Task Distribution Aware Psychomotor Skill Training with Probabilistic Programs and Bayesian Knowledge Tracing in Virtual Reality

Edward Kim
Alton Sturgis
Zachary Pardos
Kyle Cui
James Hu
Yunzhong Xiao
Boxi Fu
Daniel He
Issac Gonzalez
Alberto L. Sangiovanni-Vincentelli
Sanjit A. Seshia
Björn Hartmann

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2024-16
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2024/EECS-2024-16.html

April 17, 2024

Task Distribution Aware Psychomotor Skill Training with Probabilistic Programs and Bayesian Knowledge Tracing in Virtual Reality

Edward Kim[*], Alton Sturgis[**], Zachary Pardos[***], Kyle Cui[**], James Hu[**], Yunzhong Xiao[**], Boxi Fu[**], Daniel He[**], Issac Gonzalez[*], Alberto Sangiovanni-Vincentelli[*], Sanjit Seshia[*], Bjoern Hartmann[*]
University of California, Berkeley,
Electrical Engineering and Computer Sciences (*), Computer Sciences (**), Education (***) Department

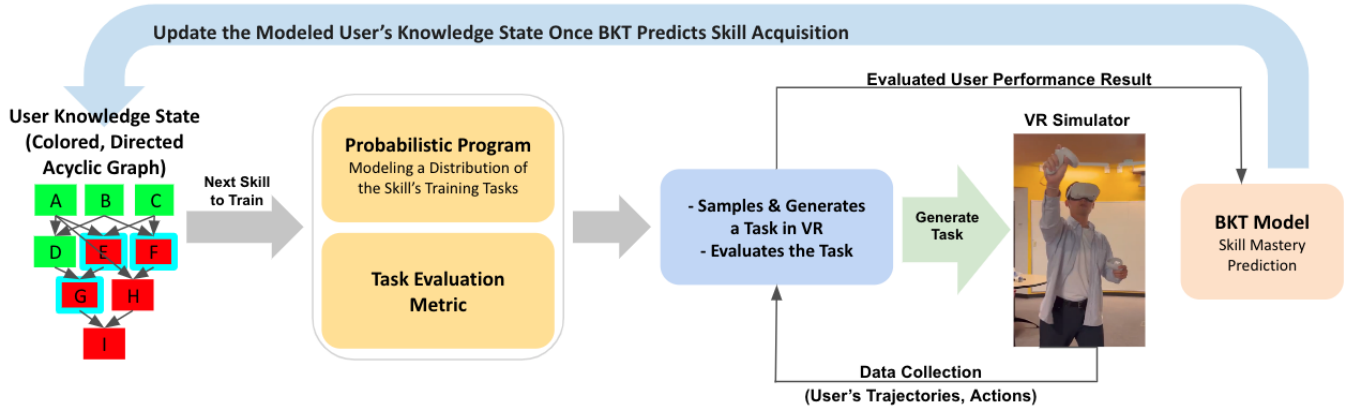

Figure 1: An overview of our algorithm to personalize psychomotor skills in virtual reality with distributions of tasks modelled as probabilistic programs.

ABSTRACT

Virtual reality (VR) is used to train psychomotor skills for domains both within VR, e.g. games, and beyond VR, e.g. sports and healthcare. Although it is a common practice to employ variations of tasks to train psychomotor skills, how to algorithmically predict psychomotor skill acquisition given the task variations, or a distribution, has not been investigated. To address this problem, we derive and adapt ideas from intelligent tutoring systems (ITS), a sub-field of learning sciences. We formally model and generate task distributions with physical constraints that are designed by instructors using a probabilistic programming language. We investigate the effectiveness of Bayesian knowledge tracing (BKT) from ITS to predict psychomotor skill acquisition. Our algorithm sequentially sample a task from a probabilistic program, generates it in VR, and updates the BKT prediction using the performance of a user on the task. We conduct a between subject study that compares BKT to self-prediction of skill acquisition. Our study shows that the experimental condition outperforms the control, and BKT contributes to much more consistent learning outcomes than self-prediction.

1 INTRODUCTION

Virtual reality (VR) has been used as an immersive medium to train psychomotor skills, which consist of physical movements with cognitive conscious planning. These skills can be applied to either within or beyond VR. Within VR, the skills can be utilized to engage in virtual artwork such as painting in three dimension with Tilt Brush [17] and dynamic games like EchoArena [28], which require users to interact with body-driven interfaces. Beyond VR, literature have shown that psychomotor skills trained in VR can transfer to various domains such as healthcare [35] and sports [41], despite the lack of or limited haptics (e.g. weight, tactile). Furthermore, it is especially appealing that, in VR, one can fully control and track objects and agents in the environment to generate a task in a 3D immersive setting, whereas other immersive platforms, such as Kinect [61], may not. For these benefits, VR-based psychomotor skill training has been a topic of interest to HCI community focusing on various aspects of training. These include visual, tactile, and auditory haptic feedback [6, 49, 57] for correction, design of novel physical devices [34, 53]) to enhance sensory realism and engagement for VR training, and construction of high fidelity VR training simulators [24, 45].

In this paper, we focus on an *algorithmic approach to personalize psychomotor skill training in VR*. Previous literature have established that it is important to introduce structured variations to train psychomotor skills. In neuro-physiology, the structured variability in training tasks have been shown to enhance generalization of psychomotor skills [4, 11, 52, 58]. This generalization is crucial because psychomotor skills require individuals to solve variations of tasks. For example, to train a sports player how to accurately pass a ball to a running teammate, coaches may vary the teammate's

EECS Technical Report, April, 2024

speeds (e.g. 10 to 30 km/hr) and directions (e.g. 0 to 180 degrees). The expectation is that if a player experiences enough task variations during training, they will later be able to perform the task for parameters that were not trained but within the distribution (e.g. speed 17 km/hr and direction 85 deg). This variability principle has been incorporated and evaluated in psychomotor skill training in reality (e.g. [22, 54]) and in VR (e.g. [9, 47]). These prior works demonstrate that the principle does enhances psychomotor training in practice.

We study an unaddressed challenge that arise from incorporating structured variations, or distributions, to train psychomotor skills. Suppose an instructor designs a distribution of tasks to train a skill. Tasks are sequentially sampled from the distribution and generated in VR to train and evaluate a user following each task. In this setting, how can we algorithmically predict when a user acquires the skill, meaning *the user can solve all the tasks in the distribution*? While existing literature show that variations in tasks enhances training, they do not propose a mechanism to predict how many variations users should experience for skill acquisition. Rather, they prescribe a fixed amount of time to train a skill with task variations. However, individuals vary in their learning speeds. Ideally, VR training systems should adaptively allocate more time to skills each user find difficult and less on ones that are *predicted* to be acquired. Yet, there is currently no algorithmic means for this prediction.

To develop a personalized algorithm principled in learning sciences, we derive and adapt ideas from intelligent tutoring systems (ITS), a sub-field of learning sciences. First, to formally represent the instructors' domain knowledge of psychomotor tasks to train a skill, for the first time, we demonstrate that a probabilistic programming language (PPL) can be adopted to model and generate a task distribution with physical (spatio-temporal) constraints. Second, we identify an existing, relevant design component in ITS called Bayesian knowledge tracing (BKT) [60] to algorithmically predict psychomotor skill acquisition. Given a probabilistic program, we sequentially sample and generate a task in VR to train a user. After each task, we update BKT's prediction with the user's performance. Once BKT predicts acquisition, it transitions the user to train for the next skill as visualized in Fig. 1. The scope of psychomotor skills we target is defined by the capability of the formalism we use to model and generate physical task distributions. We use a domain-specific PPL called Scenic [15] which can model tasks consisting of objects and agents with distributions over their behaviors and initial conditions. Thus, we target a broad range of psychomotor skills that require fine (e.g. hands) and/or gross (e.g. arms) motor executions to interact with objects and agents in VR. However, Scenic cannot model tasks that are not related to objects. For example, language-based tasks for keyboard typing are outside the scope of Scenic.

We conduct a between subjects study to investigate the accuracy of BKT's prediction of skill acquisition across users and its impact on user experience and learning gains. As many training involve personalization of curriculum, i.e. the order of skills to train, for ecological validity, we investigate BKT in a setting where a set of skills are trained with a set of task distributions. While a number of personalized curriculum approaches have been proposed, the lack of benchmark makes it difficult to identify the state of the art. Thus, we integrate BKT to an existing generic curriculum personalization algorithm [38, 56]. The control condition employs self-guided learning to self-predict skill acquisition and determines when to transition to the next skill, given an expert-designed curriculum. In contrast, our BKT-based algorithm adaptively predicts skill acquisition and generates personalized curriculum for the experimental condition. The study shows that experimental condition results in higher average learning gains than the control with much more consistency (over 50% reduction in standard deviation). Our analysis shows that BKT induces this notably higher consistency in learning. However, we observe the automated system's skill transitions determined solely based on BKT predictions (without any user inputs) induces discomfort from some users. We share suggestions for improvements.

