Modelling Learner Competencies: Does MKM Have an Influence?

Jonas Betzendahl and Michael Kohlhase

Computer Science, FAU Erlangen-Nürnberg, Germany

Abstract. To be effective, learning support systems need to be able to adapt to the individual learner or cohort, just as good teachers do. A clear and reliable idea of which parts of the domain of discourse any given learner already knows how much about is central to tailoring learning services to be useful and successful. In this paper, we describe and discuss the three different approaches to learner modeling used in the ALeA system, a learning platform in development and in use at FAU. This includes the structure of the underlying model and its foundations in didactics research as well as the update dynamics of each approach. We also present an empirical evaluation of the three approaches with each other and against performance in weekly quizzes. We conclude with a discussion of how this framework of evaluating learner modeling design choices can also function as a data-driven testing ground for additional learner model features.

Keywords: Learner Model · User Model · Digital Twin · EduTech

1 Introduction

There is a crucial difference between having a complex mathematical theory explained by a colleague (Sarah) and reading about it in an article or a book (most of us prefer the former to the latter): Even though the content of the article may be more carefully worded, an explanation by a colleague is more efficient because it may be tailored to me and my particular background. Most of us have enjoyed situations where a whole chapter in a book has been condensed to a single sentence like "*I know that you are familiar with the concept of a bar; now a foobar is the same as a bar but with an additional foo spliced in modulo the kernel.*", without losing information.

Here, Sarah made use of the fact that she has a model of my previous knowledge and preferences, and has tailored an explanation to them, which is impossible for the author of an article or a book, where instead of a particular addressee we have to tailor the exposition to the "average reader", who may be quite different from me, forcing me to skip over expositions of things I already know or look up material I do not know elsewhere. In fact my colleague can be optimistic in simplifying explanations, since she knows that I will ask back, if and only if her model of my competency is off.

The crucial ingredient in Sarah's – and indeed most human's – communication ability is that she has a model – we call it a **competency model**, as it models her interlocutor's scientific competencies – and adapts her communication to that. Moreover, this competency model is updated with any our interactions. So, if we want to make scientific communication more like a blackboard discussion situation while preserving the precision and polish of the underlying knowledge presentation, we will need to add a competency modeling component to the mix.

While we contend that competency modeling is – or at least should be – a crucial ingredient for general (scientific) communication, it has only really been studied in the context of computer-supported education systems; see **??**. Indeed, the only difference between communication and teaching is that in the former, the roles of instructor and learner are more fluid, and may change frequently in a single discussion.

Contribution In this paper we present a particular learner modeling approach we have developed for the ALeA system (Adaptive Learning Assistant; see [Ber+23; ALeA] and evaluate it on real student data. ALeA has been used in multiple Computer Science-related courses at FAU Erlangen-Nürnberg totaling up to 1000 students per semester for the last three years. So we can evaluate the induced learner models against the actual exam grades: the results show a surprisingly good match for the best of our three learner models. This is important for ALeA, as most of the learning support services in the system use the learner model in some form.

Overview In the next section, we will briefly recap the state of the art in learner modeling and introduce the concrete approach in the ALeA system in Section 2. Section 3 evaluates the learner models and Section 4 concludes the paper and discusses future research.

Acknowledgments The work reported in this article was conducted as part of the VoLL-KI project (see `https://voll-ki.de`) funded by the German Research/Education Ministry under grant 16DHBKI089 and the ongoing FAUstairs project (see `https://faustairs.fau.de`) funded by the Stiftung Innovation in der Hochschullehre under grant StIL:1001-3096.

2 Learner Modeling in ALeA

The learner models in other systems are usually distinguished by exactly what data they include (which is part of the research efforts by [PMV22] and [AAH14] amongst others), what parts of the learners they model and how (for a comprehensive literature review as well as a classification of different approaches, please see [Bö+25]), as well as who can use the data and for what (for example, open learner models (OLM) are learner models that the learners themselves have read or even write access to, see for example [AB14]).

