INVESTIGATING THE ROBUSTNESS OF KNOWLEDGE TRACING MODELS IN THE PRESENCE OF SHIFTING STUDENT DISTRIBUTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Knowledge Tracing (KT) has been an established problem in the educational data mining field for decades, and it is commonly assumed that the underlying learning process being modeled remains static. Given the ever-changing landscape of online learning platforms (OLPs), we investigate how concept drift and changing student populations can impact student behavior within an OLP through testing model performance both within a single academic year and across multiple academic years. Four well-studied KT models were applied to five academic years of data to assess how susceptible KT models are to concept drift. Through our analysis, we find that all four families of KT models can exhibit degraded performance. Bayesian Knowledge Tracing (BKT) remains the most stable KT model when applied to newer data, while more complex, attention-based models lose predictive power significantly faster when features provided are diverse enough to allow overfitting. Finally, we apply well-studied measures of dataset similarity to assess which kinds of drifts may account for the performance degradation we document. To foster more longitudinal evaluations of KT models, the data used to conduct our analysis is available at `https://osf.io/hvfn9/?view_only=b936c63dfdae4b0b987a2f0d4038f72a`.

1 INTRODUCTION

To extract meaning from large amounts of data, one may generally assume that the underlying processes which generate said data are static, or at least relatively stable over long periods of time. One of the core goals of educational data mining (EDM) is, of course, using data to model aspects of students' learning processes. These models can then be used to predict, explain, or challenge our current understanding of how students learn in online learning platforms (OLPs). Even in cases where the goal is to better suit individual learners' needs, the methodology remains the same: create a model of student behavior and learning by analyzing previously collected data, identify components of that model which may fail to account for individual differences, and update the model to account for the ways in which particular learners differ. These powerful methodologies have allowed researchers to detect the affective state of students Botelho et al. (2017); Calvo & D'Mello (2010), model the procedural acquisition of knowledge Corbett & Anderson (1994), predict student success Kovačić (2010), and detect problematic or unhelpful student behavior Baker et al. (2010); Vanacore et al. (2024).

As this is the case, the goalposts have moved beyond within-context, task-specific learning to questions of *generalizability*. Rather than training models and detectors that are tailored to their hyperspecific learning environments, can we instead train systems whose insights are able to be transferred across contexts, perhaps by training a model in one platform and assessing its efficacy in another? After all, base learner characteristics (propensity to game, for example) almost certainly have some intrinsic components. Even if a particular system is more or less likely to interact with individual student tendencies, cross-context modeling would show that we can truly model learning process and not just measure statistical noise.

A pressing question, then, is how well do existing frameworks handle within-system variations between learner populations? We *assume* that data collected from different learners within the same

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

system represents the same underlying learning process. We *assume* that the ways in which students learn that are measurable by scientists and practitioners remain consistent over time. Given the maturation of EDM as a field and the availability of learner data spanning generations of learners, perhaps it is now possible to verify that our assumptions are correct, or at least to identify the circumstances where they are safe assumptions to make.

The educational best-practices of 30 years ago are obviously not the educational best-practices of the modern day. Educational policy has shifted towards meticulous measurement of student progress Hallinger & Heck (2011), identifying failing schools Nicolaidou & Ainscow (2005), and standardizing subject curricula to better facilitate rigorous measurement Popkewitz (2004). Simultaneously, OLPs have risen in popularity, automating student practice and proliferating student engagement data Ritter et al. (2007); Heffernan & Heffernan (2014). These educational platforms have also matured since their creation, and every pedagogical and cosmetic change to these platforms could impact the way students interact with these platforms. Even ignoring educational policy changes, students are individuals in a large and changing world, and world events which change how humans relate to one another impact students as much as anyone else. In a particularly extreme example, an entire generation of students experienced learning losses due to the COVID-19 pandemic Donnelly & Patrinos (2022). In a changing world, how can we be sure our modeling techniques are still valid?

More specifically, we wish to understand how Knowledge Tracing (KT) models are impacted by changes in student populations over time. KT is a foundational problem of the educational data mining field, and as such we intend to investigate how a number of different modeling techniques behave when applied outside of their temporal context, aiming to answer the following research questions:

RQ1. How robust are KT models to changing student populations?

RQ2. Does the complexity of a KT model impact its longevity?

RQ3. Do some years produce more generalizable models than others?

RQ4. How similar are student $\times$ problem interactions in XXX OLP in different academic years?

To achieve this, we borrow from Data Mining literature the concept of dataset shift, and discuss its applicability to online learning platforms. We then propose a methodology for evaluating KT models both within their temporal context and across student populations, taking steps to ensure that our datasets contain similar exercise banks and Knowledge Concepts. We then apply this methodology to four well-studied KT models and examine how each model performs outside of its temporal context. Following this, we investigate which variants of dataset shift may be present in our data, and discuss relevant implications for our findings. We conclude with a more thorough discussion of this study's limitations, and propose future work which addresses these known shortcomings.

In addition to the listed analyses above, there is a notable lack of knowledge tracing datasets that span multiple years. Alongside this paper we will be releasing a publicly available dataset containing student interaction data spanning multiple academic years, allowing other researchers to evaluate their own models under controlled conditions of "aging."

2 Background

In this section, we introduce and discuss literature relevant to our investigation of KT model robustness. First, we introduce KT as a specific machine learning task and discuss specific models which will be investigated in this work (Section 2.1). Finally, we discuss frameworks for analyzing the drift of software systems from their original contexts (Section 2.2).

2.1 Knowledge Tracing

Long established in EDM literature, KT is defined as a many-to-many time series binary classification problem attempting to predict the correctness of future student responses based on prior performance. Numerous machine learning architectures have been applied to this task, including Factorization Machines Vie & Kashima (2019) and psychometric models like Item Response Theory Yeung (2019). The work of Shen et al. (2024) provides a comprehensive survey of historical and contemporary methods. In our work, we replicated four well-studied KT models: Bayesian

Knowledge Tracing, Performance Factors Analysis, Deep Knowledge Tracing, and Self-Attentive Knowledge Tracing.

Bayesian Knowledge Tracing Originally proposed by Corbett & Anderson (1994) and deeply connected to mastery learning Bloom (1968), Bayesian Knowledge Tracing (BKT) models the acquisition of knowledge as a latent Markov process. The student's knowledge state is modeled as a latent variable that is noisily observed through performance on exercises in an intelligent tutoring system. Exercises in said tutoring system are tagged with Knowledge Components (KCs) signifying related items, and items tagged with the same KC are treated as having uniform difficulty. Mastery of different KCs is computed independently, meaning that mastery of one KC is completely independent of other KCs. Due to the assumptions made about students' learning processes, each parameter of a BKT model has a direct interpretation that is explainable to teachers and other educational practitioners Williamson & Kizilcec (2021). The latent learning process was originally modeled as unidirectional, modeling students as unable to forget KCs once they have been mastered, but later model variants explore forgetting behavior Qiu et al. (2011), individual estimations of prior knowledge Yudelson et al. (2013), and contextual guess and slip parameters Baker et al. (2008).

Performance Factors Analysis Closely related to Learning Factors Analysis Cen et al. (2006), Performance Factors Analysis (PFA) was first proposed in Pavlik et al. (2009) as a logistic regression based alternative to traditional Knowledge Tracing models. Rather than sequentially modeling a student's learning process and updating mastery estimates based on individual student exercises, PFA instead considers the number of correct exercises (wins) and incorrect exercises (fails) by a student on a given KC, along with a KC-level intercept to account for the relative difficulty of a KC. While BKT models KCs as independent entities, PFA can easily predict future performance while accounting for student knowledge of multiple KCs. More recent evaluations of PFA found it to be competitive with more contemporary KT models in certain scenarios Gervet et al. (2020); Chu & Jr (2023).

Deep Knowledge Tracing Piech et al. (2015) represents the first application of deep learning methods to the problem of KT. Broadly speaking, Deep Knowledge Tracing (DKT) is the application of a recurrent neural network (RNN) to sequences of exercise-response pairs to predict a student's ability to correctly answer future exercises.Though deep learning methods can provide performance gains over classical, "shallower" methods, Khajah et al. (2016) achieve similar performance to the original DKT paper by analyzing certain advantages DKT has over BKT and extending BKT while keeping the underlying model and assumptions the same. In turn, later papers improve on DKT's learning gains by incorporating rich side information into the model Zhang et al. (2017); Wang et al. (2019b).