The novel contributions of our work are the following: (1) we demonstrate that a domain-specific PPL can be adopted as a formalism to model and generate psychomotor task distribution with spatio-temporal constraints, (2) BKT contributes to much more consistent learning outcomes than self-prediction when training with a distribution of tasks, and (3) we identify the shortcomings of BKT on user experience as its predictions are used to automate skill transitions and provide suggestions to improve a user interface when adopting BKT.

2 RELATED WORK

2.1 Psychomotor Skill Training

In HCI, numerous dimensions of psychomotor skill training have been investigated in extended reality (XR), an umbrella term for augmented and virtual reality. Some projects focus on the construction of high fidelity VR training simulators [24, 45]. Others investigate diverse forms of visual, tactile, and auditory haptic feedback [6, 49, 57], including life-sized augmented mirrors [2]. Others design new physical devices such as actuated rackets [51, 53] for racket sports to enhance sensory realism and engagement in training. Our work in algorithmic personalization of psychomotor skills in VR is largely orthogonal to these contributions, and could thus be combined with them in future simulators.

Algorithmic approaches have been proposed to incorporate structured variations in adaptive training systems to train psychomotor skills. For instance, Adapt2Learn [54] proposes an adaptive algorithm to physically alter the training environments in reality to generate incrementally difficult tasks. To train a basketball player, it alters the height and the size of the basket hoop to show an improvement in shooting skills based on a user's performance in training. In XR, structured variations are introduced in training in diverse domains such as dental training [9] and space flight training for astronauts [47]. Some VR adaptive training systems introduce structured variations via pre-defined games with varying difficulty levels [18, 55]. To our knowledge, these systems do not employ adaptive means to predict skill acquisition. In our work, we investigate BKT to algorithmically predict skill acquisition across users.

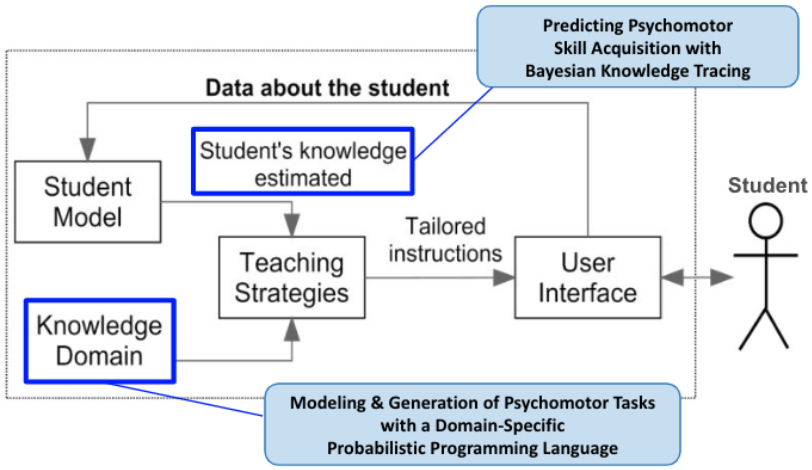

Figure 2: A simplified, generic architecture of intelligent tutoring systems, which is adapted from a survey on ITS architectures [37]. The design components of ITS where our work makes novel contributions are highlighted in blue.

2.2 Intelligent Tutoring Systems for Psychomotor Skills

Intelligent tutoring systems (ITS) [50] 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 [19, 43, 44]. Although ITS are traditionally designed for purely cognitive domains (e.g. mathematics), their applications to psychomotor skill training have started to be investigated with successes [39]. The relevance is that psychomotor skills consists of cognitive skills to consciously plan a sequence of actions to solve physical tasks. Prior work shows that different architectures of ITS have been shown effective for training psychomotor skills in diverse domains such as healthcare [46], defense [16], and sports [31].

Our two novel contributions in relation to prior literature on ITS for psychomotor training are visualized in Fig. 2. The figure represents a simplified, generic ITS architecture, which is adapted from a survey on ITS architectures [37]. In general, ITS (1) models each student's knowledge of skills, (2) models instructors knowledge domain which consists of skills to train and associated sets of cognitive training tasks, and (3) employs a teaching strategy that procedurally generates tasks and tailored instructions, or feedback, to a student via a user interface. The highlighted blue components represent the two aspects where we make novel contributions to ITS. First, there has been a lack of a formalism *model* and algorithms to automatically *generate* instructors' domain knowledge of psychomotor tasks. Without a formalism, each researcher had to devise one's own way to (a) model a distribution with physical (spatial and temporal) constraints and (b) sample from the distribution while satisfying the constraints. Otherwise, the generated tasks can be unrealistic. We demonstrate that a domain-specific probabilistic programming language (PPL) can be adopted to address these challenges. Second, even though task variations are commonly employed in psychomotor training, there has been no adaptive algorithm to predict how many task variations each user should experience in order to acquire a skill, given a task distribution. Consequently, prior work which employs task variations have relied on users to self-predict skill acquisition or prescribed a fixed training time regardless of user's learning speed. We investigate BKT which has only been used in cognitive domains (e.g. mathematics) of ITS. Our study compares BKT to users' self-prediction and shows that BKT can considerably enhance consistency in learning outcomes over self-prediction.

3 BACKGROUND

3.1 Bayesian Knowledge Tracing

Bayesian Knowledge Tracing (BKT) [60] has become the standard in education research for modeling a student's mastery of cognitive skills in domains such as algebra. BKT is used in intelligent tutoring systems (ITS) [50], a sub-field in learning sciences, such as Cognitive Tutor [43] to estimate mastery of cognitive tasks with respect to a distribution of questions sampled from a provided question bank. For example, to educate a student on multiplication, questions are sequentially sampled from a library of questions related to multiplication. After a student completes each question, BKT updates the system's belief of the student's mastery.

Although BKT is designed for purely cognitive tasks, we find it directly relevant for training psychomotor skills as they consist of *cognitive* and motor skills. These cognitive aspects of psychomotor skills involve accurately understanding the physical surroundings and planning which sequence of actions to take. We believe that BKT should be extended to include physical factors to more accurately estimate mastery for psychomotor skills. The scope of this

study is to investigate the effectiveness of BKT in its original form to better understand how it should be extended. The extension of BKT is out of scope for this study.

On this premise, we explain the mechanics of BKT. It assumes a binary knowledge state, meaning that the student either mastered or not mastered a skill. It also assumes a binary-graded response from a student's attempt to solve a task (i.e. correct or incorrect). The underlying statistical architecture of BKT is a hidden Markov model with observable nodes representing the student's history of binary responses obs_t to a sequence of training tasks indexed with t, and hidden nodes representing students' latent knowledge state after experiencing t-th task. BKT parametrizes cognitive learning into four parameters: the student's initial probability of having mastered the skill from prior knowledge before training (prior), probability of the student mastering a previously not mastered skill after experiencing a training task (learn), probability to make a mistake when applying an already mastered skill (slip), and probability of correctly applying a skill that is not mastered yet (guess).

In this paper, we apply these four parameters to model psychomotor skills without any modification. 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 BKT's *prior* parameter, we also define $P(L_t)$ as the probability that the student has mastered the skill after experiencing t-th task. BKT 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, is defined by:

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

3.2 Modeling Task Distribution as Probabilistic Programs

We define a task distribution to operationalize BKT. A task distribution is analogous to a problem template (or variabilized problems) in ITS, which contain random variables. For example, to teach addition, a problem is sampled from a template (e.g. x + y = ?, where x, y are random variables). To use BKT with the template, instructors carefully design the value ranges to maintain the same level of difficulty. Thus, the problems sampled from the same template are evaluated equally. We acknowledge that there are tasks where different values sampled from the task distribution could lead to different performance. If a problem difficulty were to change due to instantiated variable values, then it is considered to be loading on an additional skill which should be explicitly modeled with the creation of another skill [26]. Similarly, we assume that experts are capable of designing physical task variations of the same difficulty, which we encode as a probabilistic program. Thus, we equally evaluate sampled tasks from the same distribution. Investigating the potential challenges for experts to design such task variations and how to update BKT prediction if difficulty varies within a task distribution is left for future work.

