature on ITS for psychomotor skills do not elucidate why certain components of ITS architecture are relevant to these skills requiring motor skills. In this paper, we focus on a particularly salient aspect of psychomotor skills - their high slip rate. We propose to design our ITS in a way that takes slip rate as a configuration parameter explicitly. To this end, we propose a novel use of Bayesian knowledge tracing (BKT) as a skill mastery estimator. We pair BKT with an adaptive curriculum generator. To robustly train learners for these skills with higher slip rate, we model a *distribution* of training scenarios using a probabilistic programming language, Scenic, and iteratively sample various training scenarios from the Scenic program to sequentially generate them in a virtual-reality headset for the learner to interact with. To address the efficacy of our ITS, we conducted between subjects study and compared to a control condition based on self-guided learning. Our results show that our ITS had 32.3% higher learning gains than the self-guided baseline (p-value < .05) with an effect size of 0.41.

CCS Concepts: • **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

Additional Key Words and Phrases: intelligent tutoring systems, virtual reality, psychomotor skills, knowledge tracing

ACM Reference Format:

Anonymous Author(s). 2022. An Intelligent Tutoring System to Train Psychomotor Skills in Virtual Reality with Programmatic Generation of Training Exercises. In *CHI '23: Computer Human Interaction Conference on Human Factors in Computing Systems, 2023, Hamburg, Germany*. ACM, New York, NY, USA, 17 pages. https://doi.org/XXXXXXX.XXXXXXX

Manuscript submitted to ACM

1 INTRODUCTION

Intelligent tutoring systems (ITS) [32] are a manifestation of a long-held aspiration of many researchers and instructors to personalize education. These systems *adapt* the tutorial pace to each student's learning speed, *adjust* the curriculum (i.e. the order in which concepts are taught) according to the student's knowledge state, and provide relevant feedback if the student has any misunderstanding. Therefore, ITS can help students personalize learning to their learning speed and prior (or background) knowledge when student-to-instructor ratio is high or there is no access to an instructor. For these reasons, ITS have been deployed and proven successful in academic courses ranging from K-12 through college [9, 26, 29].

Psychomotor skill training is another domain that needs ITS. These psychomotor skills require spatial awareness, cognitive planning, and precise execution of the desired plan by coordinating one's body joints in unison. Sports, medical surgery, and first responder training are some of the many domains which require specialized psychomotor skills. Similar to academic classrooms, learners widely vary in terms of hand-eye coordination skills, spatial awareness, learning speed, and prior knowledge, and instructors may not be accessible or expensive to afford. Hence, ITS could also potentially help personalize learning in this case. The present may be an opportune time for ITS to aid psychomotor training due to the recent mass commercialization of virtual reality (VR), which substantially lowered the manual labor needed to reconstruct physical, interactive training environments.

ITS are traditionally developed for teaching cognitive skills which do not involve any physical motor skills. Nevertheless, there has been an increasing number of attempts to use existing ITS to train psychomotor skills in medical surgery [30], sports [17], military [8], and driving [28] with some notable success. However, according to a literature survey [23] from 2020 on ITS for psychomotor skills, these literature make use of existing ITS architectures originally designed for academic skills without elucidating their relevance to psychomotor skills.

In this work, we designed an ITS with a clear mapping from its components to characteristics of psychomotor skills. We focus on a particularly salient aspect of psychomotor skills - their high *slip rate*, meaning that there is a high chance of failure to demonstrate mastery for a skill one already learned or mastered. In a book, "Human Error [25]," James Reason proposed a theory of human errors. He categorized cognitive tasks into planning, storage, and execution stages, where each stage is associated with a form of error. The execution stage, when the plan is implemented by the process of carrying out the actions as specified by the plan, is associated with *slip*. Given that one has to coordinate multiple body joints (e.g. waist, arm, wrist, fingers, legs) to execute psychomotor skills, the chance of slip is higher than for cognitive tasks. We propose to design our ITS in a way that explicitly takes slip rate as a configuration parameter. To this end, we use Bayesian knowledge tracing (BKT) as a skill mastery estimator, which has been a de facto standard model for cognitive mastery estimation in traditional ITS. To the best of our knowledge, none of the existing literature has provided a solution to address high slip nature of psychomotor skills and we are the first to employ BKT as a component in ITS for psychomotor skills. Additionally, to robustly train against variations in the environment where skills are applied, we model a *distribution* of training scenarios per skill as a program using a probabilistic programming language called Scenic [7]. Then, we generate training scenarios from the Scenic program and render them in a VR headset until the learner reaches mastery. Finally, we pair BKT and Scenic programs with an adaptive curriculum generator to allocate more training time to skills that a learner has not mastered yet.

Our proposed ITS architecture is visualized in Fig. 1. We first consult domain experts to define a set of skills to train, prerequisite relations among these skills, training/evaluation scenarios, and BKT parameters for each skill. A generic methodology for tuning BKT parameters with experts is explained in the Methodology (Sec. 4). Given these as

inputs, we first represent knowledge as a directed, acyclic, pre-order graph, where each node is a psychomotor skill and directed edges encode prerequisite relations. Then, we identify the skills that a learner already mastered and not mastered, i.e. prior knowledge. With this prior knowledge, we construct a zone of proximal development (ZPD) [15] which is a set of not mastered skills that are in close proximity to mastered skills. We sample a skill to train next from this ZPD set. From this, we iteratively sample a training scenario from the skill's corresponding Scenic program, which encodes a distribution of training scenarios, and render it in a VR headset until the skill's BKT model estimates that the learner has mastered the skill. Then, we update the knowledge state as well as the ZPD set, and then sample the next skill from the ZPD to train. This process repeats until either training time expires or the ZPD set is empty.

To investigate the efficacy of our proposed ITS architecture, we conducted a between-subjects study with 25 participants who were divided into the control and the experimental groups. The experimental group was trained with our ITS. The control group employed self-guided learning, where they were trained with a fixed, non-adaptive curriculum designed by the domain experts but were allowed to jump around skills to create their own curriculum. Furthermore, the control group was instructed to self-estimate one's skill mastery to determine the learning pace, i.e. when to transition to the next skill to train. Note that our ITS is *adaptive* in that the tutoring system is adapting to the learners. On the other hand, the self-guided control group, or the baseline is *non-adaptive* due to the lack of any such system.

Through this evaluation, we aim to answer three research questions:

(1) **R1:** How does the combined adaptivity (i.e. personalized curriculum, skill mastery estimation, distribution of training scenarios) of our intelligent tutoring system affect the learning gains compared to the non-adaptive baseline?
(2) **R2:** How does the accuracy of the BKT in comparison to learners' self-estimation of skill mastery affect the learning gains?
(3) **R3:** Does the adaptivity affect the users' experiences in any way in comparison to the baseline?

