Optimizing Mastery Learning by Fast-Forwarding Over-Practice Steps

Anonymous Author(s)

Anonymous Institution(s)

Abstract. Mastery learning improves learning proficiency and efficiency. However, the overpractice of skills–students spending time on skills they have already mastered–remains a fundamental challenge for tutoring systems. Previous research reduced overpractice through better problem selection algorithms authoring focused practice tasks. However, few efforts have focused on reducing overpractice through step-level adaptivity, which can avoid resource-intensive curriculum redesign. We propose and evaluate Fast-Forwarding as a technique that enhances existing problem selection algorithms. Based on simulation studies informed by learner models and problem-solving pathways from student data, Fast-Forwarding can reduce overpractice by up to one-third by not requiring students to complete problem-solving steps if all remaining pathways are fully mastered. Fast-Forwarding is a flexible method that enhances any problem selection algorithms, though its effectiveness is highest for selectors that preferentially serve difficult problems. Therefore, our findings suggest that while Fast-Forwarding may improve student practice efficiency, the size of its practical impact may also depend on students' ability to stay motivated and engaged at higher difficulty levels.

Keywords: mastery learning, knowledge tracing, problem selection, intelligent tutoring systems, data-driven optimization

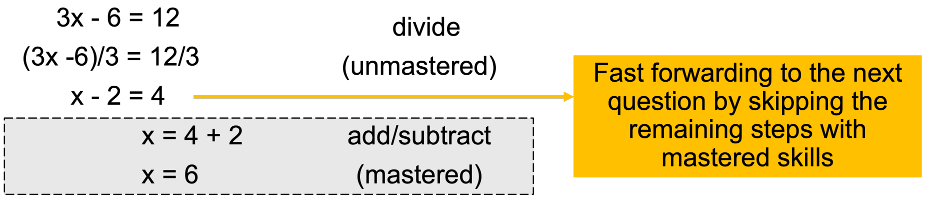

Fig. 1: Fast-Forwarding reduces overpractice by avoiding problem-solving paths students have fully mastered. In the example, linear equation-solving steps in gray are known to be mastered by the student and can be fast-forwarded to reduce overpractice.

1 Introduction

The field of technology-enhanced learning has succeeded in optimizing problem selection strategies and improving learning efficiency through methods such as Bayesian Knowledge Tracing (BKT) [9]. BKT, one the most widely used knowledge tracing algorithms, has proven effective in modeling student knowledge acquisition, enabling tutoring systems to adaptively select practice problems with relevant and adequately difficult skills. In addition, several extensions to BKT have been proposed, including deep knowledge tracing- (e.g., [31, 35, 39] and regression-based approaches such as AFM and PFA (e.g., [30, 13, 34]).

Adaptive problem selectors are informed by knowledge tracing and grounded in mastery learning [5]. Mastery learning assumes that learning and instruction can be divided into atomic skills that students acquire by exercising distinct cognitive operations and receiving feedback from which they learn. This idea aligns with large-scale evidence from several data sets [20] and has improved student performance at scale [32]. Mastery learning underlies the design of many contemporary tutoring systems and the idea that students require several practice repetitions–learning opportunities–to master each skill. On average, they require seven [20]. However, when tasks or activity sequences are poorly designed, practice activities might lead to *overpractice* and *underpractice* of individual skills.

Overpractice occurs when students practice skills they have already mastered, which is undesirable for learning efficiency if students have not yet mastered other skills [8]. Overpractice typically occurs when students practice static problem sets where problems include multiple skills, and there is little variability in what skills the problems target. In contrast, underpractice can be due to limited problem pools, meaning students do not master content upon completing a problem set or insufficient time to practice. While curricular redesign based on learning analytics has been suggested as one method to mitigate over- and underpractice [8], avoiding both remains a fundamental challenge in adaptive learning, with learning analytics frameworks on the data-driven redesign of adaptive learning systems considering problem selection a core dimension for design improvement [17]. In particular, the consequences of reducing overpractice on learning are known to be positive: empirical research has shown that reducing overpractice in CognitiveTutor geometry decreased practice time without impacting learning outcomes [8], and such reductions have thus far required a careful and labor-intensive curricular redesign. In the present study, we study whether overpractice can be reduced through more automated means, based on optimization of step-level adaptivity ("inner loop" [4]) and having the system take over the mastered steps under some conditions. Such optimization balances the allocation of repetitions, giving students time to master challenging skills.

One reason why the challenge of overpractice is not yet solved is that serving practice problems with a single skill is not a solution to the problem, as multi-step problems give learners important opportunities to learn, most notably when completing two skills together is more challenging than practicing them in isolation: the composition effect [14]. Therefore, it is important to develop novel ways for students to master the skills required to learn while optimizing practice time *and* retaining multi-step problem solving.