In this paper, we formally model a task distribution using a domain-specific probabilistic programming language (PPL) called Scenic [14, 15]. Unlike existing many general PPLs (e.g. Scala [40], BLOG [36]), Scenic provides domain-specific syntax and semantics (i) to intuitively model a distribution with physical (spatial and temporal) constraints and (ii) to sample from the distribution while satisfying the constraints. Scenic is simulator-agnostic and its language is generic such that it has been used in simulation across domains such as self-driving, aviation, robotics, and sports [14]. To briefly illustrate Scenic's capability to model a distribution with spatio-temporal constraints, an example snippet of a Scenic program is shown in Fig. 3. The program models a distribution of training tasks to train a user to accurately pass a frisbee disc to a moving teammate. A snapshot of a task sampled from the program and generated in VR is shown in Fig. 5 (right). Line 9-18 models a distribution with spatial constraints, and line 1-6 models a distribution with temporal constraints. In line 9, the position of `ego`, i.e. the user, is uniformly randomly sampled from a pre-defined `user spawn region`. In line 10, a disc is modeled to be spawned `ahead of` ego, with respect to where ego is facing, by uniformly randomly 1 to 2 meters. Note that this `ahead of` syntax imposes a spatial constraint. In line 12-18, a teammate player's initial position (line 12) and destination (line 13) are uniformly randomly sampled from pre-defined regions as visualized as purple regions in Fig. 5 (right). The green dots within the purple regions represent different sampled initial and destination positions over multiple tasks. A declarative spatial constraint is specified over these two sampled positions in line 18, which states that the distance between the two positions must be greater than 15 meters. In line 1-6, a temporal constraint is assigned over a distribution to define a behavior of the teammate. The `try` and `interrupt` block defines an interactive behavior. The semantics is that, by default, the teammate executes the behavior in the `try` block which is to wait. If the interrupt condition is satisfied at any point in time, then Scenic pauses executing the behavior in the `try` block and executes the action in the `interrupt` block, which is to move to a destination point, until the interrupt condition is no longer satisfied. Note that these `try`, `interrupt` block imposes temporal constraints and a distribution is assigned over teammate's speed. For more details of Scenic language, refer to [15].

4 METHODOLOGY

In this section, we explain our algorithm, as visualized in Fig. 4, to train a set of psychomotor skills with a corresponding set of task distributions. The objective of the algorithm is to maximize the number of skill acquisitions within a bounded training time. The key component of our algorithm is BKT which predicts acquisition of a skill and its predictions are used to transition a user to train for another skill. To account for the complexity of task distributions in skill acquisition prediction, we first solicit domain knowledge

```
behavior TeammateBehavior(user, destination):
    teammate_speed = Range(1, 3) # m/s
    try:
        do Wait()
    interrupt when user.possess_disc:
        take MoveTo(destination, teammate_speed)

# Model Distributions of Initial Positions
ego = User on user_spawn_region
disc = Disc ahead of ego by Range(1,2)

initial_pos = Point on teammate_spawn_region
dest_pos = Point on teammate_destination_region
teammate = TeammatePlayer at initial_pos,
            facing toward ego,
            with behavior TeammateBehavior(ego,dest_pos)

require (distance from initial_pos to dest_pos) > 15 #meters
```

Figure 3: An example snippet of a Scenic program to model a task distribution

from instructors or experts and embed it to the prediction process. This domain knowledge is used to formalize the relations between tasks, distributions, and skills to enable algorithmic personalization. Although our methodology is not specific to any specific VR domain, for ease of explanation, we use the interactions with VR esports experts from our study as a running example.

4.1 Soliciting Domain Knowledge from Instructors

We describe the procedures of our interactions with instructors to solicit their domain knowledge. Specifically, we inquire the following: (i) a set of skills to train and prerequisite relations among the skills, and (ii) corresponding distributions of training and evaluation tasks for each skill with task evaluation metrics, and (iii) BKT parameters for each skill (refer to Sec. 3.1). Each of the three subsections below correspond to (i), (ii), and (iii).

4.1.1 Identifying Skills to Train & Their Pre-requisite Relations. First, we conduct to a joint meeting with experts to identify and represent the training skills and their pre-requisite relations as a knowledge graph shown in Fig. 6. To facilitate the discussion, we used a shared PowerPoint slide. We ask the experts to first brainstorm which psychomotor skills are fundamental to engage in the chosen VR domain, and type the names of the skills on the slide so others can see. Once a sufficient number of skills are written down, we ask the experts to discuss and reach a consensus on which skill to train. Once the set of training skills are determined, on the shared slide, we create a set of blocks, each with a skill name inscribed as shown in Fig. 6 to facilitate the discussion on pre-requisite relations among skills. We ask the experts to first identify blocks (i.e. skills) that do not have any pre-requisite skills. We re-arrange the identified blocks (e.g. T, GR, SP) to form a top level in the shared slide as shown in Fig. 6. Then, we inquire the experts to place the blocks, which immediately require the skills at the top level, right underneath the top level and indicate the pre-requisite relation with directed arrows, where the skill pointed at requires the skill pointed from. We iterate this process until all blocks are consumed, forming a directed, acyclic, pre-order graph, i.e. in short, knowledge graph, for example, as shown in Fig. 6. Note that these pre-requisite relations among skills are not necessary for our methodology to apply (refer to Sec. 4.4).

4.1.2 Designing Task Distributions. Next, per skill, we inquire the experts to verbally and visually explain variations of physical tasks for training and evaluation as well as metrics to assess user performance. We inform the experts that the task variations they design to train each skill should be of the same difficulty (refer Sec. 3.2). In the video call, we create a shared online document[1] for the experts to draw out the physical environments and their variations with their mouses as they verbally explain. For each skill, we ask the experts to collectively discuss and determine the tasks and their evaluation metric by drawing them in the shared document. A snapshot of experts' drawing on Figma visualizing training task distribution for a skill on passing a disc to a dynamically moving teammate is shown in Fig. 5. By the end of this interaction, we have a set of skills, each of which is associated with a task distribution for training and evaluation, respectively.

4.1.3 Tuning BKT parameters. The predictions for skill acquisition should account for the complexity of training task distribution. The more complicated the task distribution is, the more practices with tasks should be offered to reach acquisition. To account for the complexity in prediction, we inquire the experts to tune three of the four parameters of BKT with the knowledge of the task distributions which they designed and the pre-requisite relations among skills. For the BKT's "prior" parameter, we set the parameter to be very low, e.g. 5%. Because the experts may likely not have good mental of distributions of the BKT parameters, we simplify questions to Likert 5-point scale [32]. Then, we map the 5 point scale to probability. Per skill, we ask "given that the user has already acquired pre-requisites for this skill, please answer the following questions in 5 point scale, where each point has the following meaning: 1(Strongly Disagree), 2 (Disagree), 3 (Undecided), 4 (Agree), 5 (Strongly Agree)."

(1) There is a high chance a novice user will learn the skill after a single training task. (learn)
(2) A user is likely to solve the task in a training task without having acquired the skill via random actions. (guess)
(3) Considering the complexity of the maneuvers that a novice user has to make to solve for the training task, a user is likely to make a mistake and fail to solve a task in this task even if they had already acquired the necessary skills. (slip)

The "guess" parameter for psychomotor skills tend to be near 1 point. However, for the example skill on passing a disc to a moving teammate as shown in Fig. 5, the experts' responses vary from 1 to 3 points considering the cases where a user can accidentally pass correctly to the teammate. This practice of enlisting experts to manually tune BKT parameters based on their knowledge of training tasks and their pre-requisite relations among skills, is not unique to our work. In the first few years of operation, this was the practice established by the Cognitive Tutor [43] for setting their BKT parameter values.

[1] For example, a Figma [12] document can be used

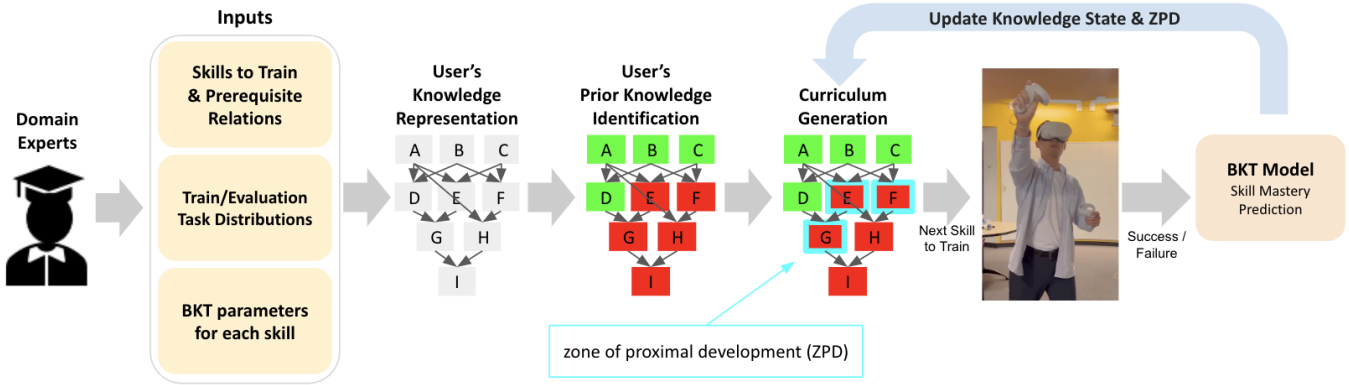

Figure 4: A visualization of our methodology to construct our algorithm.