In our investigation, we found that learners trained with the adaptive ITS had 32.3% learning gains on average with nearly 50% less standard deviation compared to the baseline ($p < .05$) with the effect size of 0.41. We also found that the BKT models were more highly correlated with skill mastery (correlation coefficient, $r = 0.96$, with p-value < 0.01), compared to the self-estimation ($r = 0.59$, p-value $= .09$). Finally, on average, learners reported positive user experience on par with the baseline with respect to engaging, incrementally challenging, and helpful aspects of the training to learn new skills ($p < 0.05$). These results show that our design choices in ITS help to train psychomotor skills in virtual reality, *without* sacrificing user experience.

2 RELATED WORK

2.1 Training Systems for Psychomotor Skills

A large body of literature have proposed various systems for training psychomotor skills with different emphasis. One focus is on implementing novel simulators to train people in VR, such as industry workers to learn how to weld [11]. Another emphasis is on exploring optimal medium of feedback to efficiently provide corrective guidance to the learners. For example, Physio@Home [31] investigates single versus multiple camera view feedback to provide visual guidance to physiotherapy patients going through rehabilitation exercises at home. Subtletee [35] explored visual, tactile, and auditory feedback to train beginner golfers. Another focus is to design and implement realistic haptic feedback to

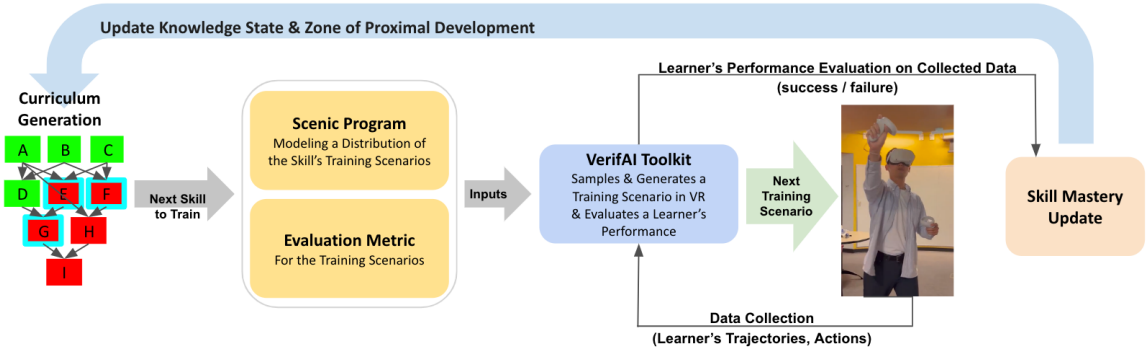

Fig. 2. This figure expands the curriculum generation aspect visualized in Fig. 1. Once we select which next skill to train for, we input a Scenic program modeling a distribution of training scenarios for the skill and its evaluation metric to VerifAI toolkit [5]. Then, VerifAI samples a training scenario from the program, generates it in the learner's VR headset, and assesses the learner's performance using the given metric. This assessment is used to update the skill's BKT model.

improve realism. For example, AirRacket [33] is a directional force feedback system to provide enhanced realism in playing racket-based sports in VR.

Our work has a different focus compared to these prior work. Assuming that a simulator is already given, we focus on designing a intelligent tutoring system which adapts the curriculum (i.e., the order in which skills are taught) and the learning pace (i.e. how many training exercises be given per skill) to the learners varying abilities. A closer related work to our focus is YouMove [1] which uses augmented reality to provide training contents and corrective guidance. But, it asks the learners to construct their curriculum rather than the system adaptively generating one for them. The closest related work is Selfit v2 [22], which is an intelligent tutoring system specifically for health exercise training focused on strength development. However, its design did not account for the high slip aspect of psychomotor skills as we do and our proposed ITS is not specific to a domain.

2.2 Handling the Lack of Data in Psychomotor ITS

Regarding training psychomotor skills, there are barely any mass online systems and dearth of open-sourced big data. Hence, for our study, we referenced the existing ITS architectures which do not require data a priori. Our proposed ITS architecture is inspired by [34] and [21] which were developed to tutor foreign languages, not psychomotor skills, but their architecture is general and does not assume big data. More specifically, we adopted their use of knowledge representation as an acyclic, directed, pre-order graph, formulation of prior knowledge identification approach as graph coloring problem, and curriculum generation using zone of proximal development. However, these previous work did not make use of skill estimation (assuming the first success in solving a problem associated with a particular skill is a sign of mastery, which is not the case in psychomotor skill due to higher slip rate). Hence, we incorporated these ITS components from [34] and [21] with bayesian knowledge tracing and Scenic to propose an ITS for psychomotor skills to conduct our study.

```
from scenic.simulators.vr.actions import *
from scenic.simulators.vr.behaviors import *
model scenic.simulators.vr.model

behavior teammateBehavior(endPoint, disc, trainee, reaction_distance, catch_radius):
    try:
        do Idle()
    interrupt when (distance from ego to disc) < reaction_distance:
        take MoveToAction(endPoint, TEAMMATE_SPEED)
    interrupt when (distance from disc to ego) < catch_radius
        take GrabDiscAction(True, CATCH_RADIUS)

# Define Regions
egoRegion    = MeshVolumeRegion(dimensions = (4, 4, 4), position = (0, 0, 0))
discRegion   = MeshVolumeRegion(dimensions = (2, 4, 4), position = (13, 0, 1))
tmRegion1    = MeshVolumeRegion(dimensions = (2, 2, 3), position = (9.88, -7.34, 1.2))
tmRegion2    = MeshVolumeRegion(dimensions = (2, 2, 3), position = (10.58, 7.34, 1.2))
region_list = [teammateRegion1, teammateRegion2]
random.shuffle(region_list)

# Define Objects using Regions
ego          = HumanPlayer in egoRegion, facing toward goal
disc         = Disc in discInitialRegion
destination = Point in reg_list[0]
teammate     = Teammate in region_list[1], facing toward ego,
                with behavior teammateBehavior(dest, disc, ego, 1.5, 1.5)
```

Fig. 3. An example of a Scenic program (left) modeling a distribution of training scenarios for throwing a frisbee disc to a moving teammate and a snapshot (right) of a generated scenario in a VR headset where the teammate is moving as the learner is throwing a disc towards it

3 BACKGROUND

3.1 Scenic: Probabilistic Programming Language for Scenario Modeling

To robustly train or evaluate a learner's mastery of psychomotor skills against realistic variations in the environment, we need to generate various scenarios in a VR headset for training or evaluation. For example, suppose we train a learner to throw a ball to a moving teammate. The teammate can be moving in various directions with different speeds. How can we efficiently model and generate such variations in a VR headset to train psychomotor skills? To address this issue, we used Scenic [7], a scenario modeling language whose syntax and semantics are designed to intuitively *model* and *generate* dynamic and interactive scenarios involving multiple agents and objects. More technically, Scenic is a probabilistic programming language that allows users to easily specify distributions over environment parameters (e.g. teammate's speed and moving direction). A Scenic program, therefore, models a distribution of concrete scenarios. A *concrete* scenario is defined as a tuple, (I_{v1}, B_{v2}), where I_v is an initial state consisting of different objects, their positions, orientations, etc., with concrete values, v1. B_v defines a behavior assigned to each object in the scenario, where each behavior is parametrized with concrete value, v2. For training or evaluation, we iteratively sample a concrete scenario and generate it in a VR headset.