An emerging approach to improve adaptive problem selection efficiency is called *focused practice*. This approach involves isolating specific skills that the learner currently lacks. Recent problem selection algorithms consider the overall redundancy in mastery of knowledge and the degree of isolation of unmastered skills in problem selection [15]. However, this approach still necessitates the availability of a diverse pool of problems with sufficient breadth and variation in skill density. Additionally, the requirement of designing focused tasks that exercise a limited number of skills necessitates additional content authoring time.

Can a general-purpose, problem pool agnostic, and selector agnostic method for focus practice increase student learning efficiency as measured in overpractice? We propose Fast-Forwarding (Fig. 1), allowing learners to complete multistep problems up until the point all remaining problem-solving steps are mastered and would hence only contribute to overpractice. Compared to prior approaches to focused practice [17, 8], Fast-Forwarding **neither requires a specific problem selection mechanism nor depends on curricular redesign** while retaining multi-step problem solving. We quantify the effectiveness of Fast-Forwarding across various contemporary and legacy problem selection methods in intelligent tutoring systems through simulation. We ask:

> RQ1: How much does learning efficiency as measured in overpractice benefit from Fast-Forwarding in a mastery problem selector given limited practice time?
>
> RQ2: How much overpractice does Fast Forwarding reduce until content mastery across different problem selection regimes?

To answer these RQs, we conducted experiments through simulation based on solution pathways and parameters estimated from open-source data sets of students practicing equation solving in a tutoring system. We chose equation solving as a domain due to its recursive nature: foundational skills are often repeated at the end of solving for X. However, we note that the **methods studied here apply to numerous other domains of instruction with step-level problem solving, for which tutoring systems are available** [3]. In STEM problem solving, science domain decisions tend to happen first (e.g., selecting physics principles), while later steps involve algebra or arithmetic, which many students may have mastered. In all of these domains, fast-forwarding could remove overpracticed steps by completing mastered steps on behalf of the learner.

2 Related Work

2.1 Learner Models and Problem Selectors

Various learner models have been developed to model students' learning and problem-solving behaviors. Bayesian knowledge tracing, arguably the most widely used model in adaptive learning systems today, models learners' latent knowledge of distinct skills as a probability of knowing each [9]. Other dynamic probabilistic models have been proposed to model the dependency of different learning concepts. For example, Learning Factor Analysis [7] modeled learner knowledge states via logistic regression to deal with learning concepts at different levels, and Performance Factor Analysis [30] fits distinct learning rates for correct and incorrect responses. Other models, such as deep knowledge tracing, use neural network architectures to model and predict learner performance [31]. These learner models measure learners' mastery of different skills and how they change with given practice opportunities.

Many problem selection algorithms use estimated content mastery in adaptive learning in diverse ways. Legacy mastery learning algorithms used in Cognitive Tutor either select the problem with the most unmastered skills (hard) or problems with the least, but at least one, mastered skills (easy) [2, 19]. However, all these selection algorithms, while generally reducing student practice time [32], are limited in that they require a static pool of content and skills. Few notable exceptions exist, where tutoring systems have been designed to adaptively take over specific, commonly overpracticed steps [33, 21]. Other researchers proposed to supplement problem sets with focused tasks to reduce overpractice, which are problems that target few skills simultaneously. They particularly designed practice problems that involve only a focused set of skills that students have not mastered and then selected these focused tasks with higher probability [16, 17]. However, designing focused tasks is time-consuming and requires content authoring. Fast-Forwarding, the method proposed in this study, in contrast, can be applied to any existing question pool and dynamically monitors whether certain steps can be skipped based on mastery estimation to mitigate overpractice.

2.2 Learner Simulation

Simulating learner data to study learning processes and outcomes is a core method in the field, particularly where obtaining real-world data is challenging, expensive, or impractical [10]. These simulations leverage cognitive learning theories to generate performance patterns that can be used to refine instructional methods. For instance, past research has employed simulated learners to identify skill models based on performance patterns, which were subsequently validated on real learner data, leading to improved instruction tailored to these identified skills [24]. Similar refinements have been applied to analyzing error types [38]. Recent research, using Generative Adversarial Networks (GANs), has also been applied to enhance the assessment of student knowledge through simulation: Zhang et al. [39] augmented log data from tutoring systems using GANs, which improved the reliability of knowledge estimates.

However, despite these advances in simulations for refining knowledge tracing and instruction, there is a lack of simulated learner experiments that quantify the impact of problem selection algorithms on practice time and efficiency. Instead, previous work has primarily focused on simulating the effects of setting mastery thresholds for error types [11] or on performance prediction given different mastery levels [29]. In addition, validating simulated learners on outcomes of interest remains a key research challenge. According to a recent literature review by Käser and Alexandron ([18]), a major issue in previous research using simulated learner data is the lack of systematic studies on the validity of synthetic data. However, there are exceptions, such as the work that used simulated student data in tutoring systems to predict real student performance [27]. Similarly, recent studies have increasingly used synthetic data to improve predictive models and predictive validity [39]. Our research simulates the practice efficiency of different problem selection strategies in tutoring systems. We incorporate empirical estimates of learning rates and solution pathways into our simulations to improve validity.