Self-Attentive Knowledge Tracing With the introduction of attention mechanisms to deep learning in Vaswani et al. (2017), time series classification problems across numerous domains achieved new state-of-the-art methods. KT was no exception to this, with Pandey & Karypis (2019) proposing an attention mechanism for knowledge tracing. Their aptly named Self-Attentive Knowledge Tracing (SAKT) surpassed previous models in performance, while simultaneously showing great promise as a more interpretable deep model, given the ability to visualize attention weights to understand particular exercises which the model weighted as more important or explanatory.

2.2 Distributional Shifts

Broadly speaking, for a given supervised learning problem with training set X and labels Y, we are interested in modeling the joint probability distribution:

$$P(X, Y) = P(Y|X)P(X) = P(X|Y)P(Y)$$

These decomposed probabilities map directly onto components in the supervised learning framework where :

1. $\mathbf{P}(\mathbf{X})$ is the probability distribution of input features.

2. $\mathbf{P}(\mathbf{Y})$ is the probability distribution of outputs, either discrete classes in a classification context or continuous values in a regression context.
3. $\mathbf{P}(\mathbf{Y}|\mathbf{X})$ is the probability distribution of outputs considering input features. This is the probability distribution most directly modeled in a supervised learning context.

A change in any part of these distributions is a form of **dataset drift**. The type of dataset drift which is present in a given data stream depends on the part (or parts) of the decomposed joint distribution that changes Huyen (2022):

1. **Covariate Shift**, a change $P(X)$
2. **Label Shift**, a change in $P(Y)$
3. **Concept Drift**, a change in $P(Y|X)$

Concept drift is perhaps the most difficult of these three shifts to adapt to, since a supervised learning model explicitly attempts to estimate $P(Y|X)$. Prior works have created methods to detect Klinkenberg & Joachims (2000), explain Wang et al. (2019a), and adapt to Madireddy et al. (2019) concept drift.

In this paper, we will utilize multiple different metrics for assessing the difference of two distributions. For the following definitions, let P and Q be two distributions. These definitions will be for the categorical case, thus individual categories P_i and Q_i are the proportions of category i found in each distribution.

Kullback-Leibler Divergence Kullback & Leibler (1951) proposed a measure for how well one distribution approximates another. It is defined in the categorical case as follows:

$$D_{KL}(P, Q) = \sum_{\forall i} P_i \ln(\frac{P_i}{Q_i})$$

A notable drawback of Kullback-Leibler (KL) divergence is its lack of symmetry: $D_{KL}(P, Q) \neq D_{KL}(Q, P)$.

Jensen-Shannon Divergence The Jensen-Shannon (JS) Divergence Menéndez et al. (1997) provides a symmetric alternative to KL divergence by constructing a midpoint distribution ($\frac{P+Q}{2}$) by averaging the proportions of each category, and calculating the KL divergence of both P and Q with said midpoint distribution:

$$D_{JS}(P, Q) = \frac{1}{2} D_{KL}(P, \frac{P+Q}{2}) + \frac{1}{2} D_{KL}(Q, \frac{P+Q}{2})$$

Population Stability Index Alternatively, the Population Stability Index Yurdakul & Naranjo (2020) is another symmetric alternative formed by summing both orderings of the KL Divergence of P and Q:

$$PSI(P, Q) = D_{KL}(P, Q) + D_{KL}(Q, P) = \sum_{\forall i} (P_i - Q_i) \ln(\frac{P_i}{Q_i})$$

These metrics were chosen due to their prior use by Deho et al. (2024), one of the only applications of dataset drift as a framework to educational contexts. Deho et al. (2024) employ both PSI and JS Divergence in their investigation. The most straightforward way of computing both of these metrics involves the computation of KL divergence, given their common origin, thus we see no reason to avoid reporting this measure of divergence as well. PSI and KL divergence show similar levels of sensitivity, while JS divergence tends to be moderately more sensitive.

3 METHODS

3.1 DATA COLLECTION & PREPARATION

Data for this study originates from the XXX OLP's developer database, spanning the five academic years between 2019 - 2020 and 2023 - 2024. Data not suitable for conducting Knowledge Tracing

was filtered out of a broad query, consisting of all data collected in the months of June, July, and August, as well as problem logs for non-computer-gradable questions, and all problem logs from problem sets assigned fewer than 100 times total during the five academic years of interest. As in Lee et al. (2023) and Lee & Heffernan (2025), we find that summer student populations often differ greatly to the population of students using an OLP during the school year, while non-computer-gradable problems are incompatible with standard KT models, and removing low-use problem sets from the data lowers the likelihood of models differing solely due to out-of-vocabulary KCs and exercises. It is notable that these low-use problem sets account for 36.7% of the interactions (see figure 8 for more information about the data loss at different filtering thresholds). Information about the size of the data gathered from each academic year can be found in table 1. The relative size of each year's data is worthy of note. Different years have great differences in the number of available problem logs, with the largest year having over twenty-one times the amount of total problem logs. Different years also vary in how much usage each individual student contributes to the data. For example, students in the 2020-2021 school year completed 12.5 assignments on average, compared to an average of 7.2 assignments just one year prior. Since the amount of available training data has a large impact on model fitness, this disparity in dataset sizes presents an issue.

AY	Total Rows	Assignment Logs	Unique Students	Unique KCs	% Correct
2019–2020	17,962,663	1,645,060	228,207	408	0.728
2020–2021	69,760,692	5,478,914	437,500	411	0.697
2021–2022	11,421,033	1,309,773	122,397	412	0.686
2022–2023	5,382,200	754,299	71,284	407	0.668
2023–2024	3,254,928	519,700	50,896	408	0.660

Table 1: Dataset sizes after filtering out ineligible problem logs.

To mitigate the impact of our dataset sizes, rather than using all available data for each year, we draw random samples from each available academic year. Randomly sampling *user/exercise interactions* would isolate those rows from their surrounding context, while sampling *per user* reintroduces concerns over differences in training set sizes, as the total number of exercises completed per user varies widely. Instead, similar to Lee et al. (2023) and Lee & Heffernan (2025), we randomly sample assignment logs. Every assignment log is an instance of a student/problem set interaction. This allows us to draw samples of consistent size, since problem set length is more consistent, while collecting coherent sequences of student/exercise interactions in their full context. Our final dataset consists of ten such samples[1] per academic year, with samples containing 50,000 assignment logs each[2]. While every assignment log is a unique student/problem set interaction, one student may have multiple assignment logs. A simplified example of this sampling strategy for a single academic year can be found in figure 1.

3.2 Study Design

In order to effectively investigate the susceptibility of KT models to concept drift (RQ1), we need to establish baseline performance for each model on each target year and evaluate models in a cross-year context. To measure within-year performance, we conducted a ten-fold cross-validation, training one model per sample and evaluating it on the other nine samples[3]. To investigate model performance across years, for each sample of a target year, we trained a model on the full sample and evaluated the fit model on one sample from all other years, before and after. The choice to evaluate models on prior years is unorthodox, but necessary in order to better measure the overall effect of training years on model performance (RQ3). To explicitly investigate the effect of overparameterization on model performance over time, two different versions of SAKT were evaluated: one using KCs as model input and one using exercises directly (RQ2).

Additionally, to investigate the presence of covariate and label shift present within each sample (RQ4), Kullback-Leibler divergence, Jensen-Shannon divergence, and population stability index are

[1] In this paper, "sample" refers specifically to one of these random samples of 50,000 student/assignment interactions, *not* individual examples of model inputs & outputs.

[2] These samples are available here

[3] Rather than doing a classic 90/10 train-test split for cross-validation, we opt to train on one sample and evaluate on the other nine to make sure all models are trained on roughly the same amount of data

computed pairwise for every sample, for both skills (to assess covariate shift) and for discrete scores (to assess label shift).

4 RESULTS

Based on the above methodology, each architecture was evaluated on multiple metrics: binary cross entropy loss (also known as logistic loss or log loss), area under the ROC curve (ROC-AUC, or simply AUC), and F1 score (the harmonic mean of two other measures, precision and recall). As the loss function used by all implemented models, log loss provides a measure of how "well trained" a particular model is, in a mathematical sense. AUC and F1 score measure different aspects of a model's ability to discriminate between positive and negative examples. All five models have significant correlations between model fit metrics and the number of years between training & evaluation data (see table 4 in the appendix for estimated spearman correlation coefficients) indicating that increased age since training worsens model performance. Average values for these metrics are plotted against the number of years between the training and evaluation year, grouped both by model and by training year (figures 3a–5b).