3.2 VerifAI Toolkit

For both training and evaluation, we need to assess the performance of a learner in VR using an evaluation metric. However, Scenic cannot specify an evaluation metric. Hence, we use VerifAI toolkit [5] which takes a Scenic program and an evaluation metric as inputs. Then, as shown in Fig. 4, it samples a concrete scenario from the program, generates it in a VR headset, collects the learner's telemetry data (e.g., trajectory and actions), and evaluates the learner's performance by computing the given evaluation metric on the collected data.

3.3 Bayesian Knowledge Tracing

Bayesian Knowledge Tracing (BKT) [36] has become the standard in education research for modeling a student's mastery of a skill. BKT assumes a binary knowledge state, meaning that the student is either in the learned (i.e. mastered) or

unlearned state with respect to a skill. It also assumes a binary-graded response from a student's attempt to solve a tutorial/training exercise (i.e., the student either correctly or incorrectly solved the exercise). The underlying statistical architecture of BKT is a hidden Markov model with observable nodes representing students' known binary response sequences obs_t to training exercises and hidden nodes representing students' latent knowledge state at a particular time step t. A canonical BKT model has four parameters: initial probability of knowing the skill a priori (prior), probability of student's knowledge of a skill transitioning from not known to known state after an opportunity to apply it (learn), probability to make a mistake when applying a known skill (slip), and probability of correctly applying a not-known skill (guess). For more detail, please refer to [36]. The mathematical definitions of these parameters and the Bayesian update rule is formulated below.

$$\text{prior} = P(L_0)$$
$$\text{learn} = P(T) = P(L_{t+1} = 1 | L_t = 0)$$
$$\text{guess} = P(G) = P(obs_t = 1 | L_t = 0)$$
$$\text{slip} = P(S) = P(obs_t = 0 | L_t = 1)$$

Note that while $P(L_0)$ denotes the prior parameter, we also define $P(L_t)$ as the probability that the student has mastered the skill at time step t. Bayesian Knowledge Tracing updates $P(L_t)$ given an observed correct or incorrect response to calculate the posterior with:

$$P(L_t | obs_t = 1) = \frac{P(L_t)(1 - P(S))}{P(L_t)(1 - P(S)) + (1 - P(L_t))P(G)}$$
$$P(L_t | obs_t = 0) = \frac{P(L_t)P(S)}{P(L_t)P(S) + (1 - P(L_t))(1 - P(G))}$$

The updated prior for the following time step, which incorporates the probability of learning from immediate feedback and any other instructional support, is defined by:

$$P(L_{t+1}) = P(L_t | obs_t) + (1 - P(L_t | obs_t))P(T)$$

4 METHODOLOGY

We present the methodology for designing our intelligent tutoring system (ITS) for psychomotor skill training in virtual reality. To maximize the number of skills mastered, or learned, within a bounded training time, two adaptive mechanisms are deployed in our ITS.

First, we allocate the training time efficiently, by primarily focusing on the skills that the learners have not yet learned. At the beginning of the training, we characterize the prior knowledge (i.e. skills the learner mastered versus not mastered), which varies across learners. Then, we adaptively determine the order of skills to train from the skill not yet learned.

Second, we use Bayesian knowledge tracing (BKT) as the adaptive mechanism that estimates a learner's mastery of a particular skill. For additional details, please refer to the background (Sec. 3.3). An accurate skill mastery estimation is crucial to determine *when* is appropriate to transition a learner to the next skill for training. An underestimation and an overestimation both result in low number of mastered skills. The former results in an incomplete training, where the learner does not have a chance to train for certain skills at all. The latter results in covering all skills during training but none of the skills are mastered.

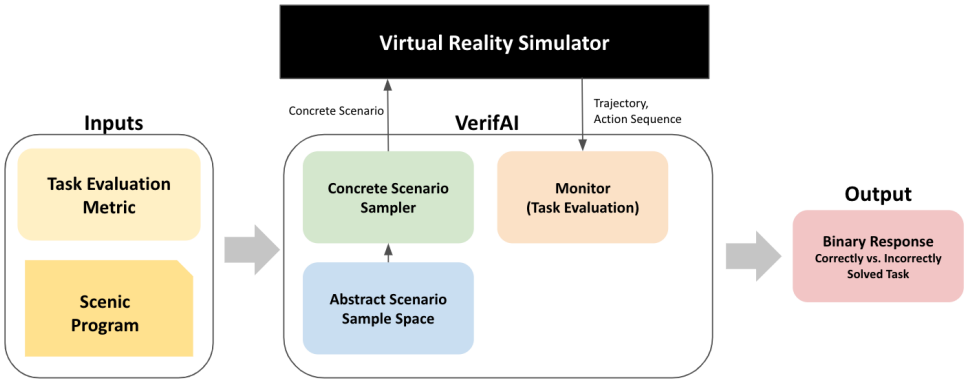

Fig. 4. VerifAI is an open-source tool that we use to generate training/evaluation scenarios in VR and assess a learner's performance in those scenarios. Its architecture is visualized in this figure. VerifAI takes as inputs a Scenic program and an evaluation metric. It samples iteratively a concrete scenario from the program, generates it in VR, and outputs the learner's performance in the training scenario.

4.1 Domain Expert Informed Curriculum

We first recruit experts to gather domain knowledge. We ask domain experts to help build our curriculum by asking them to provide the following information: (i) a set of skills to train, (ii) prerequisite relations among the skills, and (iii) corresponding sets of training and evaluation scenarios for the skills, and (iv) parameters to tune a distinct knowledge tracing model for each skill (refer to Sec. 4.1.2).

4.1.1 Modeling and Generating Training and Evaluation Scenarios in VR. To model and generate training/evaluation scenarios with realistic environment variations and to evaluate a learner's performance, we used open-sourced Scenic (Sec. 3.1) and VerifAI (Sec. 3.2). For each skill, we encoded two Scenic programs, each encoding a distribution of training/evaluation scenarios, respectively. Experts crafted these scenarios with specific tasks designed to train/evaluate particular skills. As shown in Fig. 2, for each skill, we input the corresponding Scenic program and the task evaluation metric to VerifAI which iteratively samples a scenario from the program, generates it in a VR headset to train/evaluate a learner, and measure the learner's performance according to the given metric.