3 Methods

3.1 Study Context and Dataset

We utilize an open source dataset from APTA, a collaborative tutoring system for middle school equation-solving [37] available on PSLC DataShop (datasets #5549 and #5604). [22]. The study sample consists of data from IRB-approved classroom studies conducted in math classes at two public suburban middle schools in a mid-sized city in the eastern U.S. Overall, 164 6th to 8th grade students engaged with a series of multi-step linear equation-solving problems.

We analyzed the data comprising 10,937 entries (N = 10,937) of completed and timestamped problem-solving steps using the DataShop student step rollup tool [22]. The log data provide a detailed description of the student solution paths, meaning the order of operations used to solve individual problems. This allows for a fine-grained analysis of skills (e.g., add constant) students employ to derive solutions, offering deeper insights into their step-level problem-solving processes and serving as reference pathways for our simulation studies.

We only considered students' first attempts since they reflect the students' initial response to solving linear equation steps without tutor assistance, with steps exhibiting hint usage being classified as incorrect attempts [22].

3.2 Computation of Overpractice

We define overpractice on a skill as the number of practice opportunities (e.g., problem-solving steps) on a skill given a predicted error rate on that skill below 0.05, in line with the mastery criteria in the Lynnette tutor [26]. Table 1 reveals several key findings regarding student performance and practice behavior across different skills. **This analysis served as a baseline for how much overpractice was present in our data set with real students.** Simulations, as described below, where then used to determine how much students could potentially save if they would have unlimited practice time and under different, alternative problem solving regime scenarios.

Table 1: Student mastery rates, practice and overpractice opportunities for each skill based on student data. The first row shows the % of students who mastered each skill, the second row the average number of problem-solving steps per skill, and the third row the number of steps taken after mastery (overpractice).

	cancel-const	cancel-var	comb-const	comb-var	division-complex	division-simple	simplify-division	add/subtr-const	add/subtr-var
Mastered (%)	74.4	6.7	73.8	6.7	26.8	79.3	79.3	68.9	6.7
Avg Opps.	15.9	1.1	16.2	0.9	4.1	15.4	18.3	16.3	1.2
Avg Overpr.	6.4	0.5	6.7	0.5	1.5	7.8	9.8	6.2	0.5

Varied Skill Mastery and Practice Engagement The "Mastered" column indicates the proportion of students who achieved mastery for each skill, while the "Avg Opps" column denotes the average number of completed practice opportunities per skill. Average practice engagement varies significantly, with some skills having a higher average number of completed practice opportunities than others, such as *division-simple*.

Severe Overpractice The column "Avg Overpr." delineates the average number of practice opportunities associated with overpractice for each skill. A considerable number of students experience overpractice across all skills. While more students overpractise the skills, some students achieve mastery of them.

3.3 Fast-Forwarding

Fast-Forwarding is a novel technique introduced in Fig. 1 that terminates practice problems early and forwards students to the next problem during the active problem-solving process ("inner loop" [36]). Fast-Forwarding is a flexible technique that can augment existing problem selection algorithms. Its only requirement is access to a pool of multi-step practice problems–a precondition for any step-level adaptation.

Fast-Forwarding optimizes practice efficiency by focusing on unmastered problem-solving steps, allowing students to achieve proficiency more quickly without redundant repetition of already mastered skills (i.e., overpractice). However, a key consideration was to exclude unnecessary skills from the solution pathway **if and only if all remaining skills had already been mastered**. This decision draws on the literature about the composition effect [14], explained below.

Specifically, we did not adopt a model that would fast-forward students through problem-solving steps for *any* skill they had already mastered in the middle of the problem. Advancing students this way might limit the benefits of practicing combinations of skills in multi-step problem solving; in that case, unmastered in combination with mastered skills. This effect, known as the composition effect, suggests that a skill can be more challenging when practiced alongside a second skill and may require integrated practice [14]. Thus, Fast-Forwarding ensures students continue to practice skill combinations until all involved skills are mastered, preventing premature advancement that could leave important compositions unpracticed or systematically underpracticed.

3.4 Simulation Methodology

We simulate student performance (AFM; [7]) and trace student knowledge (BKT; [9]) through two distinct models. AFM and BKT work together to simulate student responses and different problem-selection regimes. Our simulations focus on problem-solving processes at the step-level, in line with AFM [7] and prior analyses of our study data [6].