To evaluate differences in the rates of model performance loss, we performed the following regression on all three reported metrics:

$$\text{metric} = \alpha + \alpha_m + \alpha_t + \alpha_e + (\beta + \beta_m + \beta_t)y$$

The core covariate of this regression is y, the number of years between training and evaluation. With m model architecture, t training year, and e evaluation year, we fit global intercept α and global coefficient β. The effects of model type, training year, and evaluation year are addressed by estimating intercept terms for each, along with interaction terms for model and train year with years between. Intuitively speaking, we use an interaction term with years between to evaluate whether or not model type and training year moderate performance degradation, and incorporate interactive intercepts to control for individual differences between model types, training years, and evaluation years. To address heteroskedasticity concerns, standard errors and resulting p-values are calculated using heteroskedasticity consistent standard errors suggested by Davidson & Mackinnon (1993). These regression results may be found in appendix tables 5–7.

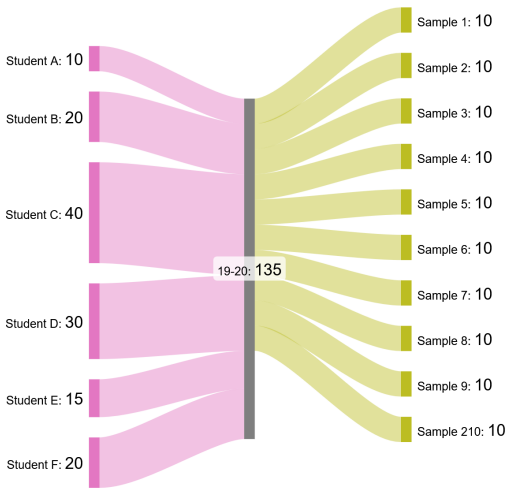


Figure 1: Simplified example of our sampling strategy. Each student contributes varying numbers of assignment logs to the total year's pool, samples are created by drawing from these assignment logs

4.1 RQ1: KT MODEL ROBUSTNESS

Broadly, our results suggest that all KT models we tested experience some kind of performance loss linked to the passage of time. Differing measures offer various perspectives into the nature of this decline. AUC provides the clearest relationship between relative training data age and model performance, while F1 score for BKT, PFA, and DKT remained quite stable across all 4 years. SAKT-E exhibited significance performance loss across all three metrics, a fact which is particularly notable when looking at our results for log loss, seeing as all other models showed insignificant associations with model type & performance loss. This is not to say that our models did not experience an increase in log loss over time, but rather that all models (save for SAKT-E) experienced increases in log loss at similar rates, which we attribute more to the particular year they were trained on. Given how AUC is calculated, it is safe to say that even if model prediction quality (as measured by F1 score) is relatively unaffected by a model's age, the model will still see differences in the distributions of its predictions: i.e. the predicted classes will be less extreme (closer to a fair decision

boundary of 0.5). More concretely: the models will be "less confident" in their predictions when used outside of their temporal context. Our results for log loss corroborate this, with larger log loss values indicating greater differences between our models' predicted probabilities and the true class labels. The relative stability of BKT's F1 scores across all temporal contexts is particularly noteworthy, given its outperformance of the other models with stable F1 scores. The regression results of this analysis can be found in 5

4.1.1 LOG LOSS

PFA has consistently higher log loss than other models, while the rankings of all other models change as the relative age of the training/evaluation data increase. SAKT-E has the lowest initial log loss, but shows a clear linear association with relative data age. BKT and SAKT-KC have remarkably similar results, though SAKT-KC shows better performance on all years but one (three years between training and evaluation). With the notable exception of 2020-2021 exhibiting the best initial log loss, models trained on data from later years perform worse, though later years are less impacted by the difference between training and evaluation year. This is possibly due to the backwards nature of evaluating models trained on those years on *earlier* data. It seems broadly more difficult to train effective models on data from later academic years, at least as shown by log loss when evaluated on the same academic year.

4.1.2 AUC

Results for AUC paint a broadly similar picture to log loss results: SAKT-E exhibits the highest initial performance and steepest rate of performance decline. BKT and SAKT-KC decline at very similar rates, with SAKT-KC outperforming (or at least matching) BKT's performance at all model ages. DKT and PFA exhibit uniformly lower scores, with similar rates of decline. Once again, models trained on earlier years have higher performance within-context than later years, with later years having less linear-appearing relationships with temporal distance between training and evaluation.

The impact of training data relative to age is much more pronounced with AUC than log loss. SAKT-KC seems to have a very similar rate of decline, with no significant differences found between its rate of decline and BKT's. SAKT-E declines at a faster rate, with a much steeper slope than the main effect. Once again, later years see worse evaluation results. The regression performed for this analysis can be found in 5.

4.1.3 F1 SCORE

Interestingly, BKT receives consistently high F1 scores across all temporal contexts. PFA and DKT seem similarly unaffected by an increase in relative training data age, but both SAKT variants see sharper declines, with SAKT-E seeing steeper declines than all other models. Similar to other metrics, later evaluation years have lower performances within their own temporal context relative to earlier years, and see more stable performance (some years even increase in F1 score as YB increases) than earlier years.

Surprisingly, the main effect for years between is *positive*, though the significant interaction terms for train years other than 19-20 would bring the slope for YB back into the negative, resulting in an overall negative relationship. DKT and PFA have similar rates of decay to BKT, given their insignificant interaction terms, while SAKT-E has the steepest slope of YB, consistent with results for SAKT-E on other metrics. Regression results for this analysis are located in 7.

4.2 RQ2: COMPLEXITY AS MODERATOR

Upon first glance, model complexity (as measured by model parameters, see table 2) seems linked to the rate of performance loss, given how SAKT-E has consistently steeper coefficients for years between compared to all other models. However, the rates of decline for DKT on all available metrics are relatively small by comparison, despite being the second largest in terms of parameter size. SAKT-KC also experienced rates of decline very similar to BKT on both log loss and AUC, despite having orders of magnitude more parameters. This indicates that model complexity may not be directly linked to lower model robustness as theorized in Levin et al. (2022). While the self-attention models had similar or worse rates of degradation to BKT, other complex models did not

exhibit such a strong relationship. Rather, the main factor which contributes to model degradation appears to be the presence of the attention mechanism in both SAKT-KC and SAKT-E. SAKT-E is the top performing model and SAKT-KC is the next-best model by AUC the when evaluated on the year in which they were trained, but all performance gains SAKT-E had over other models disappeared with only one year between training & evaluation data. SAKT-KC is similarly robust to BKT on log loss and AUC, but both SAKT models are the only ones to experience strictly negative coefficients on F1 score.

Model	Proposed	Logical Unit	Tunable Params
BKT	Corbett & Anderson (1994)	KC	2,012
PFA	Pavlik et al. (2009)	KC	1,208
DKT	Piech et al. (2015)	KC	734,026
SAKT-KC	Pandey & Karypis (2019)	KC	262,982
SAKT-E	Pandey & Karypis (2019)	Exercise	12,400,658

Table 2: Number of trainable parameters of each model.

4.3 RQ3: Training Year Differences

Additionally, there are notable differences in model performance *and* performance loss rates for different training years. Models trained on later years had notably worse performance even within their temporal context compared to models trained on earlier years (see figures 3b, 4b & 5b). Additionally, with the exception of 2020–2021 for AUC, all interaction terms between training year & years between were significant, indicating differing rates of performance loss for different years. This particular finding should be scrutinized, however, given how later models were evaluated on more data from *before* the year they were trained on. Later years were also significantly more difficult to make accurate predictions on than earlier years, as evidenced by all evaluation year intercepts being highly significant. This may indicate that differences between models trained on different years has less to do with differences in the training data used for a model, instead being more strongly linked with differences in evaluation context. As further evidence of this, consider the relative stability of models trained on 2023–2024 applied to 2019–2020, in contrast to the steep performance loss experienced applying a 2019–2020 model to data from 2023–2024

4.4 RQ4: Data Drift Presence

While our method clearly shows some form of performance degradation over time, it remains unclear which particular kinds of dataset shifts are present in our samples. To investigate this, we calculated relevant drift statistics for all relevant train/test pairs, for both KC proportions and overall correctness rates (see figure 6). KC tags are the primary input feature for most of our tested architectures, and thus differing distributions would represent the presence of covariate shift in our data. Similarly, differences in distributions of student scores would show clear evidence of label shift.

Our reported metrics show the clear presence of covariate shift in our data, especially when looking at 2019-2020 and 2020-2021. All years showed no drift within-year (to be expected), but have higher drift statistics, especially in cases where more time passes between the train year and test year. Label shift appears to be absent, as even comparing 2019-2020 to 2023-2024, the distribution of correct & incorrect answers remains very similar.