4.1.2 Fine-tuning Knowledge Tracing Models. A BKT model estimates student mastery of a single skill as a probability. Hence, a separate BKT model is used for each skill. We used 0.99 as the probability threshold such that if the knowledge tracing model's estimate of mastery is > 0.99, then we determined that a learner mastered the skill. Typically, in traditional ITS, a threshold of 0.95 or 0.98 is used. We use 0.99 because of the high slip rate which increases BKT's skill mastery more steeply than for typical cognitive skills. Hence, we use 0.99 as a more conservative measure to ensure mastery. A BKT model requires four parameters to be tuned (refer to Sec. 3.3). To tune these parameters for each skill, we asked the domain experts to respond to the following statements regarding the training scenario they provided for each skill in Likert 5-Point scale [18].

(1) There is a high chance a novice trainee will learn the skill after a single training exercise. (learn)
(2) A trainee is likely to solve the task in a training scenario without having mastered the necessary skill. (guess)

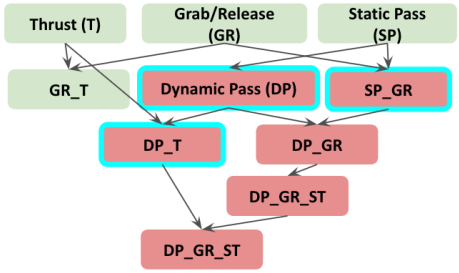

Fig. 5. We represent a knowledge state as a colored, acyclic, directed, pre-order graph as visualized in this figure. Each node represents a skill. The directed edges encode prerequisite relations. The color represents mastery (green: mastered, red: not mastered). The zone of proximal development (ZPD) highlighted in light blue is a set of not mastered skills that are *in proximity* to mastered ones.

(3) Considering the complexity of the maneuvers that a novice trainee has to make to solve for the training scenario, a trainee is likely to make a mistake and fail to solve a task in this scenario even if they had already mastered the necessary skills. (slip)

Additionally, to determine the mapping from the Likert 5-Point scale to probability $\in [0, 1]$, we also asked the expert how many times *in a row* should a learner solve the training exercises to master each skill. We found a mapping such that if the learner answers the *first* three training exercises consecutively correct, then the KT model should output > 0.99. Regarding the "prior" parameter, we conservatively uniformly set it to 0.05 across all skills since we do not have data a priori for estimation. The practice of enlisting experts to help hand set BKT parameters based on expected skill learning trajectories, is not unique to our work. In the first few years of operation, this was the practice established by the Cognitive Tutor [26] for setting their skill parameter values, although data-driven refinements were proposed after substantial student response data had been collected [13, 27].

4.2 Knowledge Graph Generation

Knowledge is represented as a *knowledge graph*, i.e. a directed, pre-order graph as shown in Fig. 5. Given a set of skills to train and their prerequisite relations by the experts, we construct a knowledge graph. Each node represents a pscyhomotor skill to train and is associated with its distinct knowledge tracing model and Scenic programs encoding abstract training and evaluation scenarios. The directed edges encode the prerequisite relation among skills such that the parent nodes pointing to other nodes are prerequisite skills to the children nodes being pointed at.

4.3 Prior Knowledge Identification

A knowledge state, as visualized in Fig. 5, is defined as a colored knowledge graph, where a binary color, red or green, represents mastered or not mastered skill, respectively. Learners have diverse prior knowledge where some may already have learned certain skills. We aim to identify the knowledge state of a learner to allocate more training time to skills not mastered yet.

There is an exploration/exploitation trade-off to consider where more time could be spent increasing our confidence in a student's prior knowledge of a particular skill by giving more assessments of that skill, but this would be at the expense of assessing prior knowledge in additional skills under a tight time budget. Hence, we *approximate* the prior knowledge using the provided prerequisite relations. If the trainee has already mastered a skill, then we estimate that there is a higher chance that the learner has mastered the prerequisite skills. In this case, we update the "prior"

parameter of the prerequisite nodes' KT models to a higher probability in consultation with the experts. On the other hand, if the learner has not mastered a skill, then the *post-requisite* skills, i.e. the skills that have the unmastered skill as a prerequisite, are also likely not mastered. To reflect this, we keep the "prior" parameter of the post-requisite skills to the default value of 0.05. After updating the prior for all prerequisites or post-requisites, if the prior > 0.99, we color the node green, otherwise red.

Using these prerequisite relations, we can efficiently approximate the prior knowledge state if we carefully sample which node to assess. To account for time-efficiency, for each node that is not colored yet, we approximate how much time is saved if the learner already mastered a skill, s, by adding the time constraint for the skill as well as its prerequisites skills'. We denote this time as t_s^+. Similarly, we approximate time saved if the learner did not master the skill by adding the time constraint for the skill and its post-requisites'. We denote this time as t_s^-. Hence, for each uncolored node, s, the worst saved time is $\min(t_s^+, t_s^-)$. We sample for the uncolored node, which maximizes the worst saved time, for a time-efficient prior knowledge identification. We mathematically formulate the algorithm for sampling skill for prior knowledge identification in Equation (1).

$$s^* = \underset{\text{s is uncolored}}{\arg\max} \quad \min(t_s^+, t_s^-) \tag{1}$$

4.4 Personalized Curriculum Generation

Zone of proximal development (ZPD) is a concept from psychology, which we adopt to generate a personalized curriculum. ZPD defines the "boundary zone" of human knowledge, which defines the zone that is not learned yet but has close relation with those already learned. Previous literature shows that, with activities selected from ZPD, students can learn on their own with little guidance from instructors [16, 19], and feel more engaged in learning[4].

As highlighted in light blue in Fig. 5, we define the ZPD to be a *set of red color nodes* that are either one edge away from the green nodes or red nodes with no prerequisite skill. An example of a knowledge state with a ZPD highlighted in light blue is shown in Fig. 5. From the ZPD set, we select for the *next* skill to train, which has the minimum number of prerequisites. If there are more than one such node, we choose the one with a shorter time constraint for its training scenario. We use these heuristics to expedite the training.

As shown in Fig. 2, once a node is selected, we generate a variable number of training exercises until the learner has mastered the skill according to its BKT model. We sample a training exercise from its corresponding Scenic program and generate it in the VR headset. After each training exercise, we compute a Boolean to represent whether the learner solved the task or not, and update the BKT model with the Boolean outcome. Once the BKT model outputs a probability > 0.99, we update the color of the node from red to green, indicating mastery. Then, we update the ZPD set and select the next skill to train and generate a variable training exercises again until mastery. We repeat this process until either the training time expires or the ZPD set is empty.