Knowledge Tracing: Mimicking the knowledge tracing within the Lynnette tutoring system, we employ BKT [9] to analyze student responses and consider mastery to be achieved at a threshold of 95%, which is the threshold in our study data's Lynnette tutoring system [26]. BKT models student learning as a hidden Markov process where each skill is represented by a binary latent state (learned or unlearned). The model updates the probability of mastery based on observed correct and incorrect responses, using four key parameters: prior knowledge (p_{init}), the initial probability that a student knows the skill before any practice, learning rate (p_{learn}), the probability of transitioning from the unlearned to the learned state on each opportunity, guess rate (p_{guess}), the probability of answering correctly by guessing when the skill is unlearned, and slip rate (p_{slip}), the probability of answering incorrectly due to a mistake when the skill is learned. Given an observed response, BKT applies Bayes' rule to update the posterior probability of mastery. If a student answers correctly, the model increases the mastery probability, accounting for possible guessing. If they answer incorrectly, it decreases the probability, adjusting for slips. Mastery is typically declared when the posterior probability exceeds a predefined threshold (here, 95%). For our simulations, we used standard TutorShop BKT parameters for all skills ($p_{init} = 0.25$, $p_{learn} = 0.2$, $p_{guess} = 0.2$, $p_{slip} = 0.1$), ensuring consistency with typical cognitive tutor implementations [1]. For this study, we simulate $10,000$ simulated students that exhibit a normal distribution in initial AFM proficiency levels.

Student Simulation: We simulate student learning data using AFM [7], a logistic regression model of knowledge acquisition (represented as the probability of getting a specific problem-solving step requiring a specific skill correct) as a function of learners' initial proficiency levels, skill-specific difficulty, and skill-specific learning rates. To promote the validity of our methodology, we fit the standard AFM model [7] to our open source data set (see Section 3.1) to obtain estimates of student acquisition rates across different skills, which then serve the simulation of performance in our experiments. Skill-level learning rates used for simulation are encoded in interactions between the learning opportunity count and the skill exercised at a given problem-solving step. This AFM learning rate should not be confused with the BKT parameters for knowledge tracing serving problem selection, where we use standard default settings from Lynnette, which assume constant skill learning rates across all skills (see above), to mimic student experiences in the original study.

We define prototypical solution paths for each practice problem based on the most frequent skill-application path for each problem in our data. At each step in the simulation, we sample a student response based on the correctness probability estimated by the AFM model. We update the AFM simulation and BKT models before moving to the next step or practice problem. In line with Lynnette [26], BKT mastery estimates provide a basis for all problem-selection algorithms–including the proposed Fast-Forwarding method. To address RQ2 we implement a mechanism that replenishes the problem pool after it has been exhausted, meaning that the problem pool is reset to its initial state. This mechanism ensures that the simulation only terminates once all skills are mastered.

Problem-selection Algorithms Fast-Forwarding offers a flexible method for enhancing the efficiency of existing problem selection algorithms. To verify its capabilities, this study integrates Fast-Forwarding into diverse problem selectors:

Random: Selects practice problem from the pool at random without replacement.

Deterministic: Selects practice problems following a domain expert-defined order taken directly from the standard Lynnette set suite [26] designed to surface increasingly difficult and complex problems over time.

Mastery Easy [2]: Employs student mastery estimates to select problems by averaging the difficulty of all available opportunities and serving the problem with the *lowest* difficulty based on mastery estimates while ensuring that the problem has at least one unmastered skill.

Mastery Hard [2]: Employs student mastery estimates to select problems by averaging the difficulty of all available opportunities and serving the problem with the highest difficulty, usually featuring the most unmastered skills. This selector is the standard problem selector in Lynnette [26] and is evaluated in RQ1.

Focused Practice [15]: More recent selection algorithm that considers problem difficulty while preferentially sampling problems with few skills via multinomial sampling.

4 Results

4.1 Learning Efficiency Benefits (RQ1)

For the legacy mastery-based problem selector, Fast-Forwarding reduced overpractice by 35.7% for the average student, as measured in learning opportunities to apply mastered skills. Assuming a median number of steps, underpractice was not changed by Fast-Forwarding (M = 0.44 for Fast-Forwarding vs. M = 0.43 without). Hence, Fast-Forwarding mitigates overpractice while not introducing more underpractice, meaning that overall mastery levels do not suffer.

Fig. 2 shows the sum of overpractice across various skills averaged across students. Overall, without Fast-Forwarding, overpractice is substantially higher, which can be attributed to continuing practice without early termination of fully mastered pathways. Notably, Fast-Forwarding removes overpractice entirely from some skills. For example, in our sample, Fast-Forwarding removed overpractice from the skills “subtract variable,”, “combining like variables,” and “cancel variable.” These are skills that typically appear at the end of practice problems. The

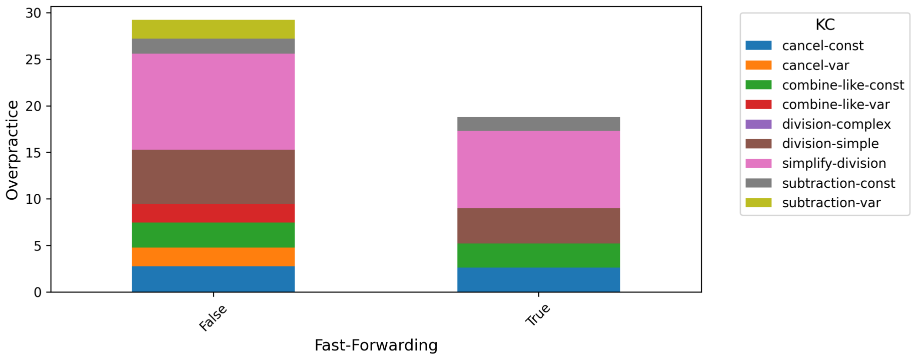

Fig. 2: Stacked bar plot showing the reduction in overpractice across skills when Fast-Forwarding is applied.