One of the central tenets of the ALeA approach is that the central algorithms should operate on **symbols** – semantic identifiers for (mathematical) concepts, objects, and their relations. Symbols can have notations for the representation in formulae and verbalizations, the words that represent them in text. Both are introduced in definitions which also specify the semantics of the symbols building on that of others inducing a **terminological dependency relation** on symbols. In ALeA, the symbols are also clustered into modules/theories and organized in a theory graph, whose edges are theory morphisms that interpret one theory in terms of another. We think of this theory graph consisting of symbols, their definitions, theories, and interpretations a **domain model**.

All **learning objects** – any media fragments that are used for learning, e.g. lectures, slides, videos, images, practice problems, exams – are annotated with the respective symbols. As the learning objects are concrete formulations of the underlying knowledge from the domain models for specific learning situations, we call their totality the **formulation model**. The **didactic model** interprets, classifies, and relates learning objects as learning tasks with respect to the (intended) changes in competency in learners see Section 2.1 and [Loh+23] for details.

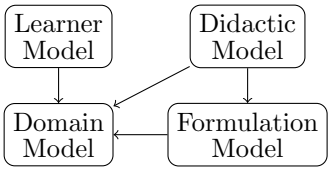

Fig. 1. Models for Teaching

While the latter two models (on the right hand side of Figure 1) form the motivation and application ground for the **learner model** this paper centers upon, the learner model itself only refers to the domain model – see Section 2.2 below. Indeed the learner's interaction with the formulation model and the interpretation of the didactic effects are used to update the learner model. The learner model in turn informs the informs the generation of all interactive course materials and all the learning support services; that makes the learner model – and the quality of its predictions – a central component of ALeA.

Before we go into the evaluation, we present more details of the setup in ALeA:

2.1 Educational Background

The actual model we use for the goals, progress, and success of learners is based on Bloom's learning taxonomy [Blo56], adapted as in Fuller et al. in [Ful+07]. In particular, this model distinguishes between two categories of skills: those for knowledge interpretation and those for knowledge production. This taxonomy splits a given learners competence into six different *cognitive dimensions*:

1.	`Remember`	**2.**	`Understand`	**3.**	`Apply`
4.	`Analyse`	**5.**	`Evaluate`	**6.**	`Create`

This split does justice to the fact that learners interact with knowledge in many different ways and can have consistently different levels of success with each of them. Someone might be able to recall a complicated definition from

memory perfectly, yet struggle to create an example or to identify the instance of said definition in a text problem. Modeling competence with just one numeric (or worse, binary) value would be overly reductive and weaken our ability to provide targeted learning services.

A more extensive exploration of the didactic and educational foundations of our approach can be found in [Loh+23].

2.2 Structure

Learner models in ALeA center around *concepts*. A concept is any element of the domain of discourse that someone might learn about. These concepts are represented by IRIs, such as:

```
http://mathhub.info?a=smglom/logic&p=mod&m=lambda-calculus&s=lambda calculus
```

Concepts are paired with unique learner identifiers (such as a hash value of their university login handle or the e-mail address used on sign-up), and one of the previously mentioned cognitive dimensions. For each such combination, the learner model keeps a *competence* and a *confidence* value, both always between 0 and 1.

The complete structure of the learner model can thus be understood as the following mathematical mapping:

$$\mathrm{C} \times \mathtt{Id}_L \times \mathbb{D} \to (\mathbb{I} \times \mathbb{I})$$

where C is the set of *concepts*, $\mathtt{Id}_L$ is the set of learner identifiers, $\mathbb{D}$ is the set of **Cognitive Dimensions** (see Section 2.1) and $\mathbb{I}$ is the *interval*, that is to say $[0, 1]$.

The competence value can be understood either as the grade of proficiency the student has reached in respect to this concept in this cognitive dimension (0 being none at all and 1 being complete mastery) or, more operationally, as the *probability* that the student can solve a problem that relies (only) on this concept in (only) this cognitive dimension. Competence is adapted after every learner interaction that involves the concept and dimension in question. If the student displays knowledge and mastery over the concept, it is generally adapted up, towards 1. In case of incorrect answers or indicated lack of understanding, it is generally adapted down, towards 0. The precise adaptations that are performed are dependent on which of the several modeling frameworks that are available is used (see Section 2.3 for details).