5 Discussion

In this section, we discuss the implications our findings have for our research questions, ask and answer clarifying questions about the mechanisms underlying the model degradation our models demonstrate, identify weak points in the way we answer our research questions, and propose ways that future investigations into this area may address the shortcomings of this particular work.

5.1 Real World Applications

All our models experienced performance loss as the time between training and evaluation increased, an important takeaway for KT applications in OLPs. Accurate KT algorithms can aid students and teachers through adaptive assignments, personalized support, and more in OLPs. However, a prerequisite to many of these applications is accurate and robust KT models. Acquiring sufficient data year over year to retrain models is not an insignificant task. This begs the question: "Is this necessary, or do models retain their accuracy?" Our results show this necessity. For KT models, an AUC of 0.92, as achieved by SAKT-E, is incredibly competitive, offering ample opportunity for personalizing content. However, as seen in figure 4a, this quickly decreases as the time between training and evaluating the KT algorithms increases. Further, as it's impossible to train a model on the data from a complete school year before that year's end, in most settings models will be trained on the prior school year's data. BKT, SAKT-KC and DKT all achieved promising AUCs one to two years after being trained, suggesting that training these models yearly when school is out or every other year could serve as a suitable retraining schedule.

5.2 Limitations & Future Work

While our findings broadly suggest that KT models are susceptible to concept drift, there are notable limitations in our analysis. As discussed in Shen et al. (2024), our four choices of models represent broader "families" of KT models, and novel KT modeling is an area of active research in EDM. Exploring how extensions to these model families impact robustness would also give more insight into the relationship between model complexity and generalizability. Additionally, while our approach shows both a clear degradation of model performance and the presence of covariate shift across our sampled data, our model-focused approach to measuring concept drift cannot describe *how* the distribution of student responses has changed, nor explain factors which could be causing said change. Clearly, *something* about the interactions students have within an OLP changes through time, and future works could employ a data-centric approach to detecting concept drift alongside evaluating models through time, perhaps adapting recent methods proposed by Cai et al. (2026) to understand how much different forms of dataset drift are contributing to the observed performance degradation. Alternatively, a more thorough investigation of the learned attention weights of the different SAKT models could yield more insight, either identifying what parts of the learning process change over time or verifying that the model is more sensitive to noisy sequences.

6 Conclusion

Expanding upon previous findings, this study indicates that many popular families of Knowledge Tracing models lose predictive power over time. BKT exhibits the slowest rates of decline, while more recently proposed attention-based models decline faster. These findings indicate that the underlying process of student learning (as monitored through student interaction logs) may not be as stable as previously theorized by Lee et al. (2023).

These results have multiple implications for researchers and practitioners in the field. There is a long-standing discussion surrounding the pros and cons of deep learning models in an educational context. The choice between simpler and more explainable models compared to more complex but less explainable models involves considerations of data availability, the desire for explainable models, and ease of model fitting, among other concerns. Based on our findings, we posit that model longevity should be another consideration when deciding how to model student learning.

We encourage other researchers interested in proposing new KT models to explore their models' fitness over longer time frames rather than relying on a single benchmark dataset, and in the interest of pursuing this goal, we are releasing the samples described in section 3.1 with this paper. Investigations targeting more granular student interaction data could yield further insights into how and why student behavior changes, and may be key in creating models and training schedules that are robust in the presence of concept drift. There are also many more common educational models, such as affect detectors and psychometric models, which may have differing levels of robustness to changing student populations. Understanding which models lose accuracy over time and why is an essential step in understanding how student learning behavior changes over time.

REFERENCES

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. URL https://arxiv.org/abs/1607.06450.

Ryan S. J. D. Baker, Albert T. Corbett, and Vincent Aleven. More Accurate Student Modeling through Contextual Estimation of Slip and Guess Probabilities in Bayesian Knowledge Tracing. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, Beverley P. Woolf, Esma Aïmeur, Roger Nkambou, and Susanne Lajoie (eds.), *Intelligent Tutoring Systems*, volume 5091, pp. 406–415. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. ISBN 978-3-540-69130-3 978-3-540-69132-7. doi: 10.1007/978-3-540-69132-7_44. URL https://link.springer.com/10.1007/978-3-540-69132-7_44. Series Title: Lecture Notes in Computer Science.

Ryan S. J. d. Baker, Antonija Mitrović, and Moffat Mathews. Detecting Gaming the System in Constraint-Based Tutors. In Paul De Bra, Alfred Kobsa, and David Chin (eds.), *User Modeling, Adaptation, and Personalization*, pp. 267–278, Berlin, Heidelberg, 2010. Springer. ISBN 978-3-642-13470-8. doi: 10.1007/978-3-642-13470-8_25.

Benjamin S. Bloom. Learning for Mastery. Instruction and Curriculum. Regional Education Laboratory for the Carolinas and Virginia, Topical Papers and Reprints, Number 1. *Evaluation Comment*, 1(2), May 1968. URL https://eric.ed.gov/?id=eD053419. Publisher: Regional Education Laboratory for the Carolinas and Virginia, Mutual Plaza (Chapel Hill and Duke Sts.), Durham, N.C. 27701.

Anthony F. Botelho, Ryan S. Baker, and Neil T. Heffernan. Improving Sensor-Free Affect Detection Using Deep Learning. In Elisabeth André, Ryan Baker, Xiangen Hu, Ma. Mercedes T. Rodrigo, and Benedict du Boulay (eds.), *Artificial Intelligence in Education*, pp. 40–51, Cham, 2017. Springer International Publishing. ISBN 978-3-319-61425-0. doi: 10.1007/978-3-319-61425-0_4.

Tiffany Cai, Hongseok Namkoong, and Steve Yadlowsky. Diagnosing model performance under distribution shift. *Operations Research*, 74(2):898–916, 2026.

Rafael A. Calvo and Sidney D'Mello. Affect Detection: An Interdisciplinary Review of Models, Methods, and Their Applications. *IEEE Transactions on Affective Computing*, 1(1):18–37, January 2010. ISSN 1949-3045. doi: 10.1109/T-AFFC.2010.1. URL https://ieeexplore.ieee.org/abstract/document/5520655?casa_token=R5ZdM_JjtlwAAAAA:46_wqA0YVZcAkxwLdaQSnMxadxZ9yXi_93PWBoDO4ggyHL4x01w5oP9w2simhHjned69fDKL. Conference Name: IEEE Transactions on Affective Computing.

Hao Cen, Kenneth Koedinger, and Brian Junker. Learning Factors Analysis – A General Method for Cognitive Model Evaluation and Improvement. In Mitsuru Ikeda, Kevin D. Ashley, and Tak-Wai Chan (eds.), *Intelligent Tutoring Systems*, pp. 164–175, Berlin, Heidelberg, 2006. Springer. ISBN 978-3-540-35160-3. doi: 10.1007/11774303_17.

Wei Chu and Philip I. Pavlik Jr. The Predictiveness of PFA is Improved by Incorporating the Learner's Correct Response Time Fluctuation. July 2023. doi: 10.5281/zenodo.8115643. URL https://zenodo.org/records/8115643. ISBN: 9781733673648 Pages: 244–250 Publication Title: Proceedings of the 16th International Conference on Educational Data Mining Publisher: International Educational Data Mining Society.

Albert T. Corbett and John R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278, December 1994. ISSN 1573-1391. doi: 10.1007/BF01099821. URL https://doi.org/10.1007/BF01099821.

Russell Davidson and James Mackinnon. *Estimation and Inference in Econometrics*, volume 28. 01 1993. doi: 10.2307/2554780.

Oscar Blessed Deho, Lin Liu, Jiuyong Li, Jixue Liu, Chen Zhan, and Srecko Joksimovic. When the Past != The Future: Assessing the Impact of Dataset Drift on the Fairness of Learning Analytics Models. *IEEE Transactions on Learning Technologies*, 17:1007–1020, 2024. ISSN 1939-1382. doi: 10.1109/TLT.2024.3351352. URL https://ieeexplore.ieee.org/abstract/document/10384787/references#references. Conference Name: IEEE Transactions on Learning Technologies.

Robin Donnelly and Harry Anthony Patrinos. Learning loss during Covid-19: An early systematic review. *PROSPECTS*, 51(4):601–609, October 2022. ISSN 1573-9090. doi: 10.1007/s11125-021-09582-6. URL https://doi.org/10.1007/s11125-021-09582-6.

Theophile Gervet, Ken Koedinger, Jeff Schneider, and Tom Mitchell. When is Deep Learning the Best Approach to Knowledge Tracing? *Journal of Educational Data Mining*, 12(3):31–54, October 2020. ISSN 2157-2100. doi: 10.5281/zenodo.4143614. URL https://jedm.educationaldatamining.org. Number: 3.