4.2 Formalization of the Solicited Domain Knowledge

We represent the pre-requisite relations among skills as a knowledge graph, i.e. a directed acyclic graph, as shown in Fig. 4 under "knowledge representation," whose nodes are skills and directed edges are pre-requisite relations. Each skill is associated with a task distribution for training and evaluation, respectively. The task distributions are formally modeled as probabilistic programs using Scenic (refer to Sec. 3.2). For example, the Scenic program in Fig. 3 encodes the experts' description of the training task distribution shown in Fig. 5. A snapshot of a task sampled from this probabilistic program and generated in VR is visualized in Fig. 5. Finally, we implement a BKT model for each skill by mapping the experts' Likert 5 point scale responses to probability. 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.

4.3 Integrating BKT to Personalized Curriculum Generation Algorithm

To train a set of skills to a user, a VR system needs to adequately transition the user from one skill to another. We use BKT's predictions to determine when to transition the user to the next skill. To study the effect of BKT on training users with a set of skills in an ecologically valid setting, we integrate BKT to an existing generic algorithm [38, 56] to personalize curriculum, i.e. the order of skills to train. This is to reflect common practices in psychomotor training where curriculum is personalized. In the following, we briefly describe the algorithm [38, 56] (Sec. 4.3.1, 4.3.2) and how we adapt the algorithm to integrate BKT and use its predictions to control skill transitions (Sec. 4.4).

4.3.1 Prior Knowledge Identification. To personalize the curriculum, the algorithm first identify the user's prior knowledge of skills. It represents the user's prior knowledge as a knowledge state, i.e. a *colored* knowledge graph as visualized in Fig. 6. Each node in the knowledge graph is colored in binary where green indicates the skill in the node is acquired, whereas red represents that it is not acquired. We assume binary knowledge per skill because BKT makes this assumption (refer to Sec. 3.1). To efficiently identify the user's prior knowledge, the algorithm makes use of the pre-requisite relations among skills to expedite the process. The intuition for prior knowledge identification is the following. If a skill is found to be acquired, then the algorithm assumes that all of its pre-requisite skills are also acquired. Therefore, it colors the nodes of the skill and its pre-requisites in green. On the other hand, if a skill is not acquired, then it assumes that all of its post-requisite skills are not acquired. Hence, the algorithm colors the nodes of the skill and its post-requisites to be red. For prior knowledge identification, the algorithm starts with an uncolored knowledge graph and iteratively samples an uncolored node that would maximize the number of colored nodes after evaluation. The following mathematical formulation is used to sample such a skill.

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

s represents a skill to evaluate. n_s^+ represents the number of nodes that will be colored green if the skill, s, is found to be acquired. In contrast, n_s^- is the number of nodes that will be colored red if s is not acquired. The algorithm iteratively samples a node until all the nodes in the graph are colored.

4.3.2 Adaptive Curriculum Generation. The algorithm uses zone of proximal development (ZPD) [29] to personalize curriculum. ZPD is a concept from psychology, which defines the "boundary zone" of human knowledge. This boundary represents knowledge that is not acquired yet but has close relation with those already learned. Previous literature shows that, with tasks selected from ZPD, students can learn on their own with little guidance from instructors [30, 33], and feel more engaged in learning[8]. The algorithm defines the ZPD on the knowledge state as highlighted in light blue in Fig. 4, under "Curriculum Generation." The nodes in ZPD are a *set of red color nodes* that are either one edge away from the green nodes or red nodes with no prerequisite skill. From the ZPD set, the algorithm selects the *next* skill to train, which has the minimum number of prerequisites. If there is a tie, then it uniformly randomly chooses a skill among the tied skills.

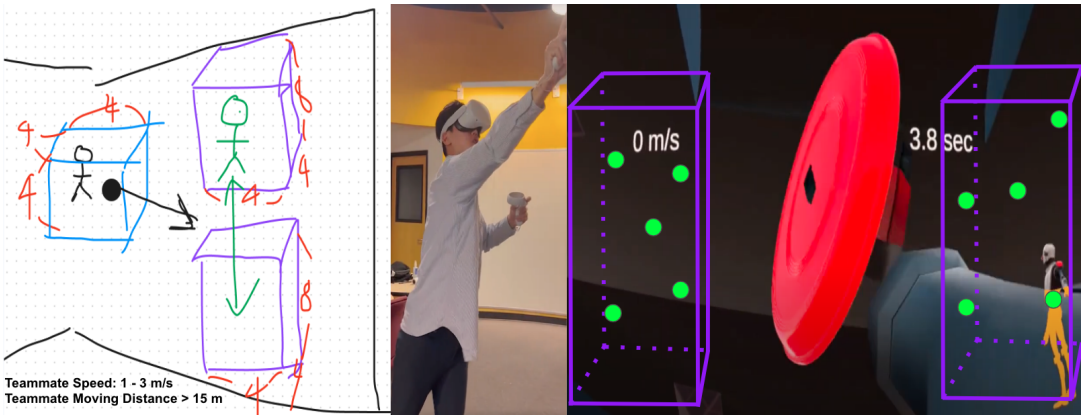

Figure 5: On the left, a snapshot of experts visual description of a training task distribution for a skill on accurately passing a disc to a moving teammate. The out most contour in black represents the top down view of the map in which a user is trained in. The variations in the initial positions of the user (black) and the virtual teammate (green) are drawn with blue and purple boxes with their dimensions in red. The variations in the teammate's moving speed and distance are typed. On the right is a snapshot of a task sampled from the distribution and generated in VR. A user is throwing a disc to a moving teammate at a distance on the right. The purple boxes in the experts' drawing on the left figure are overlaid on the right figure. The green dots in the purple boxes are sampled initial and destination positions for the teammate agent over multiple tasks. This task distribution is modeled as a SCENIC program shown in Fig. 3.2.

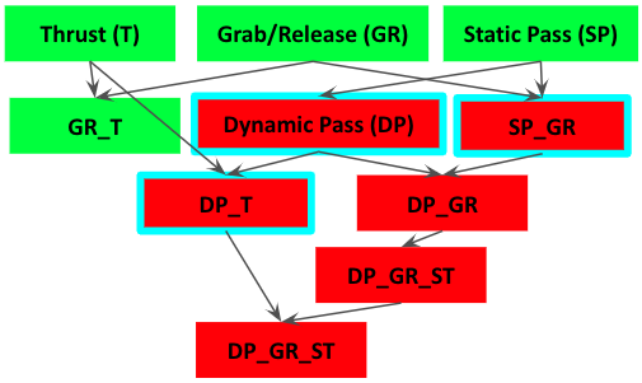

Figure 6: Our system represents a knowledge state as a colored, acyclic, directed, pre-order graph as visualized in this figure. Each node represents a skill. The directed edges encode prerequisite relations. The color represents acquisition (green: acquired, red: not acquired). The zone of proximal development (ZPD) highlighted in light blue is a set of not acquired skills that are *in proximity* to acquired ones.

4.4 BKT-Driven Skill Acquisition Prediction and Transition

We adapt the algorithm to integrate BKT in the following way. Once the algorithm selects the next skill, it retrieves the associated probabilistic program. Then, the adapted algorithm sequentially samples a task from the distribution and generates it in VR until BKT predicts acquisition as visualized in Fig. 1. After each task, the algorithm updates BKT's belief of skill acquisition with the user's performance. Traditionally, the standard use of BKT is that skill acquisition is reached if BKT's prediction (in probability) is greater than 0.99 [60]. We adopt the same criterion in the adapted algorithm. Once BKT predicts acquisition, it updates the knowledge state by changing the color of its node from red to green and then updates the ZPD to select the next skill to train. The adapted algorithm completes once all the skills are acquired, i.e. all nodes are colored green in the graph, or training time expires. Note that pre-requisite relations among skills are not necessary for our methodology to apply. If they exist, then we will leverage these relations to personalize curriculum (Sec. 4.3.2). If they do not exist, we can randomly generate a curriculum and procedurally generate task variations until BKT predicts skill acquisition.

5 EXPERIMENT

We conducted a between subjects study with 18 participants to evaluate our adapted algorithm, which used BKT to predict psychomotor skill acquisition and transition user to next skills based on BKT's predictions. We used self-guided learning as the control condition since there is no baseline algorithm to predict skill acquisition with respect to task distributions. The control condition self-predicted skill acquisition in lieu of BKT.

The three hypotheses of our study are: **(H1)** BKT is more accurate than self-prediction in predicting psychomotor skill acquisition with respect a task distribution, **(H2)** the BKT-driven personalized psychomotor training will induce higher learning gains, and **(H3)** thus, users trained with our algorithm will have a higher drop in subjective task load than the control condition. We also pose a research question: **(R4)** how does the user experience differ between the two conditions?

5.1 Example Application Domain: Esports

Esports is an interesting application domain which require skills that *encapsulate diverse characteristics of psychomotor skills in general*. It requires both fine (e.g. hand, feet) and gross (e.g. arm, legs, waist) movements, while involving careful tactical cognitive planning. Also, it involves physical coordination with other dynamic virtual agent(s). For these reasons, we select Echo Arena, a zero gravity frisbee VR esports, as our example application domain to conduct our study. We reconstruct Echo Arena in Unity [20] and interface Scenic (refer Sec. 3.2) to model and generate the desired training and evaluation scenarios in VR.