The confidence value is to be understood as our own confidence in the corresponding competence value for the same learner, concept and dimension. That is to say we are modeling how much trust we place in the corresponding competence value (0 being equivalent to a complete guess and 1 to absolute certainty). Additional interaction data generally tends to increase the confidence of the system, though not necessarily so. If the newest interaction is clearly incongruent with the existing model (e.g. a learner with high competence and high confidence entries answers question completely incorrect), it is also possible for the

competence value to fall with additional data. As always, these details will be discussed more closely in Section 2.3.

2.3 Learner Model Updates

If a learner interacts successfully with a learning object (such as answering a question correctly or indicating they have read and understood a definition), it is clear that the value of their learner model should increase but unclear by how much exactly.

Learner Modeling in ALeA can pick from one of three approaches to determine the new value (only one of which is used in production at any given time). In the following, we want to showcase the inner workings of all of them to better be able to compare them with each other. For this, we will use the running example of a learner answering a practice problem. After such an interaction, the learner modeling system would be supplied with the concepts and cognitive dimensions involved in that specific practice problem as well as with a score $s \in [0, 1]$ indicating the degree of success the learner achieved (allowing for more precision than binary classification into classes like "correct"/"incorrect"). For example, if a learner managed to answer three out of four (equally important) True/False questions in a practice problem, their score for this interaction might conceivably be $s = 0.75$.

Trivial Learner Modeling What we call the *trivial model* is the simplest of the three different approaches. It takes the most straight-forward and naïve approach to interpreting the interaction data and does the bare minimum with it. As such, it is not actually intended to be used in production, but it can serve as a common point of reference and comparison for the other models that *do* capture more complexities.

The trivial model takes whatever evidence comes in in the form of interaction events and treats it as the new gospel. For example: learner self-assessments always set the value to exactly the value that was given. A question answered with a score of 0.5 (meaning that it was awarded 50% of the maximum points) will set the value for all concepts and cognitive dimensions involved in the exercise to exactly 0.5. Maybe the only wrinkle of actual complexity is that the trivial learner model does still sharply distinguish between cognitive dimensions.

The trivial learner model utilises no connections between concepts (like dependencies, see below) to fill more of the learner model or infer information about one concept from the learner model value of another concept. It also does not take into account previous values of itself, everything is completely determined by the youngest learner interaction.

Confidence also does not play a role in the trivial learner model as it is never included in any computation.

Intuitive Learner Modeling The second approach towards learner modeling in ALeA is the *intuitive model* (based upon the *intuition* of the authors what an update should look like). It updates more than just a single concept at a time: If a learner knows about one concept (e.g. equilateral triangles), they probably also have some understanding of closely related concepts (e.g. triangles in general), even though these were not specifically annotated in the learning object in question.

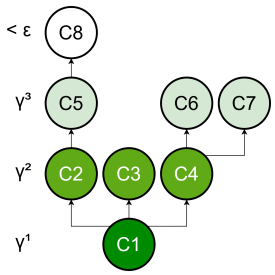

Fig. 2. A hypothetical dependency structure. Concept $C1$ depends on $C2$, $C3$ and $C4$.

To determine which concepts are "closely related" in the above sense, we utilize what we call concept *dependencies*. A concept C_1 *depends* on a different concept C_2 if and only if C_2 appears in a definition of C_1 (see also Figure 2).

In an update step for the intuitive model, the system would first calculate the difference between the *current* learner model and the *score* associated with the interaction.

To determine how to change, we use an empirically determined dampening constant γ ($\in [0,1]$). The update to the concepts directly annotated to the learning object in question moves the value towards the score, by an amount equal to γ multiplied by the total difference between score and current model. For example, say that for a γ of $\frac{2}{3}$, the current learner model for some concept had a value of 0.4 and the score the learner achieved in answering a problem was 0.7. This would mean that the computed new value would be $0.4 + \gamma \cdot (0.7 - 0.4) = 0.6$.