Philip Hallinger and Ronald H. Heck. Exploring the journey of school improvement: classifying and analyzing patterns of change in school improvement processes and learning outcomes. *School Effectiveness and School Improvement*, 22(1):1–27, 2011. doi: 10.1080/09243453.2010.536322. URL https://doi.org/10.1080/09243453.2010.536322.

Neil T. Heffernan and Cristina Lindquist Heffernan. The ASSISTments Ecosystem: Building a Platform that Brings Scientists and Teachers Together for Minimally Invasive Research on Human Learning and Teaching. *International Journal of Artificial Intelligence in Education*, 24(4):470–497, December 2014. ISSN 1560-4306. doi: 10.1007/s40593-014-0024-x. URL https://doi.org/10.1007/s40593-014-0024-x.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

Chip Huyen. *Designing Machine Learning Systems*, chapter Data Distribution Shifts and Monitoring, pp. 225–262. O'Reilly Media, 2022.

Mohammad Khajah, Robert V. Lindsey, and Michael C. Mozer. How deep is knowledge tracing? In *arXiv.org*, March 2016. URL https://arxiv.org/abs/1604.02416v2.

Ralf Klinkenberg and Thorsten Joachims. Detecting concept drift with support vector machines. 2000.

Zlatko J. Kovačić. Early prediction of student success: Mining students' enrolment data. In *Proceedings of Informing Science & IT Education Conference (InSITE)*, 2010. URL https://repository.openpolytechnic.ac.nz/items/aab19a8d-d064-40da-b9b5-9f4c752fd6d9.

S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951. ISSN 00034851. URL http://www.jstor.org/stable/2236703.

Morgan Lee and Neil Heffernan. Concept Drift Detection for Knowledge Tracing. January 2025. doi: 10.5281/zenodo.15870129.

Morgan P Lee, Ethan Croteau, Ashish Gurung, Anthony F. Botelho, and Neil T. Heffernan. Knowledge Tracing over Time: A Longitudinal Analysis. International Educational Data Mining Society, 2023. URL https://eric.ed.gov/?id=ED630851. ERIC Number: ED630851.

Nathan Levin, Ryan Baker, Nidhi Nasiar, Fancsali Stephen, and Stephen Hutt. Evaluating Gaming Detector Model Robustness Over Time. *Proceedings of the 15th International Conference on Educational Data Mining, International Educational Data Mining Society*, January 2022. doi: 10.5281/zenodo.6852961. URL https://par.nsf.gov/biblio/10352945-evaluating-gaming-detector-model-robustness-over-time.

Sandeep Madireddy, Prasanna Balaprakash, Philip Carns, Robert Latham, Glenn K. Lockwood, Robert Ross, Shane Snyder, and Stefan M. Wild. Adaptive Learning for Concept Drift in Application Performance Modeling. In *Proceedings of the 48th International Conference on Parallel Processing*, ICPP '19, pp. 1–11, New York, NY, USA, August 2019. Association for Computing Machinery. ISBN 978-1-4503-6295-5. doi: 10.1145/3337821.3337922. URL https://dl.acm.org/doi/10.1145/3337821.3337922.

M.L. Menéndez, J.A. Pardo, L. Pardo, and M.C. Pardo. The Jensen-Shannon divergence. *Journal of the Franklin Institute*, 334(2):307–318, 1997. ISSN 0016-0032. doi: 10.1016/S0016-0032(96)00063-4.

Maria Nicolaidou and Mel Ainscow. Understanding Failing Schools: Perspectives from the inside. *School Effectiveness and School Improvement*, 16(3):229–248, September 2005. ISSN 0924-3453. doi: 10.1080/09243450500113647. URL https://doi.org/10.1080/09243450500113647. Publisher: Routledge _eprint: https://doi.org/10.1080/09243450500113647.

Shalini Pandey and George Karypis. A Self-Attentive model for Knowledge Tracing, July 2019. URL http://arxiv.org/abs/1907.06837. arXiv:1907.06837 [cs, stat].

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. *PyTorch: an imperative style, high-performance deep learning library*. Curran Associates Inc., Red Hook, NY, USA, 2019.

Philip I. Pavlik, Hao Cen, and Kenneth R. Koedinger. Performance Factors Analysis – A New Alternative to Knowledge Tracing. Technical report, 2009. URL https://eric.ed.gov/?id=ED506305. Publication Title: Online Submission ERIC Number: ED506305.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. Deep Knowledge Tracing. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/hash/bac9162b47c56fc8a4d2a519803d51b3-Abstract.html.

Thomas S. Popkewitz. Educational Standards: Mapping Who We Are and Are to Become. *Journal of the Learning Sciences*, 13(2):243–256, April 2004. ISSN 1050-8406, 1532-7809. doi: 10.1207/s15327809jls1302_7. URL https://www.tandfonline.com/doi/full/10.1207/s15327809jls1302_7.

Yumeng Qiu, Yingmei Qi, Hanyuan Lu, Zachary Pardos, and Neil Heffernan. Does Time Matter? Modeling the Effect of Time with Bayesian Knowledge Tracing. pp. 139–148, January 2011.

Steven Ritter, John R. Anderson, Kenneth R. Koedinger, and Albert Corbett. Cognitive Tutor: Applied research in mathematics education. *Psychonomic Bulletin & Review*, 14(2):249–255, April 2007. ISSN 1531-5320. doi: 10.3758/BF03194060. URL https://doi.org/10.3758/BF03194060.

Shuanghong Shen, Qi Liu, Zhenya Huang, Yonghe Zheng, Minghao Yin, Minjuan Wang, and Enhong Chen. A Survey of Knowledge Tracing: Models, Variants, and Applications. *IEEE Transactions on Learning Technologies*, 17:1898–1919, 2024. ISSN 1939-1382. doi: 10.1109/TLT.2024.3383325. URL https://ieeexplore.ieee.org/abstract/document/10494775. Conference Name: IEEE Transactions on Learning Technologies.

Kirk Vanacore, Ashish Gurung, Adam Sales, and Neil T Heffernan. The effect of assistance on gamers: Assessing the impact of on-demand hints & feedback availability on learning for students who game the system. In *Proceedings of the 14th Learning Analytics and Knowledge Conference*, pp. 462–472, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need, June 2017. URL http://arxiv.org/abs/1706.03762. arXiv:1706.03762.

Jill-Jenn Vie and Hisashi Kashima. Knowledge Tracing Machines: Factorization Machines for Knowledge Tracing. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):750–757, July 2019. ISSN 2374-3468. doi: 10.1609/aaai.v33i01.3301750. URL https://ojs.aaai.org/index.php/AAAI/article/view/3853. Number: 01.

Xiaolu Wang, Zhi Wang, Wei Shao, Chunfu Jia, and Xiang Li. Explaining Concept Drift of Deep Learning Models. In Jaideep Vaidya, Xiao Zhang, and Jin Li (eds.), *Cyberspace Safety and Security*, pp. 524–534, Cham, 2019a. Springer International Publishing. ISBN 978-3-030-37352-8. doi: 10.1007/978-3-030-37352-8_46.

Zhiwei Wang, Xiaoqin Feng, Jiliang Tang, Gale Yan Huang, and Zitao Liu. Deep Knowledge Tracing with Side Information. In Seiji Isotani, Eva Millán, Amy Ogan, Peter Hastings, Bruce McLaren, and Rose Luckin (eds.), *Artificial Intelligence in Education*, pp. 303–308, Cham, 2019b. Springer International Publishing. ISBN 978-3-030-23207-8. doi: 10.1007/978-3-030-23207-8_56.

Kimberly Williamson and René F. Kizilcec. Effects of Algorithmic Transparency in Bayesian Knowledge Tracing on Trust and Perceived Accuracy. Technical report, International Educational Data Mining Society, 2021. URL https://eric.ed.gov/?id=ED615541. ERIC Number: ED615541.

Chun-Kit Yeung. Deep-IRT: Make Deep Learning Based Knowledge Tracing Explainable Using Item Response Theory, April 2019. URL http://arxiv.org/abs/1904.11738. arXiv:1904.11738.

Michael V. Yudelson, Kenneth R. Koedinger, and Geoffrey J. Gordon. Individualized Bayesian Knowledge Tracing Models. In H. Chad Lane, Kalina Yacef, Jack Mostow, and Philip Pavlik (eds.), *Artificial Intelligence in Education*, pp. 171–180, Berlin, Heidelberg, 2013. Springer. ISBN 978-3-642-39112-5. doi: 10.1007/978-3-642-39112-5_18.