5 EXPERIMENT

The purpose of the experiment is to address the research questions aforementioned in the Introduction: to investigate the effectiveness of our proposed ITS for psychomotor skills (**R1**), the accuracy of bayesian knowledge tracing (BKT) models for skill mastery estimation (**R2**), and the user experience in training with our ITS in comparison to the self-guided baseline (**R3**).

5.1 Experiment Setting

We selected Echo Arena as an example application domain, which is a zero gravity, frisbee esports which belongs to Meta [20]. It is a complex, strategic game which demands psychomotor skills, which we aim to train in this study. We reconstructed this game in Unity [10] and interfaced Scenic (refer Sec. 3.1) to model and generate desired training and evaluation scenarios in VR.

We recruited four professional Echo Arena esports players, who provided us with inputs (refer Sec. 4.1) to our ITS algorithm. These professionals have achieved the top 10 in ranking over the last few years in the VR Master League [14], which hosts the largest annual Echo Arena tournament. For context, in the most recent tournament in 2022, nearly 8,000 people around the world joined the competition. The experts provided us with (i) a set of ten fundamental skills to train for novices, (ii) prerequisite relations among the skills, (iii) training and evaluation scenarios for the skills, and (iv) a curriculum, i.e. a carefully ordered sequence of skills to train. We asked for the experts' curriculum to train the control group. The knowledge graph encoding (i) and (ii) are visualized in Fig. 5. We provide videos of the training/evaluation scenarios for these ten skills and explanations in the Supplement.

5.2 Participants

We recruited 25 participants through university online forums and mailing lists from a community of users with prior experience in VR games which require dynamic, real-time interactions with the environment such as Beat Saber [3]. However, we ensured no participant had experience with EchoArena. We only accepted participants with such prior experience to minimize participants suffering from motion sickness or nausea from our study. The study was conducted individually where each session with a participant lasted for 120 minutes, and we compensated each with a $40 gift card.

Out of 25 participants, 7 were dropped according to our predetermined criteria, leaving 18 participants for our study. Two participants dropped due to motion sickness (1 from control group, 1 from experimental group). Three participants were dropped immediately after the pre-test because they already mastered more than the majority (> 50%) of the skills prior to training (2 control, 1 experimental) in the pre-test. We dropped these participants because their prior knowledge limits the feasible range of learning gains. Finally, we dropped two participants (1 control, 1 experimental) with low initial task performance who would likely not have progressed through any of our study skills within the allotted time in either condition. To determine this, we required that all participants score at least 50% among the three fundamental skills which serve as prerequisites for the rest of the training skills in either the pre- or post-test. These two participants were dropped after their full 2-hour sessions.

5.3 Experiment Design

We conducted a between subjects experiment. We randomly divided the participants into two disjoint groups, i.e. the control and the experimental groups. The control group could progress through the curriculum at their own pace and skip forwards and backwards one lesson at a time (i.e., self-guided), whereas the progress of experimental group in the curriculum is determined by the adaptive algorithm (as described in the Methodology section). The study consists of four parts: (a) tutorial, (b) pre-test, (c) training, (d) post-test, and (e) exit interview. The experimental and the control groups both followed these same procedures. The only difference was the nature of the training session.

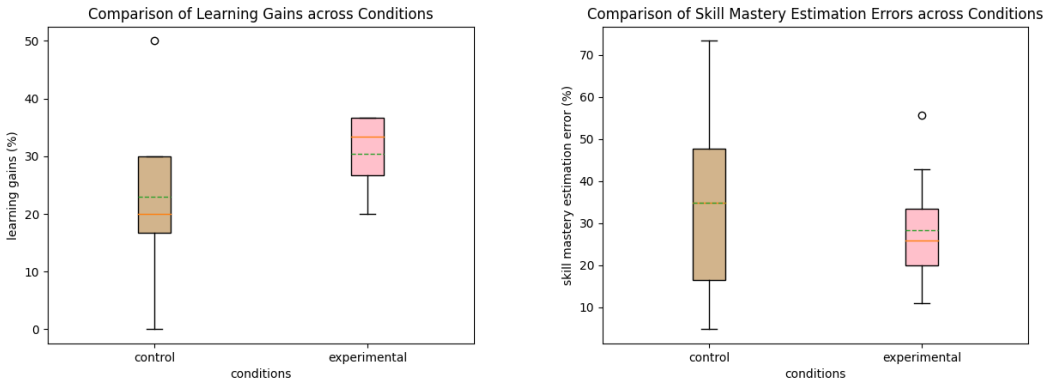

Fig. 6. The left box plot shows that the experimental group had higher learning gains in comparison to the self-guided control group. The right box plot shows that the bayesian knowledge tracing models have lower error in estimating skill mastery when compared to learners' self-assessment in the control group. The green dotted line in the box plot represents the average and the orange line, the median.

5.3.1 Tutorial Session. During the tutorial session, we asked all participants to watch a short video covering basic controls (e.g. thrusts for navigation, grabbing an object, brake) and then asked them to wear our Oculus Quest 2 VR headset and familiarize the controls in a few simple scenarios.

5.3.2 Pre / Post Test Sessions. For each skill in the curriculum enumerated by the experts, we gave participants three evaluation exercises representing the skill. Due to many moving parts (e.g. perception, cognitive planning, physical execution), psychomotor skills tend to have high slip rate, i.e. the chance of failure to demonstrate skill mastery although one has already mastered it. Hence, the experts suggested that we evaluate each skill more than once to accurately measure learning. Three was the maximum number of evaluation exercises we could afford to complete the study within 2 hours. We sampled the three evaluation exercises, i.e. concrete scenarios, for each skill from the corresponding evaluation Scenic program and generated them in VR headset in sequence (refer to Sec. 4.1.1).

5.3.3 Training Session. At the beginning of the training session, for both groups, we asked participants to watch another tutorial video we prepared, which provided tips on how to solve tasks more easily. For both groups, they were instructed to master the 10 skills in 25 minutes of training time. The experimental group trained with the fully adaptive algorithm as described in the Methodology (Sec. 4). The control group trained with the self-guided curriculum, created by the experts in the order prescribed by the experts (Sec. 5.1). The participants in the control group were given three instructions. First, the goal is to "master" 10 skills in 25 minutes. Second, they are being trained with a curriculum that experts designed. Finally, it is up to them to decide when to transition to the next skill. Otherwise, variations of training scenarios for the same task will be provided until they transition. They are allowed to jump back and forth between skills as well. We did not provide the definition of "mastery," and left it to the learners.

5.3.4 Exit Interview. We asked the participants in both groups to rate their user experience in Likert 5-Point scale.

5.4 R1: Comparing Learning Gains Across the Control and Experimental Groups

Our first analysis addresses how adaptivity of the tutoring system affects learning gains.