After updating the directly annotated concepts, the system then identifies their dependencies and performs the same update, but with a dampening constant of γ^2. This process of "rippling updates" continues until the change in competency values falls below the (also empirically determined) cut-off value ε.

Preliminary tests have shown that a value of just below 0.5 for γ and 0.15 for ε work well in practice.

Confidence values are set to $1 - |\delta|$, where δ is the difference between old and new competence value for that concept in that cognitive dimension. This reflects that when learner models change by a lot, confidence in the current value are low (and if they change little, confidence is high).

Bayesian Learner Modeling Another approach extant in the literature (e.g. in [Bij25]) is learner modeling based around Bayesian conjugate priors (see also [Gun19]). These ideas also inform the third model in our ecosystem, which we call the *Bayesian model*.

At the heart of this approach lie β-distributions, a family of probability distributions that are each defined by two so called *shape parameters*, commonly named α and β. For our particular case, the shape parameters of the distribution will be given by the number of interactions a student has had with the learning

platform that indicate they do or do not understand (or remember, or know how to apply, ...) the concept in question.

The decision to choose β-distributions over other alternatives (like Gaussian distributions) was made to take advantage of the concept of conjugate priors. In the context of Bayesian inference, if the *prior* and *posterior* probabilities belong to the same family of distribution for given a likelihood function, they are called *conjugate* to each other.[Bis06; Gun19].

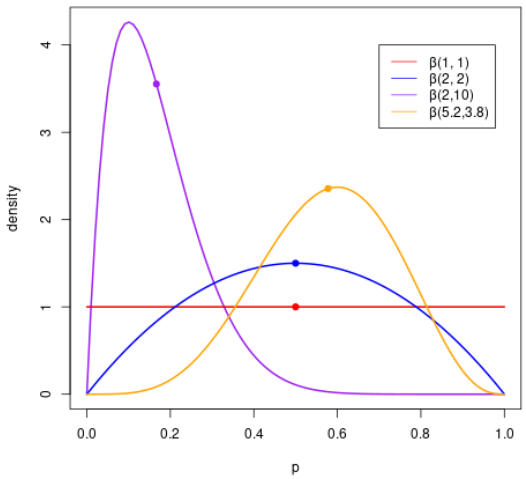

Fig. 3. Probability densities of β-distributions, their respective shape parameters and expected values.

This benefits us in the sense that it keeps the necessary numerical computations for updating learner models extremely simple and avoids the effort of numerical integration (see below). Also, since the posterior of one update can easily serve as the prior for the next, it is also excellently suited for models of sequential learning like ours.

For a β-distributed random variable X, the mean (i.e. the expected value) and the variance of the distribution are as follows. Both of these formulas will play important roles in the computation of our model values.

$$E[X] = \frac{\alpha}{\alpha + \beta} \quad (1) \qquad Var[X] = \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)} \quad (2)$$

For more on the nature of β-distributions and their role as conjugate priors in Bayesian belief modeling, see [Bis06].

While this research effort also makes use of the conjugate prior "trick", as Bijl does in [Bij25], the rest of the modeling approach is significantly different. Bijl uses a Dynamic Bayesian Network, whereas we decided to go into another direction concerning structure and incorporation of new interaction data. Updates to one concept do still influence the values for other concepts (through the "ripple-updates" introduced in the intuitive and Bayesian models), however the connections along which these changes propagate are purely derived from the semantic domain model, not an artifact of mere probabilities. In short: in this approach we are modeling student interactions as fractional Bernoulli trials. "Fractional" meaning in this context that we allow for partial credit, not just binary results of success/failure (for example, a learner may be awarded 3 out of 5 points for a particular exercise).

However, since our learner model focuses on around competence and confidence values and not directly around shape parameters, we need a translation

between the two. The interpretation currently in use in the system is that *competence* equals $E[X]$ and *confidence* is equal to $(1 - 4 \cdot Var[X])$. This is so that confidence increases as variance decreases as well as compensating for the low maximum variance of these distributions.