5.2 Experts/Instructors Recruitment

We recruited four professional Echo Arena esports players via direct messaging on Discord [23]. They provided us with necessary domain knowledge (refer Sec. 4) through 2 hours of joint video call. Each professional was paid $50 for their time. These professionals had achieved the top 10 in ranking over the last few years in the VR acquire League [28], which hosted 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 [27]. These four experts also had experience in coaching novices or amateur Echo Arena players.

5.3 Participants

We recruited participants through university online forums and mailing lists from a community of VR users. We received 25 responses of subjects with prerequisite dynamic VR game experience. Out of the 25 respondents, we excluded 7 subjects according to our three *pre-determined* exclusion criteria: 1) exhibiting motion sickness, 2) too much skill expertise (no opportunity for learning), and 3) extreme lack of hand-eye coordination (unlikely to acquire any skill during our short training session). The accepted 18 participants' ages ranged from 19 - 25 years, with 4 females and 14 males. Eligibility criteria and a summary of participants' backgrounds are listed in the supplemental material. Each participant was financially compensated with $40 gift card for their 2 hours of participation. For the participants who were excluded according to our pre-determined criteria, they were compensated for the time they participate at $20 per hour rate.

5.4 Procedure

We conducted an IRB approved between subjects experiment to avoid learning and fatigue effects. We randomly divided the accepted 18 participants into two disjoint groups, i.e. the control and the experimental groups, with 9 participants in each condition. The study is conducted individually, not in groups. The study consists of the following sessions[2]: basic tutorial (5 min), pre-test (15 min), advanced tutorial (10 min), training (25 min), post-test (15 min), and exit questionnaire (5 min), with 10 min breaks in between sessions including the half way through the training session. The details of each session is explained in the supplement. We trained and evaluated 10 different skills provided by our recruited experts. The pre/post tests examined the 10 skills with a variable number of tasks. The task distributions to train and evaluate the skills were the same for this study. For pre/post tests, we sampled a different constant number of tasks per probabilistic program; per skill, the number of minimum consecutive successes required for the skill's BKT to predict skill acquisition was used for pre/post test. This is to allow our algorithm to assess skill acquisition for prior knowledge identification.

The 10 skills and their pre-requisite relations are visualized as a knowledge graph in Fig. 6. Thrust (T) relates to a navigation skill using thrusters in EchoArena's zero gravity space. Grab/Release (GR) is another navigation skill by only grabbing and releasing static objects in space. Static pass (SP) is a skill to accurately pass a disc to a static teammate. Dynamic pass (DP) is to accurately pass to a moving teammate. These skills require fine (e.g. hand) and gross (e.g. arm) movements. For example, the snapshot of a human player in Fig. 5 is training for a dynamic pass, where the user retracts and then extends the arm to throw the disc to a moving teammate. The user grabs and releases the disc by pressing on a button on the hand controller. Grabbing and thrusting can also be executed by pressing different buttons on the controller. The rest of the skills, i.e. nodes, in the knowledge graph require a subset of these four skills as prerequisite skills. The details of these skills and videos of training are included in the supplement[3].

Both conditions followed the exact same procedure as above, *except for the training session*. During training, the experimental condition was trained with our algorithm which adapted the number of tasks to sample per skill and the next skill to transition to using BKT. In contrast, the control condition was provided with a curriculum designed by our experts who all had experience in training novices for EchoArena. Both the curriculum by the experts or the curricula from our algorithm are constructed from the same knowledge graph. The control condition self-predicted its skill acquisition and manually transitioned to the next skill in the expert-designed curriculum.

After completing each task, the control condition was asked in VR for their (i) binary self-prediction (i.e. acquired/ not acquired) on the acquisition of the current skill, and (ii) whether to transition to another skill. Either until the participant decided to transition (control) or BKT predicts acquisition (experimental), tasks were iteratively sampled from associated probabilistic program and generated in VR. Because the periodic self-prediction inquiry takes up a small portion of training time, we also asked the experimental condition to also self-predict after each task for fairness, even though it is not used. During training we collect the following data. After subjects completes each task, we record the task name, binary task evaluation result (i.e. correct/ incorrect), BKT's prediction, and binary self-prediction prediction, and time when the data are collected.

5.5 Measurements

Skill Acquisition Prediction Error This error is computed using the difference between the expected and the actual post test scores for the skills that are predicted to be acquired by either BKT or

[2]Video recordings of each session can be found in this link.

[3]Video recordings of training for each of the 10 skills can be found in this link.
To observe how fine and gross motor movements are used for these skills, refer to basic and advanced tutorial videos in this link.

self-prediction, i.e. $M - \sum_{i=1}^{M}$(# of correctly solved tasks for skill i) / (N tasks used to evaluate skill i) where M is the number of tasks that are predicted to be acquired by either BKT or self-prediction. Refer to Sec. 5.4 for how N is selected per skill.

Learning Gains A learning gain for a participant is computed by one's score improvement (i.e. post test - pre test scores), where the pre and post test scores are computed in the following way: $\sum_{i=1}^{K}$(# of correctly solved tasks for skill i) / (total # of tasks used to evaluate skill i), where K is the number of skills to train. In this study, K=10, and each skill is evaluated with a variable number of tasks (refer Sec. 5.4).

NASA Task Load Index We use the NASA task load index (TLX) [21] to measure a subject's subjective mental workload for the skills we train, before and after the training session. To measure the improvement in the subjective task load, we compute (TLX score after training - TLX score before training). A skill acquisition with respect to a task distribution should ideally mean that a user can solve all tasks in the distribution. Thus, if a skill is acquired, we should expect the subjective task load to decrease with respect to the task distribution.

Custom User Experience Survey We use our custom user experience survey to evaluate subjects' training experience for both conditions. The survey starts with an open question inquiring for any negative experience with the training followed by the three Likert 5-Point scale questions.

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

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. We focus on these three aspects of training because many theories in learning sciences and psychology support the idea that *incremental difficulty* is important for engagement in learning new skills [1, 3, 7, 10].

Statistical Significance Test We use Mann-Whitney's U test using Python Scipy's stats package [25] for all the statistical significance tests reported in the Results Section. We choose this test because the sample size is too limited to expect normal distributions to hold for unpaired t-test.

Correlation We compute the Pearson correlation [13] to compute any correlations.

5.6 Results

Skill Acquisition Prediction Accuracy Our results show that BKT has lower average prediction error than self-prediction, as visualized in Fig. 7 (right). BKT overestimates participants' skill acquisition by 28.21 ± 13.06%, whereas the control overestimate by 34.81 ± 23.67%. However, these two distributions of prediction errors is not statistically significant (p-value 0.46). The experimental condition has a noticeably higher correlation coefficient of 0.96 (p-value < 0.01) than the self prediction's 0.59 (p-value 0.09).

Learning Gains Prior to comparing the learning gains between the two conditions, we check whether there is any imbalance in the prior skills between the two conditions. The difference in the distributions of the pre-test scores is not statistically significant (p-value < 0.05). Regarding learning gains, the experimental group outperform the control group on average with statistical significance (p-value < 0.05) as shown in Fig. 7 with an effect size of 0.41. On average, the control group improves 22.96±12.90% in learning gains, whereas the experimental group improves 30.37±5.97%. We observe that the standard deviation of the experimental condition is reduced by 53.7% than the control's.

Subjective Task Loads Despite the higher average learning gains, the experimental condition does not result in lower subject task load after training than the control. Recall that lower TLX score is preferable because it means the subjective task load has decreased. The experimental condition shows a mild average decrease in NASA TLX scores by 6.56 ± 16.00%, while the control exhibits a medium average decrease by 17.56 ± 11.77%. However, the difference in the distributions of TLX score improvements is not statistically significant (p-value 0.07).

User Experience After post tests, we equally ask both conditions for their user experience with training, using our Likert 5-point scale questionnaire related to engaging, incrementally difficult, and helpfulness in learning new skills. Both conditions positively rate their training experience as plotted in Fig. 8, averaging approximately 4.5 out 5 points for all three aspects. Mann-Whitney U test show that the differences in distributions across conditions for engagement, incremental difficulty, and helpfulness are not statistically significant, reporting p-values of 0.86, 0.43, and 0.34, respectively.

We also ask both conditions for any negative experiences with the training in general. While the control condition does not share any negative feedback, four out of nine participants in the experimental condition (we denote participants as E1-E9) report negative experiences particularly with the skill transitions. Some participants share frustration from too many assigned practices for a specific skill: *"I got frustrated towards the end because I was stuck in a task"* (E3) and *"getting stuck in a task was a bit frustrating in the beginning, but frustration went down as I saw myself improving"* (E5). On the contrary, some report premature transitions: *"sometimes, the training algorithm transitioned you a bit earlier than you expected"* (E6) and *"during the training, I thought I still needed some more practice, but during evaluation I actually performed better than I expected"* (E1).