Bilal Yurdakul and Joshua Naranjo. Statistical properties of the population stability index. *Journal of Risk Model Validation*, 12 2020. doi: 10.21314/JRMV.2020.227.

Liang Zhang, Xiaolu Xiong, Siyuan Zhao, Anthony Botelho, and Neil T. Heffernan. Incorporating Rich Features into Deep Knowledge Tracing. In *Proceedings of the Fourth (2017) ACM Conference on Learning @ Scale*, L@S '17, pp. 169–172, New York, NY, USA, April 2017. Association for Computing Machinery. ISBN 978-1-4503-4450-0. doi: 10.1145/3051457.3053976. URL https://dl.acm.org/doi/10.1145/3051457.3053976.

AI USE STATEMENT

In this work, we did not use any generative AI tools at all. All code was written perhaps inefficiently but by human hand. This entire manuscript went through extensive revision exclusively under human oversight. We take responsibility for the final content of this work, including text, claims or artifacts all of which were produced without the aid of generative AI. generative AI.

ETHICS STATEMENT

This work handles anonymized student data. However, no students are able to be identified since no references to OLP usernames are used in this data. No form of de-anonymization is a concern with this data, since the only points of data that are available are purely binary and do not rely on any identifiable features of the students.

REPRODUCIBILITY STATEMENT

All data used in this work is released for study and experimentation at the following anonymized link. All code and implementation is released for the purpose of reproducibility at this anonymized link.

A Model Implementations

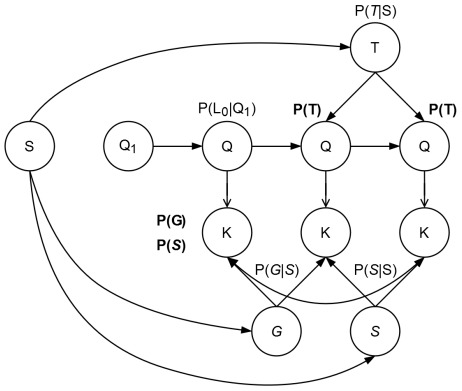
Each of our models was implemented in Python 3.12[4], with the following differences. BKT was implemented with the forgetting parameter enabled via the hmmlearn package. After fitting models for each available KC in the training set, learned parameters were averaged to make a "best guess" KT model in the case of evaluating KCs that were not present in the training set. PFA was implemented using scikit-learn Pedregosa et al. (2011), fitting separate covariates for wins and fails for each KC, along with a KC level intercept and parameters for KCs not present in the training set. Both SAKT and DKT were implemented in PyTorch Paszke et al. (2019) and trained on NVIDIA A100 GPUs. Visual representations of model architectures can be found in figures 2a–2c. For a brief mathematical explanation of each model, please refer to their respective sections in appendix B

B Hyperparameter Tuning

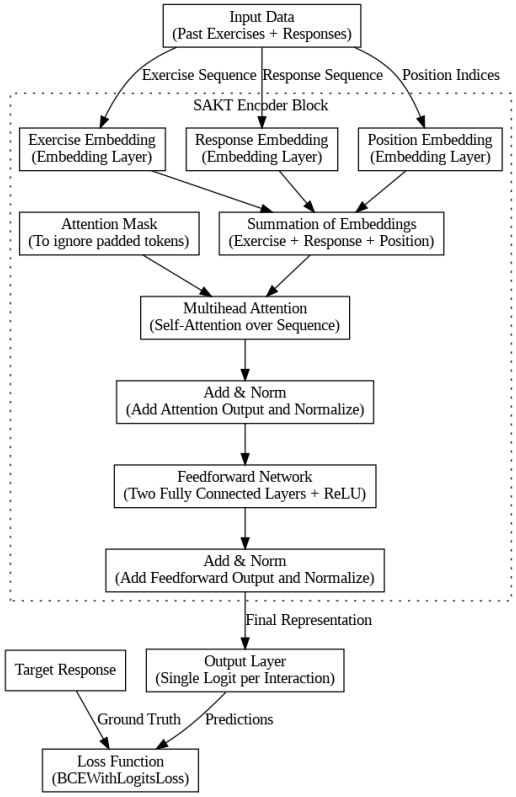
To tune the hyperparameters of our deep models, we used Bayesian hyperparameter optimization. Hyperparameters tuned for each model can be found in our appendix, in table 3. Rather than tuning these hyperparameters with the samples constructed for our main study, we constructed a new validation sample by sampling 100,000 unique assignment logs from all available academic years. This was done to avoid overtuning hyperparameters to each year, instead aiming to find the optimal hyperparameters for use across years to give our models the best chance of generalizing. Using the optuna package Akiba et al. (2019), we performed a four-fold cross-validation using the validation sample for each sampled hyperparameter combination, terminating after fifty rounds of cross-validation were performed.

[4]These implementations, along with analysis code, are available here

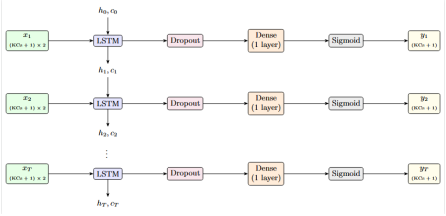
C Model Architecture Diagrams



(a) BKT



(b) SAKT



(c) DKT

Figure 2: Model architectures

D LEARNED HYPERPARAMETERS

Parameter	Range	DKT	SAKT-E	SAKT-KC
num_steps	[20,100]	40	60	100
batch_size	[16,64]	16	64	48
d_model	[64,512]	96	352	128
num_epochs	DKT [100, 300] SAKT [10, 40]	100	23	25
dropout_rate	[0.1, 0.5]	0.278	0.47	0.188
learn_rate	[1e-4, 1e-2]	2e-3	1e-4	1e-4
reg_lambda	[1e-6, 1e-2]	1.4e-5	n/a	n/a
num_heads	[2,4,8,16,32]	n/a	8	16
learn_decay_rate	[0.7, 0.99]	n/a	0.7	0.868

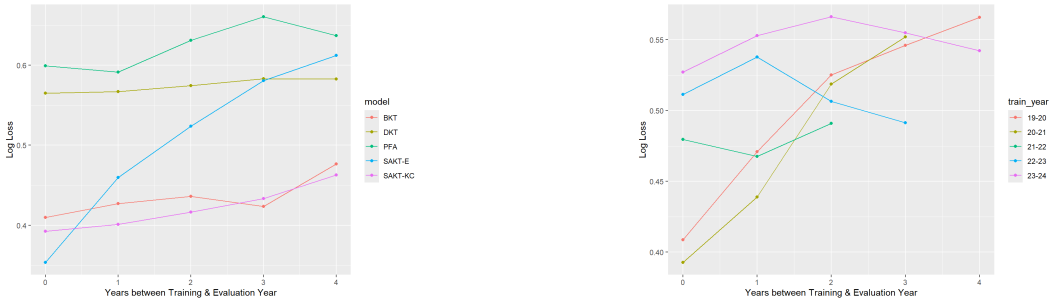
Table 3: Learned Hyperparameters for DKT, SAKT-E, and SAKT-KC

E SPEARMAN COEFFICIENTS

Model	AUC	Logistic Loss	F1 Score
BKT	-0.332***	0.185*	-0.159*
PFA	-0.365***	0.284***	-0.161*
DKT	-0.599***	0.346***	-0.205**
SAKT-KC	-0.317***	0.282***	-0.270***
SAKT-E	-0.685***	0.636***	-0.642***

Table 4: Estimated Spearman's $\boldsymbol{\rho}$ for each model fit statistic (* = $p < 0.05$, ** = $p < 0.01$, *** = $p < 0.001$).

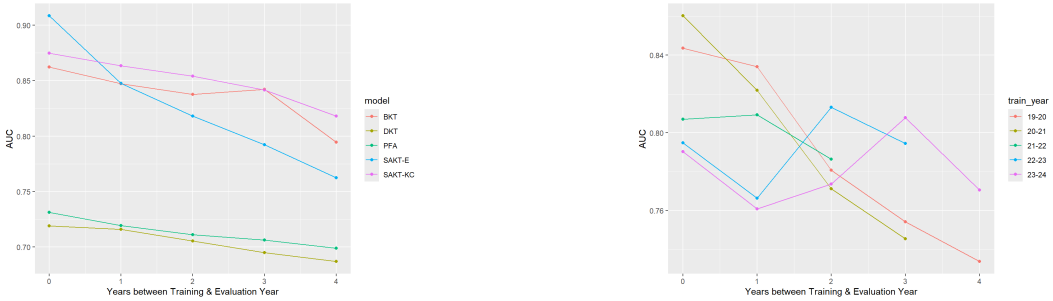
F LOS LOSS PER MODEL PER TRAINING DATA AGE



(a) (b)

Figure 3: Mean Log Loss measurements vs. training data age broken down by model (3a) and training year (3b).