Method: We first computed the score improvements, i.e. post test - pre test scores, across the control and the experimental groups, and compared their means and standard deviations, where pre-test or post test scores are computed in the following way:

$$\sum_{i=1}^{10}(\#\text{ of successes for skill i in the test})/3$$

Recall that participants are evaluated on 10 different skills where each skill is evaluated 3 times in the pre/post tests. We used Mann-Whitney's U test to determine the statistical significance of these comparisons. We chose Mann-Whitney's U test because although both the control and the experimental groups are sampled from the same population, the sample size is limited to expect normal distributions to hold for unpaired t-test. We used Python Scipy's stats package [12] to compute the significance test. Also, we computed the effect size to measure the magnitude of the adaptivity effect on learning gains.

Results: To assess whether there was any bias in the prior skill level of participants across the two groups, we conducted the Mann-Whitney U test on the control and the experimental groups' pretest scores to see if they are statistically different in distributions. The U test reported 0.26 p-value, which indicates that the two distributions are not different in a statistically significant way.

In terms of learning gains, the experimental group outperformed the control group on average with statistical significance (p-value of 0.04) as shown in Fig. 6 with the effect size of 0.41. On average, the control group improved 22.96% with a standard deviation of 12.90%, whereas the experimental group improved 30.37% with 5.97% standard deviation. In absolute comparison, the experimental group outperformed the control group by 7.41%. More prominently, in relative comparison, this means that the experimental group performed 32.3% = 7.41/26.67 higher with respect to the control group. Furthermore, note that the standard deviation of the experimental group is 46% less than the control group's.

Discussion: The result shows that adaptivity does have a significant effect on improving the learning gains. From an education perspective, the 32.3% increase in learning gains over the self-guided baseline is considerable, as evidenced by the reported effect size of 0.41. Also, the adaptivity reduced the large fluctuations in the learning gains compared to non-adaptive setting, as evidenced by 46% reduction in the standard deviation, providing quality control over the learning outcome and reducing the number of learners falling behind.

We observed three common self-guided strategies in the control group. First, some learners were conservative in skill estimation such that they repeatedly tried more training exercises for each skill although they were able to solve the task consecutively. As a result, they could not cover all 10 skills within the limited training time. Second, in contrast, some transitioned to the next skill on their first success. Considering the high slip aspect of many of the skills we trained for, it would be better to check that they could replicate their success a few times in a row to ensure mastery. Finally, the last strategy type fell somewhere in the middle of the two strategies. The learners who employed the first two strategies had low learning gains, including 0% learning gain in one extreme, which was the lowest across the control and the experimental groups. Meanwhile, some learners who adopted the third strategy had higher learning gain including 50% increase, the highest of the two groups. These extreme variations in both ends of the learning spectrum contributed to high fluctuation in the control group's learning gains. The self-guided learning can be helpful in unlocking learning potential as evidenced by the learner who achieved the highest learning gains. However, this is possible if the learner has an accurate estimation of skill mastery with the ability to efficiently structure curriculum. Our preliminary experiment shows that such learners are rare. Hence, adaptivity is helpful in training for psychomotor skills in VR.

5.5 R2: On the Accuracy of Skill Mastery Estimation across Groups

We analyze the accuracy of skill estimation, where two different estimation methods are compared: (i) self-assessment by the participants (the control group), and (ii) expert-tuned knowledge tracing models (the experimental group).

Method: To analyze the accuracy of skill estimation, we first identified the skills the participants have mastered. In the control group, for each participant, we identified all the skills the participant answered both "Yes (I mastered the skill)" and pressed the "Skip" button, which explicitly expresses one's mastery. Then, we computed the participant's actual and expected scores for the mastered skills. Using the post test result, the actual score was computed in the same way as reported in R1, but only regarding the mastered skills. The expected score was always set to 1 because the participant mastered the skill. Similarly, for the experimental group, we identified all the skills that the knowledge tracing models outputted skill mastery prediction > 0.99. Then, each participant's actual and expected scores were computed in the same manner as for the control group.

We correlated (using Pearson correlation[6]) the actual scores to the corresponding expected scores for the control and the experimental groups, respectively. Also, we computed and compared the means and the standard deviations of the errors, i.e. (expected - actual scores), across the two groups. We used Mann-Whitney's U test to determine the statistical significance of these comparisons.

Results: The knowledge tracing models' estimation of skill mastery was highly correlated to the actual performance with correlation coefficient of 0.96 with p-value < 0.01. In comparison, learners' self assessment of skill mastery was less correlated, reporting correlation coefficient of 0.59 with p-value of 0.09.

Comparing the average errors, knowledge tracing showed higher accuracy in skill mastery estimation on average as shown in Fig. 6. However, Mann-Whitney's U test showed that this result is not statistically significant with p-value 0.46. BKT overestimated participants' skill mastery by $28.21 \pm 13.06\%$, where as participants in the non-adaptive condition overestimated their own skill mastery by $34.81 \pm 23.67\%$, which is $23.40\% = ((33.32 - 28.21)/28.21)$ more error on average than knowledge tracing. Note that the standard deviation for BKT error is 55% decreased compared to the non-adaptive condition's.

In terms of the average number of skills mastered, the experimental group mastered 7.22 ± 2.70 skills out of 10 on average, where as the control group's was 6.22 ± 3.05. The Mann-Whitney's U test showed that this result is not statistically significant with p-value 0.22.

Discussion:

We observe that the expert-tuned knowledge tracing models, which accounts for the learning, slip, and guess rates of the skills, is more accurate than self-assessment. The knowledge tracing models, on average, were $63\% = (0.96 - 0.59)/0.59$ more highly correlated to skill mastery than the self-assessment. This explains the reduction in the average overestimation errors, although not statistically significant.

The average overestimation error of 34.81% in the control group shows the high slip nature of the psychomotor skills we trained for. This means that learners achieved merely 65% success in demonstrating mastery for the skills that they reported to have mastered. From Fig. 6, we also observe a wide variations in mastery estimation error ranging from 4.7% to 73.3% in the control group. This result affirms our assumption that psychomotor skills have high slip rate. In

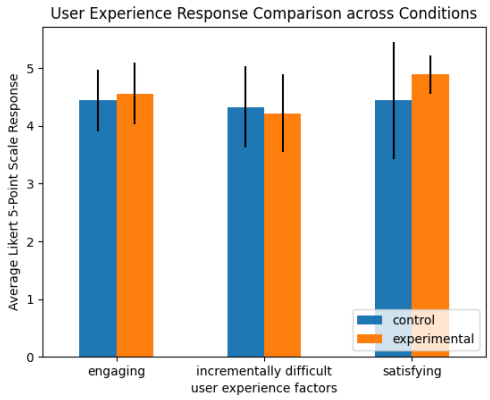

Fig. 7. The bar plot compares the learners' Likert 5-point scale responses on their learning experience. The experimental group's responses are on par with the control's, where both reported positively on their learning experience.

contrast, we note a significant reduction in the standard deviation, which likely contributes to more adequate pacing of tutorial exercises to help learners master skills.