Generated Curriculum The comparison between the control condition's expert-designed curriculum and the algorithm's personalized curricula is visualized in Fig. 9. These curricula are all constructed from the same knowledge graph shown in Fig. 6.

6 DISCUSSION

In this section, we analyze our results in relation to our hypotheses (H1-H3) as stated in Sec. 5 in Sec. 6.1. Based on observations of our study, we suggest directions to improve observed shortcomings of BKT in Sec. 6.2.

6.1 Analysis

H1: Prediction Accuracy for Skill Acquisition Contrary to our hypothesis, BKT is not more accurate than self-prediction. Although the results show that the BKT's skill acquisition prediction error (28.21 ± 13.06%) is lower on average than the self-prediction's

	Measures	Results
H1	1. Skill Acquisition Prediction Error 2. Correlation	1. The average predictions errors were not statistically different. 2. However, BKT predictions are substantially more highly correlated to skill acquisition.
H2	Learning gains	EC has higher and more consistent average learning gains.
H3	NASA TLX	Despite higher average learning gains, EC do not result in higher average improvements in subjective task load after training.
R4	Custom User Experience Survey	Both conditions report positive user experience with no statistical difference. However, EC reports some negative user experience.

Table 1: This table maps the measurements listed in Sec. 5.5 to the summary of results in Sec. 5.6. The three hypotheses (H1,2,3) are explained in Sec. 5. EC stands for experimental condition. Assume that EC is being compared to the control condition.

(34.81 ± 23.67%), there is no statistically significant difference. Instead, we observe a different outcome where *BKT noticeably lowers fluctuations* in skill acquisition prediction errors. The standard deviation of the BKT's prediction errors is nearly 50% lower than the self-prediction's. The two box plots shown in Fig. 7 visually contrasts this difference and its impact on learning gains. The left box plot compares the skill acquisition prediction errors between the control (brown) and the experimental (pink) groups. We observe that the self-prediction's prediction error ranges from as low as nearly 0% to over 70%, whereas the experimental's is considerably more consistent. Consequently, due to its consistency, BKT's predictions are 37% more highly correlated to skill acquisition than the self-prediction's. Since skill acquisition prediction directly controls skill transition, it directly impacts learning gains. Indeed, on the right plot concerning learning gains, we identify the similar trend. The learning gains for the control group ranges as low as 0% (i.e. learned nothing) to 50%, while the experimental's are noticeably more consistent.

We also note that BKT's high correlation does not derive from underestimation of skill acquisition, thereby providing over-practices to train a fewer number of skills. The control and the experimental conditions trains for on average 7.78 ± 1.98 and 7.56 ± 2.51 skills, respectively, out of 10 skills during 25 minutes of training session, with no statistical difference. These numbers represent how many skills self-prediction and BKT predicted the users to have acquired and, thus, transitioned them to train for the next skills. They do not represent how many skills the users actually acquired in each group. The analysis of these prediction errors are analyzed above. In short, BKT exhibits much more consistent predictions of skill acquisition than self-prediction without noticeable underestimation of skills.

These disparity in the consistency of predictions becomes clearer as we contrast the processes involved in the two methods. Ideally, skill acquisition with respect to a task distribution means that the user can solve all tasks in the distribution. For self-prediction, this means that each user needs to: first, accurately approximate the task distribution from experiencing sampled tasks. Second, the user needs to accurately assess confidence in solving tasks *with respect to one's estimated mental model of the the task distribution.* This approximation of the task distribution likely becomes challenging as task complexity increases.

On the contrary, for modeling BKT per skill, domain experts tunes BKT parameters (refer to Sec. 3.1) using their understanding of the task distribution *that they designed* and mental models of novice students' learning processes. As their mental student models may be biased, we recruit three instructors to reduce the bias. Hence, BKT is better informed of the complexity of task distribution than self-prediction. Nevertheless, the sources of errors for BKT could derive from experts' bias in mental models of students and BKT's lack of consideration for physical factors.

H2: On Learning Gains The results are visualized in Fig. 7 (left). The experimental condition exhibits higher average learning gains than the control condition (p-value < 0.05), with an effect size of 0.41. Much more noticeable is the consistent learning outcome with the experimental condition, whose standard deviation of learning gains (±12.90%) is 53.7% lower than the control condition's (±5.97%). This shows that our algorithm considerably lowers fluctuations in the learning outcomes than the control, reducing the chance of users falling behind. However, in trade-off, this also means that our algorithm could potentially stifle the learning of exceptional users who may benefit more from self-guided learning. Although an outlier, we observe the highest learning gain is achieved by a user in the control condition in Fig. 7 (right plot). Hence, the utility of our BKT-driven personalization may be higher in domains where users are expected to all exceed certain levels of learning gains. This may include safety-related training such as in healthcare, first responders, construction, manufacture, etc., where VR has been utilized for training [59].

There are two compounding factors that likely contributed to the higher learning gains of the experimental condition. First, although not statistically significant, BKT has a lower average prediction error (28.21 ± 13.06%) than self-prediction (34.81 ± 23.67%). Second, our algorithm's personalized curriculum generation (refer Sec. 4.3) automatically skips over skills that a user already acquired from prior experience and focuses on skills not acquired. In contrast, users in control condition manually progress through a fixed (non-personalized) curriculum and need to decide whether to transition to the next skill. Fig. 9 visualizes the differences in curricula between conditions. This personalization results in more diverse curricula in the experimental condition as highlighted in blue, in contrast to a single curriculum the control adhered to. The combination of these two factors contributed to the experimental group's efficient training time allocation to achieve higher learning gains.

H3: On Subjective Task Loads Counter-intuitively, despite higher average learning gains, the experimental condition does not result in lower average subjective task load than than control after training. Recall that it is desirable to lower subjective task load (i.e. lower

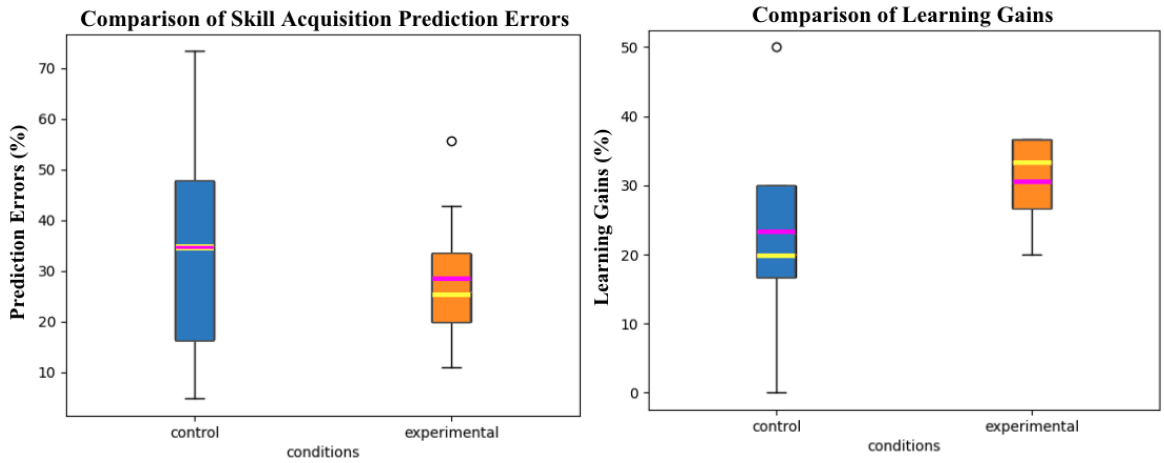

Figure 7: The left box plots compare the BKT and the self-prediction's average errors in predicting psychomotor skill acquisition. The right box plots compare the average learning gains between the two conditions. The pink line in the box plot represents the average and the yellow line, the median.

NASA TLX score) through training (refer Sec. 5.5). The control condition decreases in the NASA TLX score by 17.56 ± 11.77% on average, whereas the experimental only decreases by 6.56 ± 16.00%. Although the difference is not statistically significant, we observe the average to actually lean more favorably towards the control. This discrepancy in the learning gains and subjective task load has been observed in academic learning setting (e.g. [48]) as well, where the condition achieving the highest objective learning gains also counter-intuitively result in the highest subjective task load. The following user experience results explain the potential cause of this discrepancy.

R4: On User Experience As pointed out in the verbal interview, this potential cause of the discrepancy derives from our use of BKT predictions to control skill transitions. Recall that our algorithm trains a skill until its BKT predicts skill acquisition and only then transitions a user to the next skill. While the control group reported no negative experience with training, nearly half of the experimental condition complained of the algorithm's skill transitions. The participants in the experimental condition (E3,E5) repeatedly use the word "stuck" to share their frustration from *not being able to stop excessive training* on a particular skill. Also, E1 and E6 complain that the system *prematurely transition* them to training new skills even though they do not feel prepared to move on. The algorithm's disregard of users' mental state when determining skill transition is a potential cause of the observed discrepancy.