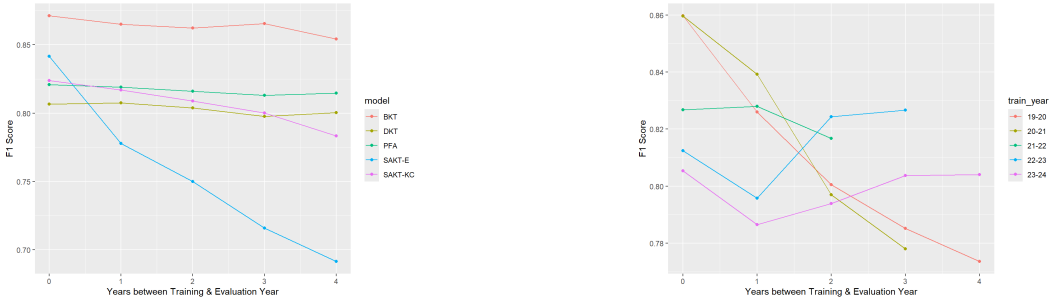
G AUC PER MODEL PER TRAINING DATA AGE



(a) (b)

Figure 4: Mean AUC measurements vs. training data age broken down by model (4a) and training year (4b).

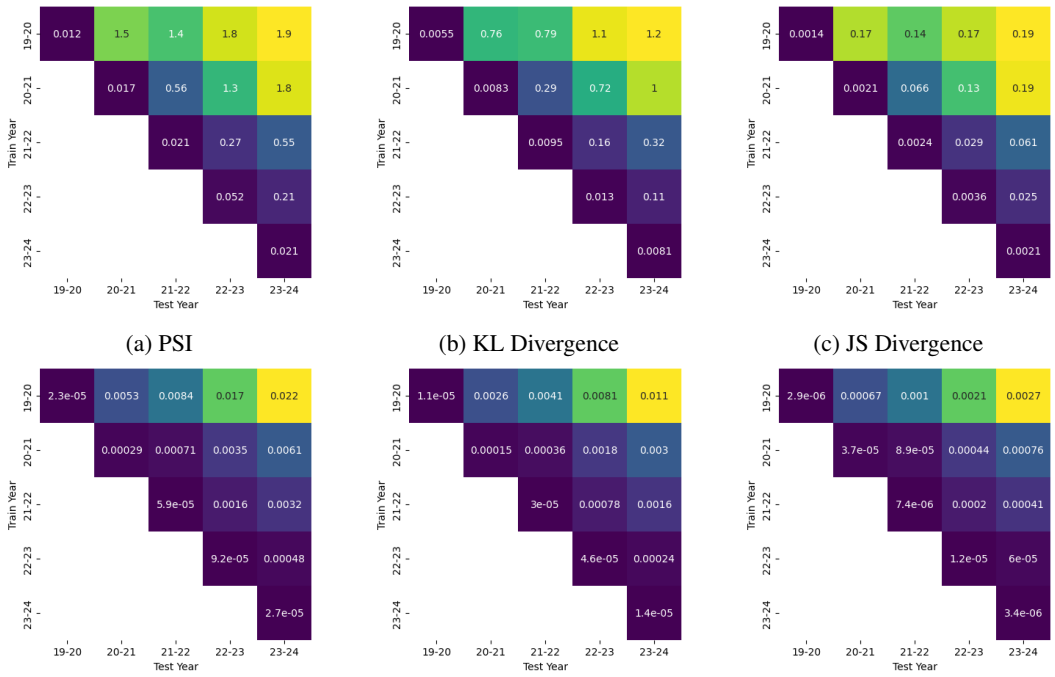
H F1 PER MODEL PER TRAINING DATA AGE



(a) (b)

Figure 5: Mean F1 score measurements vs. training data age broken down by model (5a) and training year (5b).

I DRIFT STATISTICS FOR SKILLS

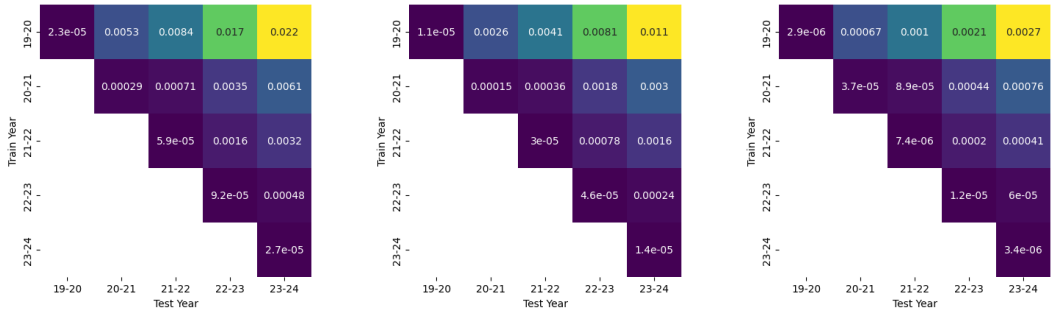


(a) PSI (b) KL Divergence (c) JS Divergence

(d) PSI (e) KL Divergence (f) JS Divergence

Figure 6: Drift statistics for skills (6a-6c) and scores (7a-7c)

J DRIFT STATISTICS FOR SCORES



(a) PSI (b) KL Divergence (c) JS Divergence

Figure 7: Drift statistics for scores

K LOG LOSS REGRESSION RESULTS

Coefficient	Estimate	Std. Err.	p-value
Intercept	0.364	$6.09e-3$	$< 2.2e-16$
Years Between (YB)	$-3.09e-3$	$3.39e-3$	0.363
DKT	0.151	$5.13e-3$	$< 2.2e-16$
PFA	0.177	$5.55e-3$	$< 2.2e-16$
SAKT-E	-0.034	$7.81e-3$	1.30e-5
SAKT-KC	-0.025	$5.11e-3$	$1.14e-6$
Train 20-21	-0.051	$6.97e-3$	$3.00e-13$
Train 21-22	-0.032	$8.01e-3$	$8.76e-5$
Train 22-23	-6.17e-3	8.31e-3	0.458
Train 23-24	2.55e-4	9.64e-3	0.979
Eval 20-21	0.031	6.77e-3	6.02e-6
Eval 21-22	0.085	5.52e-3	$< 2.2e-16$
Eval 22-23	0.102	5.67e-3	$< 2.2e-16$
Eval 23-24	0.110	6.12e-3	$< 2.2e-16$
YB $\times$ DKT	$-5.02e-3$	$2.77e-3$	0.070
YB $\times$ PFA	7.17e-3	3.79e-3	0.059
YB $\times$ SAKT-E	0.057	5.46e-3	$< 2.2e-16$
YB $\times$ SAKT-KC	5.02e-3	3.57e-3	0.113
YB $\times$Train 20-21	0.016	3.99e-3	7.56e-5
YB $\times$ Train 21-22	0.013	4.82e-3	8.29e-3
YB $\times$ Train 22-23	0.018	4.67e-3	8.81e-5
YB $\times$ Train 23-24	0.023	4.70e-3	1.72e-6

Table 5: Regression results for Log Loss (adjusted $\mathbf{R}^2 = 0.749$), with BKT, Train Year 19-20, and Eval Year 19-20 as reference categories.

L AUC REGRESSION RESULTS

Coefficient	Estimate	Std. Err.	p-value
Intercept	0.885	2.64e-3	¡2.2e-16
Years Between (YB)	-4.29e-3	1.46e-3	3.72e-3
DKT	−0.140	2.77e-3	¡2.2e-16
PFA	−0.133	2.48e-3	¡2.2e-16
SAKT-E	0.032	3.91e-3	1.03e-15
SAKT-KC	0.015	2.88e-3	4.45e-7
Train 20-21	0.014	2.66e-3	1.08e-7
Train 21-22	0.011	3.59e-3	2.56e-3
Train 22-23	0.012	3.62e-3	1.20e-3
Train 23-24	0.019	4.30e-3	1.18e-3
Eval 20-21	6.74e-3	2.36e-3	4.32e-3
Eval 21-22	−0.043	2.24e-3	¡2.2e-16
Eval 22-23	−0.065	2.48e-3	¡2.2e-16
Eval 23-24	−0.075	2.71e-3	¡2.2e-16
YB × DKT	3.29e-3	1.55e-3	0.034
YB × PFA	3.94e-3	1.38e-3	4.34e-3
YB × SAKT-E	−0.024	2.08e-3	¡2.2e-16
YB × SAKT-KC	-7.76e-4	1.79e-3	0.665
YB × Train 20-21	-2.83e-3	1.64e-3	0.084
YB × Train 21-22	-5.48e-3	2.21e-3	0.013
YB × Train 22-23	−0.013	2.10e-3	3.63e-10
YB × Train 23-24	−0.014	1.92e-3	1.33e-12

Table 6: Regression results for AUC (adjusted $\mathbf{R}^2 = 0.914$, with BKT, Train Year 19-20, and Eval Year 19-20 as reference categories.