When an update with score s arrives, the shape parameters are recovered from the current competence (μ) and confidence (ϕ) values stored in the database. This is achieved via simple algebraic formulas that can be derived from Equations (1) and (2):

$$\alpha = \mu \cdot \left(\frac{4\mu(1-\mu)}{1-\phi} - 1 \right) \quad (3) \qquad \beta = (1-\mu) \cdot \left(\frac{4\mu(1-\mu)}{1-\phi} - 1 \right) \quad (4)$$

They can then be updated by simple addition ($\alpha' = \alpha + s$ respectively $\beta' = \beta + (1-s)$) and translated back into competence and confidence values (based on the expected value and variance of the distribution given by shape parameters α' and β') via equations Equations (1) and (2).

To gain a better intuition for the meaning of these constructions, the shape parameters can be understood as the amount of evidence *for* (α) and *against* (β) the hypothesis that the learner has mastered the associated concept in the associated cognitive dimension. Should α/β rise (say because the learner keeps answering questions correctly), the mean of the distribution rises and the variance falls, and vice versa. This corresponds to how one would expect it to work, because it also means that both the competence and the confidence values increase as well.

2.4 Learner Model Time Machine

To evaluate the learner models against real student data (exam results and quiz performances), a snapshot of the learner model of every student who participated in the exam in question ($n = 140$) at the day of the exam was recomputed from interaction logs.

This method (as opposed to simply grabbing a copy of the database at the time) allows to always test against the most current versions of the learner model algorithms while still only considering interaction data that was available at the given point in time. Should someone find a bug in the algorithm, an inconsistency in the data or another error that needs to be corrected *after* the time of the exam, this can compute what the learner model *would have been* if the mistake had been corrected, not just what it actually was at the time.

Since this allows us to basically jump back to any point in the history of a learner's interactions and compare it against others, the relevant software component is affectionately called `LMTM` (the `Learner Model Time Machine`).

This setup also makes it possible to easily test different versions of update algorithms against each other or empirically test different values for constants that are not derived from theory alone (e.g. the dampening constant γ from the intuitive model). It provides the foundation for all investigations in the following section.

3 Evaluation

To evaluate a learner model such as the above, we need to find out how useful it can be for current and future learning services. The principal metric of interest here is *accuracy*. How well do predictions of the learner model about how well a student is likely to do in a given task and their actual, real-life performance in that task align?

One of the most straightforward perspectives under which to test the accuracy of the model in a institutional education context is its predictiveness towards end-of-semester exam scores.

If a learner model such as the ones discussed above were shown to be reasonably accurate in predicting the exam score a student is likely to achieve, that would indicate sufficient quality to be used as a gauge for how well a student is doing with the material overall.

We will now conduct such an evaluation to test our learner models. This is important for ALeA, since most of the learning support services in the system use the learner model in some form.

For this evaluation, we are using real data that has been collected from ca. 150 students of an introductory artificial intelligence course at FAU [Ai1] over a time span of one winter semester. Students were introduced to ALeA at the start of the semester, including to its status as research software still in development, what sort of data it does and does not collect and the fact that the data it does collect from them might be used for scientific purposes. They were then invited to use ALeA for their studies (e.g. looking up course notes and definitions or answering practice questions), refining their learner models with each interaction.

An additional source of data available to us is the results of weekly quizzes held during the lecture. Once per week, students were encouraged to participate in an optional quiz with a duration of 10 minutes that asked a few relatively simple questions about the material discussed during the previous week's lecture. The questions were automatically scored and at the end of the semester, every student got assigned the average of their 10 best quizzes as their overall **quiz performance**. In the past, quizzes like these have been lauded as an excellent predictor for exam performance, both for theoretical – i.e. they encourage students to familiarize themselves with the material *during* the semester, not just shortly before the exam – as well as empirical – historically, quiz performance *does* correlate significantly with exam performance – reasons. Students were additionally motivated to use the quizzes by awarding up to 10% bonus on the final grade based on the quiz performance.