A potential critique may be that this higher accuracy may be due to underestimation of skill mastery, providing more training exercises per skill to lower slip. This would likely increase the performance for the underestimated skills, but at the expense of wasting the limited training time. The result shows that this is not the case, where the average number of skills mastered by learners was higher for the experimental group, although not statistically significant, meaning the experimental group's transition rate through the skills was at least on par with the control's. This shows that KT models are not underestimating skill mastery in comparison to self-assessment.

In short, our result shows that the expert-tuned BKT models more accurately estimated skill mastery and adjusted the learning pace accordingly. This partially explains why the experimental group had higher learning gains.

5.6 R3: User Experience Feedback from Learners

We also analyzed how adaptivity affected the qualitative experience of training in comparison to the self-guided baseline.
Method: To measure whether the training experience was engaging, incrementally difficult, and satisfying, we conducted an exit interview after the post test, where participants were asked to respond to the following three statements in Likert 5-Point scale [18]. A table listing out the 5 point scale and their meanings, i.e. strongly disagree, disagree, neutral, agree, strongly agree) was provided underneath each statement. To test for statistical significance in differences in user experiences, we used the Mann-Whitney U test.

(1) The training session was engaging.
(2) The training session was incrementally challenging.
(3) The training has helped me learn physical skills in virtual reality.

Results: The summary of the results are shown in Fig. 7. The experimental group reported positive qualitative feedback about the training that is on par with the control group on average. Mann-Whitney U test showed that the differences in distributions for engagement, incremental difficulty, and satisfaction are not statistically significant, reporting p-value

0.86, 0.43, and 0.34, respectively. The experimental and control groups both responded that the adaptive training algorithm's curriculum was engaging, incrementally difficult, and helpful to learn psychomotor skills in virtual reality.

Discussion: The results show that our adaptive training algorithm increased the learning gains (as discussed in **R1**), *without* sacrificing learners' training experience. Three adaptive mechanisms employed in our ITS potentially explain the positive learning experiences. First, we efficiently approximate the learner's prior knowledge (Sec. 4.3) to allocate more training time to skills not mastered yet. This helped learners to master more new skills, resulting in a high satisfaction. Second, we adaptively construct curriculum from the zone of proximal development (Sec. 4.4). The activities selected from this zone are known to be engaging and incrementally difficult. Finally, the adaptive tuning of learning pace based on a more accurate skill mastery estimation using BKT models (Sec. 4.4) kept the training engaging and helpful for mastering new skills.

6 LIMITATIONS & FUTURE WORK

In our work, we evaluated the *combined* effects of adaptive curriculum, skill mastery estimation, and feedback on learning gains. To isolate the impact of bayesian knowledge tracing on the learning gains, we would need to design a new experiment to disentangle the combined effects. We also note another limitation that currently we use binary response (i.e. success / failure) to update BKT models. However, in contrast to academic setting, we have access to learners' telemetry data in VR (e.g. trajectory, actions). Using this data, we can compute a continuous, quantitative response to capture how "close" a learner is to solving a task in a given training scenario. Modifying the current BKT model to accept a continuous quantitative response from VR may provide more accurate skill mastery estimation. Finally, in our work, we assumed low effort is needed for other researchers to learn open-sourced Scenic to model distributions of scenarios, since it resembles natural English. Yet, this needs to be validated through a user study.

Additionally, neural approaches to knowledge tracing [24] and reinforcement learning approaches to adapting curriculum [2], for example, could be trialed in VR; however, these are data-hungry approaches and VR technology does not yet enjoy the large scale deployment in educational settings to evaluate these methods. Telemetry data from participants' VR activities could be potentially used to conduct more efficient assessment of skills.

7 CONCLUSION

Our experimental results show that our proposed ITS has higher learning gains compared to the self-guided learning baseline, *without* sacrificing users' learning experience. Learners trained with our ITS improved 32.3% more on average than the self-guided baseline within 25 minutes of training. While the learning gains improved considerably with the ITS, the standard deviation notably decreased by 46% in comparison to the baseline's. In other words, our adaptive system consistently improved learning gains across learners while reducing those falling behind compared to the self-guided learning. This success is partially attributed to more accurate skill mastery estimation using expert-tuned BKT models than learners' self-assessment. Our results show that these BKT models are 63% more highly correlated than learners' self-assessment. This helped significantly reduce the fluctuations in the skill overestimation error by 55% in comparison to the baseline. We hope that our work informs the HCI community on how to design and adapt existing ITS architectures and their components (which are developed for pure cognitive skills) for training psychomotor skills which may involve higher slip rate.

REFERENCES

[1] Fraser Anderson, Tovi Grossman, Justin Matejka, and George Fitzmaurice. 2013. YouMove: enhancing movement training with an augmented reality mirror. In *Computer Human Interaction (CHI) Conference on Human Factors in Computing Systems*.

[2] Jonathan Bassen, Bharathan Balaji, Michael Schaarschmidt, Candace Thille, Jay Painter, Dawn Zimmaro, Alex Games, Ethan Fast, and John C Mitchell. 2020. Reinforcement Learning for the Adaptive Scheduling of Educational Activities. In *Computer Human Interactions Conference on Human Factors in Computing Systems (CHI)*.

[3] BeatSaber. 2022. BeatSaber. Retrieved September 15th, 2022 from https://beatsaber.com/

[4] M. Csikszentmihalyi and I. S. Csikszentmihalyi. 1988. *Optimal experience: Psychological studies of flow in consciousness.* Cambridge University Press.

[5] Tommaso Dreossi, Daniel J. Fremont, Shromona Ghosh, Edward Kim, Hadi Ravanbakhsh, Marcell Vazquez-Chanlatte, and Sanjit A. Seshia. 2019. VerifAI: A Toolkit for the Formal Design and Analysis of Artificial Intelligence-Based Systems.. In *International Conference on Computer Aided Verification (CAV)*.

[6] David Freeman, Robert Pisani, and Roger Purves. 2007. Statistics. *WW Norton & Company (4th Edition)* (2007).

[7] Daniel J. Fremont, Edward Kim, Tommaso Dreossi, Shromona Ghosh, Xiangyu Yue, Alberto L. Sangiovanni-Vincentelli, and Sanjit A. Seshia. 2022. Scenic: A Language for Scenario Specification and Data Generation. *Machine Learning Journal* (2022).