Although the experimental condition complained of specific occurrences in training, in general, both conditions positively reported to our custom user experience survey. As shown in Fig. 8, both conditions responded that the training was engaging, incrementally difficult, and helpful for learning new skills, averaging around 4.5 out of 5 across all three aspects, with no statistically significant difference. The high average user ratings for the experimental condition along with the learning gains outcome show that BKT could be integrated with an existing personalized curriculum generation algorithm. More importantly, this indicates that BKT may be used in the current adaptive training ecology where curriculum is often personalized.

6.2 Our Suggestions to Improve BKT

The key insight to take away from our work is that, when training with task distributions, BKT is more reliable design component than self-prediction for psychomotor skill acquisition prediction. However, our study results also reveals BKT's shortcomings related to (a) errors in BKT predictions and (b) the use of the predictions to control skill transitions. We suggest directions to improve BKT's accuracy and its usage in this section.

Suggestions to Improve BKT Predictions the canonical formulation of BKT needs to be extended with more variables relating to influential physical factors such as fatigue. During pre- and post-test phase of our study, we frequently observe participants failing to solve tasks in post-tests, which they were able to solve correctly before in the pre-test. We conjecture that the accumulation of visual fatigue from exposure to VR and physical fatigue from exertions may have induced the outcomes we observed. However, further investigation is necessary.

Suggestions to Improve Skill Transitions Our study reveals two issues regarding our algorithm's use of BKT's predictions for skill transitions: (a) premature transition before the users feel confident with a skill and (b) frustration from excessive practices of a same skill. To prevent premature transitions, it may be appropriate to probe and incorporate the user's self-prediction of the current skill *after BKT predicts skill acquisition*. If the user is not confident, then more tasks should be sampled and generated in VR until the user is confident. This way, we can align the BKT's prediction with the user's subjective confidence. However, this comes at the risk of, in the worst case, a consistent underestimation of skill acquisition, resulting in redundant training due to the user's low confidence. For this reason, it may be reasonable to explore effective ways to *share the BKT's estimate of skill acquisition with the user during training*

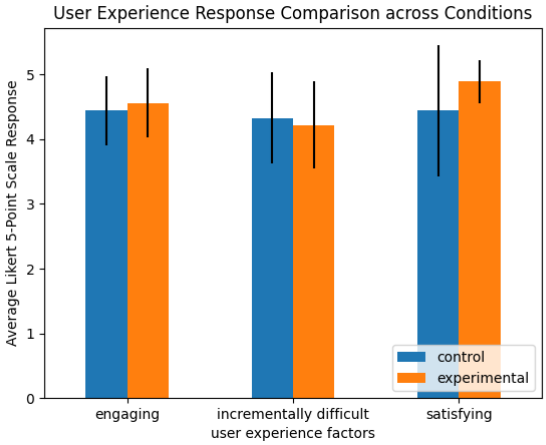

Figure 8: The bar plots compare the user experience on the training curriculum with respect to engaging, incrementally difficult, and helpfulness in learning new skills across the two conditions.

(e.g. a bar graph in percentage representing the skill acquisition in VR). This way, users can align their self-prediction with the BKT's.

To lower users' frustration from excessive practices, there are important factors to consider. Recall that BKT parameters (refer Sec. 3.1) for each skill are tuned *with the assumption that its pre-requisite skills are already mastered*. Hence, if the algorithm carelessly transition the user to a new skill that requires the current one to avoid frustration, this violates the BKT's assumption and, therefore, degrades its prediction accuracy. Furthermore, this transition would also likely overload the user to simultaneously learn the pre-requisite and new skills, potentially incurring more frustration. To circumvent these issues, scaffolding [42] a skill could help users master each skill before transitioning to the next skill, while lowering frustration. This means to use domain knowledge to divide the associated task distribution to the skill into different sections of according to difficulty, and sample from relatively easier section to assist learning. However, this scaffolding may be labor intensive in trade-off.

7 LIMITATIONS & FUTURE WORK

There are a number of limitations in our study.

Too much variability degrades learning: Although introducing variability in training tasks has been shown to induce better learning and generalization of psychomotor skills [4, 11, 52, 58], too much variability in training can actually impair learning [5]. In our study, we assume that the instructors who design the task distributions would introduce adequate amount of variations to train each skill. We do not have any mechanism in place to measure and determine whether the size of variations in the provided task distribution would negatively impact learning.

Extracting Tacit Knowledge: It can be challenging to extract *tacit* domain knowledge from the experts to specify accurate evaluation metrics per task distribution as well as pre-requisite relations among skills. We do not experience this issue in our study, but we foresee this may be an issue depending on the skill to train. We have not investigated a methodological approach to cope with this difficult problem.

Tangled Effects in Our Study: And, in our algorithm, the effects of personalizing the curriculum and the BKT-driven skill transitions are *jointly* taking place. To better evaluate the isolated effect of the two independent variables, further ablation study is necessary.

Limited Number of Subjects Recruiting participants with pre-requisite background in dynamic VR activities was a major difficulty in this study. This is due to the highly dynamic nature of EchoArena VR esports, where participants need to fly around in zero gravity space. In fact, even with the pre-requisite background, some participants (Sec. 5.3) were excluded during the study because they suffered from motion sickness. In the future, we plan to expand our study with more participants in augmented, not virtual, reality where motion sickness is much less induced.

Authoring a Probabilistic Programming Language The scope of this study did not include whether other developers or researchers can easily write scenarios with the probabilistic programming language that we used in this study. This is to be explored in the future.

Designing a Task Distribution of the Same Difficulty We assume that the experts are capable of designing a task distribution of the same difficulty to operationalize BKT (Sec. 3.2). We did not investigate the challenges as experts design such task distributions nor how to handle potential variations in task difficulty with the distribution when updating BKT predictions.

Limited Haptics in VR The lack of or limited haptics (e.g. weight, tactile) in VR introduces a gap in virtual training environment compared to reality. If we are to train for a skill to be used in reality, it is likely that this gap would impede the direct transfer of skills acquired in VR to reality. Therefore, VR may not be used to directly learn how to solve a task in reality. However, it can be used to learn to solve its subtasks, thereby expediting the training in reality. For example, suppose we train a user how to accurately pass a baseball to a running teammate. The task consists of subtasks which could be trained in VR (prior to training in reality), such as (i) perceiving how fast the teammate is running and how far away the teammate is, (ii) identifying how much ahead of the teammate one should throw for the teammate to receive, (iii) determining how fast to throw, and (iv) learning when to release the ball as one coordinates joint movements to throw the ball to the teammate. VR could generate potentially many more variations of tasks than an actual human teammate would in reality to help a user learn to solve the subtasks. Then, the user could further train in reality to cope with weight of the ball as one throws it.

Potential Applications to Other Immersive Platforms Our algorithm may be applicable to other immersive platforms such as Kinect. We leave this exploration for future work.

8 CONCLUSION

We investigate the effectiveness of BKT to train psychomotor skills with distributions of tasks. In particular, we examine the accuracy of BKT to algorithmically predict mastery of a skill accounting for the complexity of the distribution of training tasks. Furthermore, we study the effects of utilizing BKT predictions to control skill

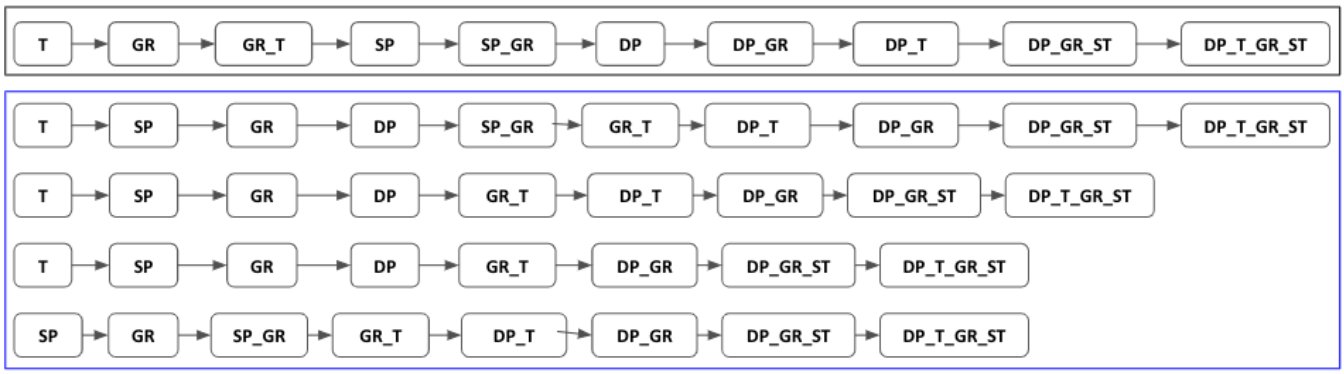

Figure 9: A comparison in curricula between the control and the experimental conditions. The small black boxes represent skills to train, and the arrows, the order of skills to train. The control condition's curriculum is highlighted with a large blackbox, whereas the experimental condition's curricula are highlighted in blue.

transitions on user experience. Our study shows that BKT is an effective design component than self-assessment to predict skill mastery with respect to a task distribution. However, the study also reveals that solely relying on BKT predictions to control skill transitions could incur frustrations or deter improvements in subjective task loads. We hope our findings serve as a foundation for HCI community to design VR-based personalized psychomotor training algorithms that are cognizant of the complexity of training task distributions.