same is true for skills involving handling constants. However, these skills may also be required to set up more advanced skills through transformation, such as division. These "set up" skills may not always be eliminated by Fast-Forwarding. For example, 3x - 2 = 4 requires first subtracting as division by 3 upfront is not permitted in Lynnette, which only accepts whole numbers [26]).

4.2 Effects Across Problem Selectors (RQ2)

One strength of Fast-Forwarding is that it does not depend on the specific problem selection algorithm used. Fig. 3 shows the impact of applying Fast-Forwarding on overpractice across various problem selection regimes. This simulation aimed to determine how much overpractice students would encounter to reach mastery, assuming unlimited time. Notably, Fast-Forwarding consistently reduced overpractice across all schedulers, ranging from 0.1% to 35.7%. The largest reduction for the legacy mastery hard scheduler aligns with RQ1 findings, but with the difference that no learning opportunity limit was imposed here, reducing overpractice somewhat more.

As error bars in Fig. 3 indicate, between-student variation in overpractice differed across schedulers. As expected, random scheduling led to the largest variation, and deterministic scheduling led to the second-lowest variation. Despite this variation, Fast-Forwarding still led to a notable reduction in overpractice of 0.67 *SD* for this scheduler. Surprisingly, mastery easy led to the lowest variation in overpractice and only marginal overpractice reductions from Fast-Forwarding. Notably, the modest reduction in the focused practice scheduler was concurrent with the second-largest variation in overpractice across students (7.9%, corresponding to a reduction of 0.41 *SD*). We suspect this variation stems from increased randomness in the scheduler, both choosing skills *and* problems randomly. Accordingly, the mastery hard scheduler was more desirable from the reported largest reduction in overpractice (35.7%; RQ1) *and* lower between-student variation, corresponding to a substantial effect size of 3.4 *SD*.

Finally, given the content unit being designed to include sufficient problems to allow all students to master content, the deterministic scheduler, requiring all problems, was associated with substantial overpractice, over ten times larger than mastery-based schedulers. Overall, assuming students exert continued practice effort, mastery is obtained with the least overpractice for schedulers that Fast-Forwarding, especially those who preferentially serve difficult problems.

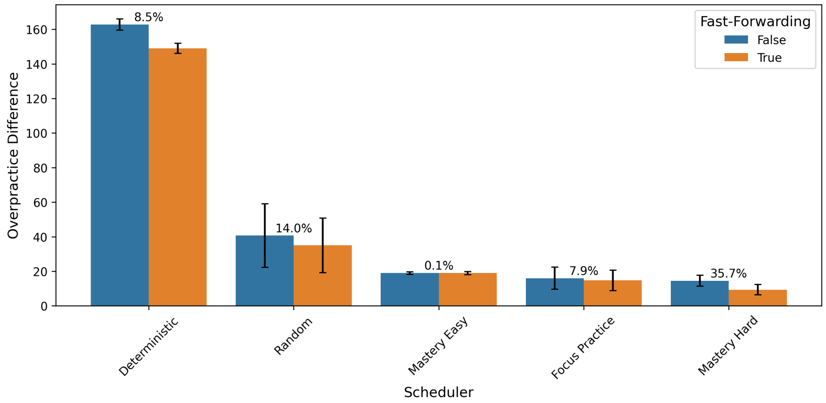

Fig. 3: Overpractice difference across schedulers for Fast-Forwarding, including % decreases when Fast-Forwarding is applied. Error bars are 2 SD across students.

5 Discussion

Mastery learning makes learners improve efficiently [32]. Problem selection is key in improving learning analytics applications that support mastery learning, most notably tutoring systems [17]. The objective of the present study was to evaluate a novel method that promises to improve the efficiency of mastery learning as measured in overpractice [7] while preserving multi-step problem solving and its benefits to learning [14]. Through simulation based on parameters informed by learner data, we applied our method, Fast-Forwarding student problem solving, to avoid overpractice across contemporary and legacy methods for problem selection in a tutoring system.

5.1 Fast-Forwarding Under Legacy Practice Conditions (RQ1)

RQ1 focused on learning efficiency gains in a legacy mastery problem selector, prioritizing hard problems given limited practice time. Informing practice time by average step completions of students in an open-source data set and using learning parameters from the same data, our analysis aimed at closely estimating how much practice students could save by our Fast-Forwarding method. Under limited practice time, our findings indicate that students could save up to about a third of overpractice in a linear equation-solving tutor [26] by Fast-Forwarding. In particular, this overpractice gain did not come with an underpractice loss, which means the overall efficiency of practice improved.