[8] Benjamin Goldberg, Charles Amburn, Charlie Ragusa, and Dar-Wei Chen. 2018. Modeling Expert Behavior in Support of an Adaptive Psychomotor Training Environment: a Marksmanship Use Case. In *International Journal of Artificial Intelligence in Education*, Vol. 28. 194–224.

[9] A.C. Graesser, X. Hu, B.D. Nye, and et al. 2018. ElectronixTutor: an intelligent tutoring system with multiple learning resources for electronics. In *International Journal on STEM Education*, Vol. 5. https://doi.org/10.1186/s40594-018-0110-y

[10] John K Haas. 2014. A history of the unity game engine. (2014).

[11] Ananya Ipsita, Levi Erickson, Yangzi Dong, Joey Huang, Alexa K Bushinski, Sraven Saradhi, Ana M Villanueva, Kylie A Peppler, Thomas S Redick, and Karthik Ramani. 2022. Towards Modeling of Virtual Reality Welding Simulators to Promote Accessible and Scalable Training. In *Computer Human Interaction (CHI) Conference on Human Factors in Computing Systems*.

[12] Eric Jones, Travis Oliphant, Pearu Peterson, and et al. 2001. SciPy: Open source scientific tools for Python. http://www.scipy.org/

[13] Kenneth R. Koedinger, Elizabeth A. McLaughlin, and John C. Stamper. 2012. Automated Student Model Improvement. In *International Conference on Educational Data Mining (EDM)*.

[14] Virtual Reality Master League. 2022. EchoArena VR Master League. Retrieved September 15th, 2022 from https://vrmasterleague.com/EchoArena

[15] C.D. Lee. 2000. Signifying in the Zone of Proximal Development. In *Vygotskian Perspectives on Literacy Research*.

[16] Carol D. Lee. 2005. *An Introduction to Vygotsky.* Routledge, London.

[17] Jeff Lieberman and Cynthia Breazeal. 2007. TIKL: Development of a Wearable Vibrotactile Feedback Suit for Improved Human Motor Learning. In *IEEE Transactions on Robotics*, Vol. 23. 919–926.

[18] R. Likert. 1932. A technique for the measurement of attitudes. *Archives of Psychology* 22, 140 (1932).

[19] Rosemary Luckin. 2001. Designing children's software to ensure productive interactivity through collaboration in the zone of proximal development (ZPD). *Information Technology in Childhood Education Annual* 2001, 1 (August 2001), 57–85.

[20] Meta. 2022. Meta: Social Metaverse Company. Retrieved September 15th, 2022 from https://about.facebook.com/

[21] Tong Mu, Shuhan Wang, Erik Andersen, and Emma Brunskill. 2021. Automatic Adaptive Sequencing in a Webgame. In *International Conference on Intelligent Tutoring Systems (ITS)*.

[22] Laurentiu-Marian Neagu, Eric Rigaud, Vincent Guarnieri, Mihai Dascalu, and Sébastien Travadel. 2022. Selfit v2 – Challenges Encountered in Building a Psychomotor Intelligent Tutoring System. In *International Conference on Intelligent Tutoring Systems (ITS)*.

[23] Laurentiu-Marian Neagu, Eric Rigaud, Sébastien Travadel, Mihai Dascalu, and Razvan-Victor Rughinis. 2020. Intelligent Tutoring Systems for Psychomotor Training – A Systematic Literature Review. In *Intelligent Tutoring Systems*. https://doi.org/10.1007/978-3-030-49663-0_40

[24] Chris Piech, Jonathan Bassen, Jonathan Huanga nd Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. 2015. Deep Knowledge Tracing. In *In Advances in neural information processing systems (NeuriPS)*.

[25] James Reason. 1990. *Human Error.* Cambridge.

[26] Steven Ritter, John R. Anderson, Kenneth R. Koedinger, and Albert Corbett. 2007. Cognitive Tutor: Applied research in mathematics education. In *Psychonomic Bulletin and Review*, Vol. 14. 249–255.

[27] Steven Ritter, Thomas K. Harris, Tristan Nixon, Daniel Dickison, R. Charles Murray, and Brendon Towle. 2009. Reducing the Knowledge Tracing Space. In *International Conference on Educational Data Mining (EDM)*.

[28] S. Ropelato, F. Zünd, S. Magnenat, M. Menozzi, and B. Sumner. 2017. Adaptive Tutoring on a Virtual Reality Driving Simulator. In *1st Workshop on Artificial Intelligence Meets Virtual and Augmented Realities (AIVRAR) in conjuction with SIGGRAPH Asia*.

[29] S. Schiaffino and A. Amandi. 2007. eTeacher: Providing personalized assistance to e-learning students. In *Computers Education*, Vol. 51. 1744–1754.

[30] Anna Skinner, David Diller, Rohit Kumar, Jan Cannon-Bowers, Roger Smith, Alyssa Tanaka, Danielle Julian, and Ray Perez. 2018. Development and application of a multi-modal task analysis to support intelligent tutoring of complex skills. In *International Journal on STEM Education*, Vol. 5.

[31] Richard Tang, Xing-Dong Yang, Scott Bateman, Joaquim Jorge, and Anthony Tang. 2015. Physio@Home: Exploring Visual Guidance and Feedback Techniques for Physiotherapy Exercises. In *Computer Human Interaction (CHI) Conference on Human Factors in Computing Systems*.

[32] Albert T.Corbett, Kenneth R.Koedinger, and John R.Anderson. 1997. Intelligent Tutoring Systems. In *Handbook of Human-Computer Interaction (2nd Edition)*. 849–874.

[33] Ching-Yi Tsai, I-Lun Tsai, Chao-Jung Lai, Derrek Chow, Lauren Wei, Lung-Pan Cheng, and Mike Y. Chen. 2022. AirRacket: Perceptual Design of Ungrounded, Directional Force Feedback to Improve Virtual Racket Sports Experiences. In *Computer Human Interaction (CHI) Conference on Human Factors in Computing Systems*.

[34] Shuhan Wang, Fang He, and Erik Andersen. 2017. A Unified Framework for Knowledge Assessment and Progression Analysis and Design. In *Computer Human Interactions Conference on Human Factors in Computing Systems (CHI)*.

[35] Mikołaj P. Woźniak, Julia Dominiak, Michał Pieprzowski, and et al. 2020. Subtletee: Augmenting Posture Awareness for Beginner Golfers. In *Computer Human Interaction (CHI) Conference on Human Factors in Computing Systems*.

[36] Michael V. Yudelson, Kenneth R. Koedinger, and Geoffrey J. Gordon. 2013. Individualized Bayesian Knowledge Tracing Models. In *International Conference on Artificial Intelligence in Education (AIED)*.