REFERENCES

[1] Sami Abuhamdeh and Mihaly Csikszentmihalyi. 2012. The importance of challenge for the enjoyment of intrinsically motivated, goal-directed activities. In *Personality and Social Psychology Bulletin*.

[2] 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*.

[3] John W. Atkinson, Jarvis R. Bastian, Robert W. Earl, and George H. Litwin. 2012. The achievement motive, goal setting, and probability preferences. In *Personality and Social Psychology Bulletin*.

[4] Daniel A. Barun, Ad Aertsen, Daniel M. Wolpert, and Carsten Mehring. 2009. Motor Task Variation Induces Structural Learning. *Current Biology* 19 (2009). Issue 4.

[5] Marco Cardis, Maura Casadio, and Rajiv Ranganathan. 2018. High variability impairs motor learning regardless of whether it affects task performance. In *Journal of Neurophysiology*.

[6] Nuno N. Correia, Raul Masu, William Primett, Stephan Jurgens, Jochen Feitsch, and Hugo Placido da Silva. 2022. Designing Interactive Visuals for Dance from Body Maps: Machine Learning and Composite Animation Approach. In *Designing Interactive Systems (DIS)*.

[7] Mihaly Csikszentmihalyi. 1991. *Flow: The Psychology of Optimal Experience*. HarperPerennial.

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

[9] Ilse R. de Boer, Maxim D. Lagerweij, Paul R. Wesselink, and Johanna M. Vervoorn. 2005. The Effect of Variations in Force Feedback in a Virtual Reality Environment on the Performance and Satisfaction of Dental Students. In *Aviation, Space, and Environmental Medicine*, Vol. 76. 352–356.

[10] Edward L. Deci and Richard M. Ryan. 2000. The "What" and "Why" of Goal Pursuits: Human Needs and the Self-Determination of Behavior. *Psychological Inquiry* 11, 4 (2000), 227–268. https://doi.org/10.1207/S15327965PLI1104_01

[11] A. K. Dhawale, Maurice A. Smith, and B. P. Olveczky. 2017. The Role of Variability in Motor Learning. *The Annual Review of Neuroscience* 40 (2017), 479–498.

[12] Pedro Faria. 2023. figma: Web Client/Wrapper to the 'Figma API'. https://www.figma.com/

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

[14] D. Fremont and et al. 2019. Scenic: A Language for Scenario Specification and Scene Generation. *Programming Language Design and Implementation (PLDI)* (2019).

[15] 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).

[16] 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.

[17] Google. 2023. Tilt Brush.

[18] M. Graafland, J. M. Schraagen, and M. P. Schijven. 2016. Systematic review of serious games for medical education and surgical skills training. In *British Journal of Surgery*, Vol. 99. 1322–1330.

[19] 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

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

[21] Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In *Human Mental Workload*, Peter A. Hancock and Najmedin Meshkati (Eds.). Advances in Psychology, Vol. 52. North-Holland, 139–183. https://doi.org/10.1016/S0166-4115(08)62386-9

[22] Hector Hernandez-Davo, Tomas Urban, Jose M. Sarabia, Casto Juan-Recio, and Francisco J. Moreno. 2014. Variable training: effects on velocity and accuracy in tennis serve. In *Journal of Sports Sciences*, Vol. 32. 1383–1388.

[23] Discord Inc. 2020. Discord. https://discord.com

[24] 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*.

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

[26] K.R. Koedinger, A.T. Corbett, and C. Perfetti. 2012. The knowledge-learning-instruction framework: bridging the science-practice chasm to enhance robust student learning. *Cognitive Science* 36, 5 (2012), 757–98. https://doi.org/10.1111/j.1551-6709.2012.01245.x

[27] Echo Arena VR Master League. 2022. Statistics. Retrieved January 15th, 2022 from https://vrmasterleague.com/EchoArena/Stats

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

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

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

[31] 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.

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

[33] 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.

[34] Martin Ludvigsen, Maiken Hillerup Fogtmann, and Kaj Gronbaek. 2010. TacTowers: An Interactive Training Equipment for Elite Athletes. In *Designing Interactive Systems (DIS)*.

[35] Randi Q. Mao, Lucy Lan, Jeffrey Kay, Ryan Lohre, Olufemi R. Ayeni, Danny P. Goel, and Darren de SA. 2021. Immersive Virtual Reality for Surgical Training: A Systematic Review. *Journal of Surgical Research* 268 (2021), 40–58. https://doi.org/10.1016/j.jss.2021.06.045

[36] Brian Milch and Stuart Russell. 2006. First-Order Probabilistic Languages: Into the Unknown. In *International Conference on Inductive Logic Programming*.

[37] María Lucila Morales-Rodríguez, Apolinar Ramírez-Saldivar, Arturo Hernández-Ramírez, Julia Patricia Sánchez-Solís, and José Antonio Martínez Flores. 2012. Architecture for an Intelligent Tutoring System that Considers Learning Styles. *Res. Comput. Sci.* 47 (2012), 37–47. https://api.semanticscholar.org/CorpusID:16379960

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

[39] 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 *International Conference on Intelligent Tutoring Systems (ITS)*.

[40] Martin Odersky, Lex Spoon, and Bill Venners. 2008. *Programming in Scala*. Artima.

[41] S. Pastel, K. Petri, C. Chen, A. Caceres, M. Stirnatis, C. Nübel, L. Schlotter, and K. Witte. 2023. Training in virtual reality enables learning of a complex sports movement. *Virtual Reality* 27 (2023), 523–540.

[42] K. Ann Renninger and Alexandra List. 2012. Scaffolding for Learning. *Encyclopedia of the Sciences of Learning* (2012), 2922–2926.

[43] 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.

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

[45] M. Schijven and J. Jakimowicz. 2003. Virtual reality surgical laparoscopic simulators. In *Surgical Endoscopy And Other Interventional Techniques*.

[46] 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.

[47] Kenneth J. Stroud, Deborah L. Harm, and David M. Klaus. 2005. Preflight Virtual Reality Training as a Countermeasure for Space Motion Sickness and Disorientation. In *Aviation, Space, and Environmental Medicine*, Vol. 76. 352–356.

[48] Daniel Szafir and Bilge Mutlu. 2013. ARTFuL: Adaptive Review Technology for Flipped Learning. In *Conference on Human Factors in Computing Systems (CHI)*.

[49] 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*.

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

[51] Fong Wee Teck. 2012. Force and Torque Simulation in Virtual Tennis. In *Proceedings of the Workshop at SIGGRAPH Asia* (Singapore, Singapore) *(WASA '12)*. Association for Computing Machinery, New York, NY, USA, 143–146. https://doi.org/10.1145/2425296.2425321

[52] E. Thorp, K. Kording, and F. Mussa-Ivaldi. 2017. Using noise to shape motor learning. *Journal of Neurophysiology* (2017).

[53] 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*.

[54] Dishita Turakhia, Andrew Wong, Yini Qi, Lotta-Gili Blumberg, Yoonji Kim, and Stefanie Mueller. 2021. Adapt2Learn: A Toolkit for Configuring the Learning Algorithm for Adaptive Physical Tools for Motor-Skill Learning. In *Designing Interactive Systems (DIS)*.

[55] Marijke Vandermasesen, Tom De Weyer, Peter Feys, Kris Luyten, and Karin Coninx. 2016. Integrating Serious Games and Tangible Objects for Functional Handgrip Training: A User Study of Handly in Persons with Multiple Sclerosis. In *Designing Interactive Systems (DIS)*.

[56] 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)*.

[57] 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*.

[58] H. G. Wu, Y. Miyamoto, L. Castro, B Olveczky, and M. Smith. 2014. Temporal structure of motor variability is dynamically regulated and predicts motor learning ability. *Nature Neuroscience* 17 (2014), 312–321.

[59] Biao Xie, Huimin Liu, Rawan Alghofaili, Yongqi Zhang, Yeling Jiang, Flavio Destri Lobo, Changyang Li, Wanwan Li, Haikun Huang, Mesut Akdere, Christos Mousas, and Lap-Fai Yu. 2021. A Review on Virtual Reality Skill Training Applications. In *Frontiers in Virtual Reality*, Vol. 2.

[60] 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)*.

[61] Zhengyou Zhang. 2012. Microsoft Kinect Sensor and Its Effect. *IEEE MultiMedia* 19, 2 (2012), 4–10. https://doi.org/10.1109/MMUL.2012.24