Although our findings are encouraging, they pose two central questions for future research. First, while Fast-Forwarding is generally applicable to any mastery-based problem selector and the content unit, equation solving is a task domain that naturally affords a high level of overpractice: More complex problems require more steps than simple problems, and intermediate problem states are similar to simpler problems, whose problem steps are repeated. Some of these "foundational steps" at the end of equations were eliminated from overpractice through Fast-Forwarding, for example, combining variables. However, other task domains do not have such hierarchical problem structures, such that Fast-Forwarding could potentially be less effective for such domains, which merits future research. On the other hand, Fast-Forwarding could not eliminate *all* underpractice in equation solving, potentially because some steps are required to set up more complex, unmastered operations, such as division, where students first need to subtract constants (a more foundational skill) to divide adequately (e.g., in 3x - 2 = 4, where division by 3 upfront is not permitted in Lynnette, which only accepts whole numbers [26]). While it could be possible to fast-forward students to the state 3x = 6 to just practice division, dealing with constants, in this case, may pose an important learning opportunity, assuming a composition effect whereby the same skill is more or less difficult depending on co-occurring skills [14]. Although the current version of Fast-Forwarding preserves such learning opportunities (by only fast-forwarding if *all* remaining possible operations are mastered), future research may investigate to what extent learners could benefit from intermediate forwarding.

5.2 Fast-Forwarding is Effective for High-Difficulty Selectors (RQ2)

Assuming unlimited practice time, RQ2 focused on the effectiveness of Fast-Forwarding flexibly applied to different contemporary and legacy problem selection regimes. We find that, in a content pool designed for enough problems to allow all students to master all skills, mastery learning as such already substantially reduces overpractice compared to deterministic schedulers that require learners to fully complete a unit before moving on. However, for all mastery-based algorithms, including recent ones that try to create more opportunities for focused practice and less overpractice [15] Fast-Forwarding further reduced overpractice. However, it especially did so for selectors that prioritized difficult problems while minimizing differences between students in overpractice, which may be desirable for creating practice regimes that are equally effective for all students [20]. One interpretation of this finding is that mastery algorithms that prefer harder problems tend to serve problems with many skills (i.e., the legacy algorithm used in this study tends to maximize the number of unmastered skills in a problem). In contrast, mastery easy tends to serve problems with fewer skills (with at least one unmastered skill) [19]. It is hence expected that longer problems allow for more opportunities for the Fast-Forwarding mechanism to fast-forward the student, reducing overpractice to a larger extent. Second, selectors that preferred harder problems in our selection also led to the least problem selection overall. This is perhaps because, unlike mastery easy, such selectors more heavily punish problems that include mastered knowledge, as mastery easy only requires at least one skill in a problem to be unmastered.

5.3 Considerations Toward Applying Fast-Forwarding in Practice

While our findings suggest that Fast-Forwarding alongside selecting hard problems reduces overpractice most, it is not self-evident that such problem selection regimes will be most effective for learning in practice. Although mastery learning is effective [32], it can be difficult for students to maintain the effort required to engage in practice. Studies published in the 1970s documented that student participation in computer-assisted instruction decreased with difficulties below 80% expected accuracy and was high when problems were easy [28]. While motivational theory suggests that a moderate level of difficulty is the most desirable level for motivation, more recent research also suggests that novelty and variation in learning tasks can compensate for some of the adverse effects of high difficulty on motivation [25]. Similar compensatory effects have been observed for topics of interest [12]. Therefore, given that engagement in practice is crucial for the learning benefits of tutoring systems [20]; and ultimately, more important than practice efficiency (i.e., much practice with a little overpractice will usually result in more learning than little or no practice with no overpractice), future research is required to determine to what extent Fast-Forwarding can benefit students and their learning most. Although our findings suggest that Fast-Forwarding is especially helpful in reducing overpractice in problem selectors prioritizing hard problems, increasing difficulty (i.e., removing mastered skills) even further through Fast-Forwarding may not always be desirable for student motivation and engagement. Increasing the novelty of tasks to potentially and partially offset this effect [25] could be realized through authoring additional focused tasks [23] or potentially only executing Fast-Forwarding sometimes and at random, which would also lower overall practice difficulty by occasionally including mastered skills. Independently, current mastery thresholds (i.e., 95%) might be better if they were higher to account for forgetting, an interesting question outside the scope of our paper. These design considerations, including how to best introduce and realize Fast-Forwarding in the tutoring system interface, are beyond the scope of the present study and subject to future research.

5.4 Limitations and Future work

First, our findings are limited to a single domain (i.e., equation solving). Given the portability of Fast-Forwarding to any problem selector and problem domain with empirically derived student solution pathways, our findings merit future research into more domains of tutoring, which our open-source code enables.[1]

[1] https://anonymous.4open.science/r/cut-early

Second, our Fast-Forwarding implementation was limited because Bayesian Knowledge Tracing (BKT) parameters were not fitted for individual students, which may have impacted the precision of skill mastery predictions. Although our BKT values used aligned with actual Lynnette settings [26], future research should consider individual fitting BKT parameters to learner data for more precise problem selection and simulation.