M F1 Regression Results

Coefficient	Estimate	Std. Err.	p-value
Intercept	0.892	2.46e-3	¡2.2e-16
Years Between (YB)	4.03e-3	1.27e-3	1.51e-3
DKT	−0.061	1.96e-3	¡2.2e-16
PFA	−0.049	1.50e-3	¡2.2e-16
SAKT-E	−0.043	3.90e-3	¡2.2e-16
SAKT-KC	−0.044	2.38e-3	¡2.2e-16
Train 20-21	0.019	3.76e-3	3.30e-7
Train 21-22	0.020	3.95e-3	7.18e-7
Train 22-23	0.014	3.96e-3	4.08e-4
Train 23-24	0.014	4.56e-3	2.06e-3
Eval 20-21	−0.015	2.85e-3	3.74e-7
Eval 21-22	−0.043	2.33e-3	¡2.2e-16
Eval 22-23	−0.058	2.61e-3	¡2.2e-16
Eval 23-24	−0.065	2.78e-3	¡2.2e-16
YB × DKT	3.34e-4	9.15e-4	0.716
YB × PFA	8.42e-4	7.96e-4	0.290
YB × SAKT-E	−0.034	2.47e-3	¡2.2e-16
YB × SAKT-KC	-6.25e-3	1.44e-3	1.49e-5
YB ×Train 20-21	-5.73e-3	2.07e-3	5.75e-3
YB × Train 21-22	-7.10e-3	2.18e-3	1.13e-3
YB × Train 22-23	−0.10	2.20e-3	3.47e-6
YB × Train 23-24	−0.012	2.07e-3	4.99e-9

Table 7: Regression results for F1 score (adjusted $\mathbf{R}^2 = 0.755$), with BKT, Train Year 19-20, and Eval Year 19-20 as reference categories.

N Thresholds for Data Kept

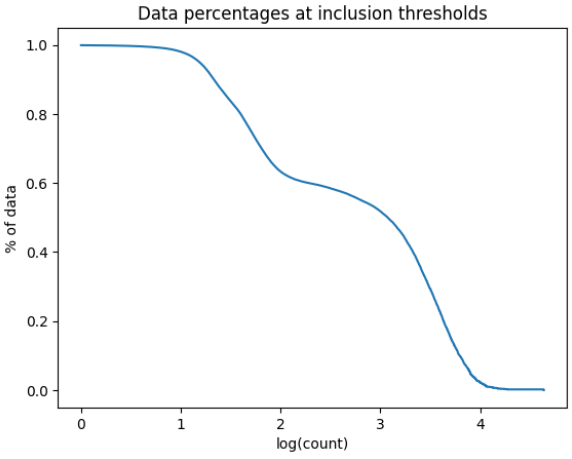


Figure 8: Percent of data kept when filtering problem sets with fewer than *count* student/assignment interactions

O Model Explanations

O.1 BKT

$$P(L_1) = P(L_0)$$
$$P(L_t|C_t = 1) = \frac{P(L_t) \cdot (1 - P(S))}{P(L_t) \cdot (1 - P(S)) + (1 - P(L_t)) \cdot P(G)}$$
$$P(L_t|C_t = 0) = \frac{P(L_t) \cdot P(S)}{P(L_t) \cdot P(S) + (1 - P(L_t)) \cdot (1 - P(G))}$$
$$P(L_{t+1} = P(L_t|C_t) \cdot (1 - P(F)) + (1 - P(L_t|C_t)) \cdot P(T)$$
$$P(C_{t+1}) = P(L_{t+1}) \cdot (1 - P(S)) + (1 - P(L_{t+1})) \cdot P(G)$$

$P(L_t)$ is the probability that the student has mastered the relevant KC at time step t, $P(C_t)$ is the probability of said student getting the next exercise of that KC correct. $P(L_0)$, $P(G)$, $P(S)$, $P(T)$, and $P(F)$ are the probabilities for prior knowledge, guess, slip, transfer (learning), and forgetting, respectively, and are learned via expectation maximization Corbett & Anderson (1994).

O.2 PFA

$$m(i, j \in KCs, s, f) = \sum_{j \in KCs} \beta_j + \gamma_j s_{i,j} + \rho_j f_{i,j}$$
$$P(m) = \frac{1}{1 + e^{-m}} = \sigma(m)$$

In this model, i represents a student, while j represents a KC. $s_{i,j}$ and $f_{i,j}$ represent collective successes and failures of student i on KC j Pavlik et al. (2009).

O.3 DKT

Our implementation of DKT uses the standard LSTM Hochreiter & Schmidhuber (1997) fed into a dense output layer. An LSTM computes the following function:

$$i_t = \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi})$$
$$f_t = \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf})$$
$$g_t = \tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg})$$
$$o_t = \sigma((W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho})$$
$$c_t = f_t \odot c_{t-1} + i_t \odot g_t$$
$$h_t = o_t \odot \tanh(c_t)$$

where i_t, f_t, g_t, o_t, c_t, h_t are the input, forget, output, cell, and hidden state at time t, x_t is the model input at time t, σ is the sigmoid function and $\odot$ is the element-wise product. All instances of W and b are trainable parameters.

O.4 SAKT

Our implementation of SAKT uses the architecture initially described in Pandey et al's SAKT paper Pandey & Karypis (2019). The architecture using exercises as features is fittingly called SAKT-E. An added change from the original paper is the addition of KCs as a potential feature for embedding, for which the architecture is appropriately titled SAKT-KC.

O.4.1 Embedding Layers

$$E_x = \{e_t \mid e_t = \hat{E}[E_t],\ E_t \in \{0, 1\}\}, \quad \hat{E} \in \mathbb{R}^{B \times T \times d}$$
$$R_x = \{r_t \mid r_t = \hat{R}[R_t],\ R_t \in \{0, 1\}\}, \quad \hat{R} \in \mathbb{R}^{B \times 2 \times d}$$
$$P_x = \{p_t \mid p_t = E_g[t],\ t \in [1, T]\}, \quad E_g \in \mathbb{R}^{B \times T \times d}$$
$$Z = E_x + R_x + P_x$$

Notations	Descriptions
B	Batch size
T	Sequence length
d	Embedding dimension
K	Total number of exercise/KC IDs
X	Past learner responses, $X \in \mathbb{R}^{B\times T\times 2}$
R	Past binary responses $R \in \{0, 1\}$
y	Ground truth labels ($y \in \{0, 1\}$)
C	Binary ground truth labels ($C \in \{0, 1\}$)
L	Number of unique exercises
E_x	Trainable exercise/KC ID embedding matrix
P_x	Trainable position embedding matrix
R_x	Trainable Response embedding matrix
W	Trainable Weight Matrix

Table 8: Notations for SAKT Architecture

O.4.2 Self-Attention Mechanism

$$Q = ZW^Q, \quad K = ZW^K, \quad V = ZW^V$$
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V$$
$$\text{Head}_h = \text{Attention}(QW_h^Q, KW_h^K, VW_h^V)$$
$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{Head}_1, \ldots, \text{Head}_H)W^O$$

Here, softmax refers to the softmax function, formally defined as

$$Softmax(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$$

where n is the number of classes.

O.4.3 Residual Connections and LayerNorm

$$O = \text{LayerNorm}(Z + \text{MultiHead}(Q, K, V))$$
$$F(O) = \max(0, OW_1 + b_1)W_2 + b_2$$
$$O = \text{LayerNorm}(O + F(O))$$
$$\hat{y}_t = \sigma(W_y O_t + b_y)$$

LayerNorm here refers to the function initially defined in Ba, et.al's paper Ba et al. (2016) on normalization of neurons, which reduces training time significantly. This formula is given by $y = \frac{x-E[x]}{\sqrt{Var(x)+\epsilon}}$ where ϵ is a small value added to avoid division by 0. In the calculation of our binary prediction $\hat{y}$, σ represents the sigmoid activation function, or $\frac{1}{1+e^{-x}}$ to squash the output into the range of (0, 1) for evalution with the loss function, binary cross-entropy loss.