Third, our findings are limited to a single skill model, while potential difficulty factors in complex skill models may better describe the learner data [26]. A risk of using Fast-Forwarding when applied to coarse-grain skill models is taking away practice from difficult, hidden skills. Although skill model refinement is a key challenge in data-driven tutoring system improvement [15], future research may study what effects Fast-Forwarding has on coarse-grain skill models.

6 Conclusion

This study introduced the generalizable Fast-Forwarding method to enhance mastery learning efficiency by reducing overpractice while preserving the benefits of multi-step problem solving. Our simulations showed that Fast-Forwarding reduces overpractice by up to one-third, particularly in selectors prioritizing harder problems. This improvement makes it a promising tool for optimizing tutoring systems, a key application area of technology-enhanced learning. These findings suggest that balancing efficiency with task difficulty, which may reduce student engagement if too high, should be a key consideration in future research to improve learning in practice.

References

1. Aleven, V., Borchers, C., Huang, Y., Nagashima, T., McLaren, B., Carvalho, P., Popescu, O., Sewall, J., Koedinger, K.: An integrated platform for studying learning with intelligent tutoring systems: Ctat+ tutorshop. arXiv preprint arXiv:2502.10395 (2025)
2. Aleven, V., McLaren, B., Sewall, J., Koedinger, K.R.: Example-tracing tutors: A new paradigm for intelligent tutoring systems (2009)
3. Aleven, V., Mclaren, B.M., Sewall, J., Koedinger, K.R.: A new paradigm for intelligent tutoring systems: Example-tracing tutors. International Journal of Artificial Intelligence in Education **19**(2), 105–154 (2009)
4. Aleven, V., McLaughlin, E.A., Glenn, R.A., Koedinger, K.R.: Instruction based on adaptive learning technologies. Handbook of research on learning and instruction **2**, 522–560 (2016)
5. Bloom, B.S.: A response to slavin's mastery learning reconsidered. Review of Educational Research **57**(4), 507–508 (1987). https://doi.org/10.3102/00346543057004507
6. Borchers, C., Yang, K., Lin, J., Rummel, N., Koedinger, K.R., Aleven, V.: Combining dialog acts and skill modeling: What chat interactions enhance learning rates during ai-supported peer tutoring? In: Proceedings of the 17th International Conference on Educational Data Mining (2024)

7. Cen, H., Koedinger, K., Junker, B.: Learning factors analysis–a general method for cognitive model evaluation and improvement. In: International conference on intelligent tutoring systems. pp. 164–175. Springer (2006)
8. Cen, H., Koedinger, K.R., Junker, B.: Is over practice necessary?-improving learning efficiency with the cognitive tutor through educational data mining. Frontiers in artificial intelligence and applications **158**, 511 (2007)
9. Corbett, A.T., Anderson, J.R.: Knowledge tracing: Modeling the acquisition of procedural knowledge. User modeling and user-adapted interaction **4**, 253–278 (1994)
10. Dawson, S., Joksimovic, S., Poquet, O., Siemens, G.: Increasing the impact of learning analytics. In: Proceedings of the 9th international conference on learning analytics & knowledge. pp. 446–455 (2019)
11. Fancsali, S., Nixon, T., Ritter, S.: Optimal and worst-case performance of mastery learning assessment with bayesian knowledge tracing. In: Educational Data Mining 2013 (2013)
12. Fulmer, S.M., Frijters, J.C.: Motivation during an excessively challenging reading task: The buffering role of relative topic interest. The Journal of Experimental Education **79**(2), 185–208 (2011)
13. Gervet, T., Koedinger, K., Schneider, J., Mitchell, T.: When is deep learning the best approach to knowledge tracing? Journal of Educational Data Mining **12**(3), 31–54 (2020)
14. Heffernan, N.T., Koedinger, K.R.: A developmental model for algebra symbolization: The results of a difficulty factors assessment. In: Proceedings of the twentieth annual conference of the cognitive science society. pp. 484–489. Routledge (2022)
15. Huang, Y.: Learner modeling for integration skills in programming. Ph.D. thesis, University of Pittsburgh (2018)
16. Huang, Y., Aleven, V., McLaughlin, E., Koedinger, K.: A general multi-method approach to design-loop adaptivity in intelligent tutoring systems. In: Artificial Intelligence in Education: 21st International Conference, AIED 2020, Ifrane, Morocco, July 6–10, 2020, Proceedings, Part II 21. pp. 124–129. Springer (2020)
17. Huang, Y., Lobczowski, N.G., Richey, J.E., McLaughlin, E.A., Asher, M.W., Harackiewicz, J.M., Aleven, V., Koedinger, K.R.: A general multi-method approach to data-driven redesign of tutoring systems. In: LAK21: 11th International Learning Analytics and Knowledge Conference. pp. 161–172 (2021)
18. Käser, T., Alexandron, G.: Simulated learners in educational technology: A systematic literature review and a turing-like test. International Journal of Artificial Intelligence in Education **34**(2), 545–585 (Jun 2024). https://doi.org/10.1007/s40593-023-00337-2
19. Koedinger, K., Pavlik Jr, P.I., Stamper, J., Nixon, T., Ritter, S.: Avoiding problem selection thrashing with conjunctive knowledge tracing. In: Educational data mining 2011 (2010)
20. Koedinger, K.R., Carvalho, P.F., Liu, R., McLaughlin, E.A.: An astonishing regularity in student learning rate. Proceedings of the National Academy of Sciences **120**(13), e2221311120 (2023)
21. Koedinger, K.R., Corbett, A.: Cognitive tutors. Smart machines in education pp. 145–167 (2001)
22. Koedinger, K.R., D'Mello, S., McLaughlin, E.A., Pardos, Z.A., Rosé, C.P.: Data mining and education. Wiley Interdisciplinary Reviews: Cognitive Science **6**(4), 333–353 (2015)
23. Koedinger, K.R., McLaughlin, E.A.: Closing the loop with quantitative cognitive task analysis. International Educational Data Mining Society (2016)

24. Li, N., Stampfer, E., Cohen, W., Koedinger, K.: General and efficient cognitive model discovery using a simulated student. In: Proceedings of the Annual Meeting of the Cognitive Science Society. vol. 35 (2013)
25. Lomas, J.D.: Optimizing Motivation and Learning with Large-Scale Game Design Experiments. Phd thesis, Human-Computer Interaction Institute, School of Computer Science, Carnegie Mellon University (2014)
26. Long, Y., Holstein, K., Aleven, V.: What exactly do students learn when they practice equation solving? refining knowledge components with the additive factors model. In: Proceedings of the 8th International Conference on Learning Analytics and Knowledge. pp. 399–408 (2018)
27. Matsuda, N., Cohen, W.W., Sewall, J., Lacerda, G., Koedinger, K.R.: Predicting students' performance with simstudent: Learning cognitive skills from observation. Frontiers in Artificial Intelligence and Applications **158**, 467 (2007)
28. Miller, R., Hess, R.D.: The effects upon students' motivation of fit between student ability and level of difficulty of cai programs. Tech. rep., Stanford University, School of Education, Stanford, Calif. (1972)
29. Pardos, Z.A., Yudelson, M.V.: Towards moment of learning accuracy. In: AIED 2013 Workshops Proceedings Volume. vol. 4, p. 3. Citeseer (2013)
30. Pavlik Jr, P.I., Cen, H., Koedinger, K.R.: Performance factors analysis–a new alternative to knowledge tracing. Online Submission (2009)
31. Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L.J., Sohl-Dickstein, J.: Deep knowledge tracing. In: Advances in Neural Information Processing Systems. pp. 505–513 (2015)
32. Ritter, S., Yudelson, M., Fancsali, S.E., Berman, S.R.: How mastery learning works at scale. In: Proceedings of the Third (2016) ACM Conference on Learning@ Scale. pp. 71–79 (2016)
33. Ritter, S., Anderson, J.R.: Calculation and strategy in the equation solving tutor. In: Proceedings of the Annual Meeting of the Cognitive Science Society. vol. 17 (1995)
34. Schmucker, R., Wang, J., Hu, S., Mitchell, T.: Assessing the knowledge state of online students - new data, new approaches, improved accuracy. Journal of Educational Data Mining **14**(1), 1–45 (2022)
35. Shin, D., Shim, Y., Yu, H., Lee, S., Kim, B., Choi, Y.: Saint+: Integrating temporal features for EdNet correctness prediction. In: Proceedings of the 11th International Learning Analytics and Knowledge Conference. pp. 490–496. ACM, New York, NY, USA (2021)
36. VanLehn, K.: The behavior of tutoring systems. International journal of artificial intelligence in education **16**(3), 227–265 (2006)
37. Walker, E., Rummel, N., Koedinger, K.R.: Adaptive intelligent support to improve peer tutoring in algebra. International Journal of Artificial Intelligence in Education **24**, 33–61 (2014)
38. Weitekamp, D., Ye, Z., Rachatasumrit, N., Harpstead, E., Koedinger, K.: Investigating differential error types between human and simulated learners. In: Artificial Intelligence in Education: 21st International Conference, AIED 2020, Ifrane, Morocco, July 6–10, 2020, Proceedings, Part I 21. pp. 586–597. Springer (2020)
39. Zhang, L., Lin, J., Borchers, C., Cao, M., Hu, X.: 3dg: a framework for using generative ai for handling sparse learner performance data from intelligent tutoring systems. arXiv preprint arXiv:2402.01746 (2024)