This gives us three points of comparison for the performance of any one of the learner modeling approaches: the other two models, simple noise (picking a random value between 0 and 1 as the value for any one entry of the "learner model", see Table 1 for the numerical comparisons of this approach against the others) and the quiz performances.

3.1 The Models in Comparison

For the comparative evaluation of the different models, we first determine the set of all concepts that are used in the exam problems. We then compute the average value from the appropriately timed snapshot of the learner model for exactly these concepts[1].

Figure 4 shows four subplots that compare the three learner modeling approaches with respect to relative predictiveness for the exam grade. The first three show the performance of one model plotted against the overall exam performance. The last subplot shows quiz performance plotted against exam performance as a baseline for comparisons. All subplots also show the average rooted mean square error for that series of data points.

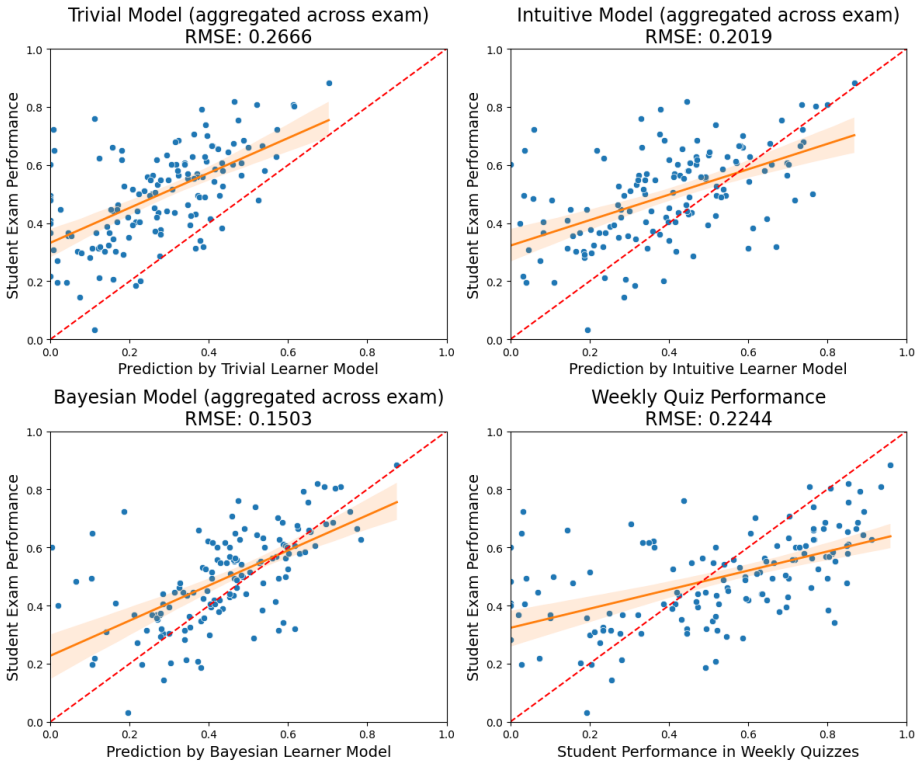

Fig. 4. Comparison of model performance vs exam performance for all three model types, as well as a similar plot of quiz performance vs exam performance. The line in red shows theoretically perfect predictions, the line in orange shows a simple linear regression.

[1] Concepts that do not appear in at least one learner model are skipped.

3.2 Different Aggregation Strategies

Another question this research hopes to make progress on is if the way you aggregate the available data makes a noticeable difference.

Figure 5 shows our results on this front. It includes the available results of the Bayesian model, aggregated three different ways.

The first way is selecting the set of relevant concepts for every single individual problem and then comparing the average model of those against the performance in that particular problem. The second way is similar, but selects the concepts of every thematic section (e.g. "Programming in Prolog", "Search Problems" or "First-Order Logic") in the exam and plots it against the performance in the section. A section here *can* consist of just one problem, but most of them include two or three instead. Lastly, the third way that was tested is including all concepts that appear anywhere in the exam, averaging the learner model over those and plotting it against the student's exam performance, as we have been doing thus far. Again, quiz performance plotted against exam performance is included as a reference point for comparison.

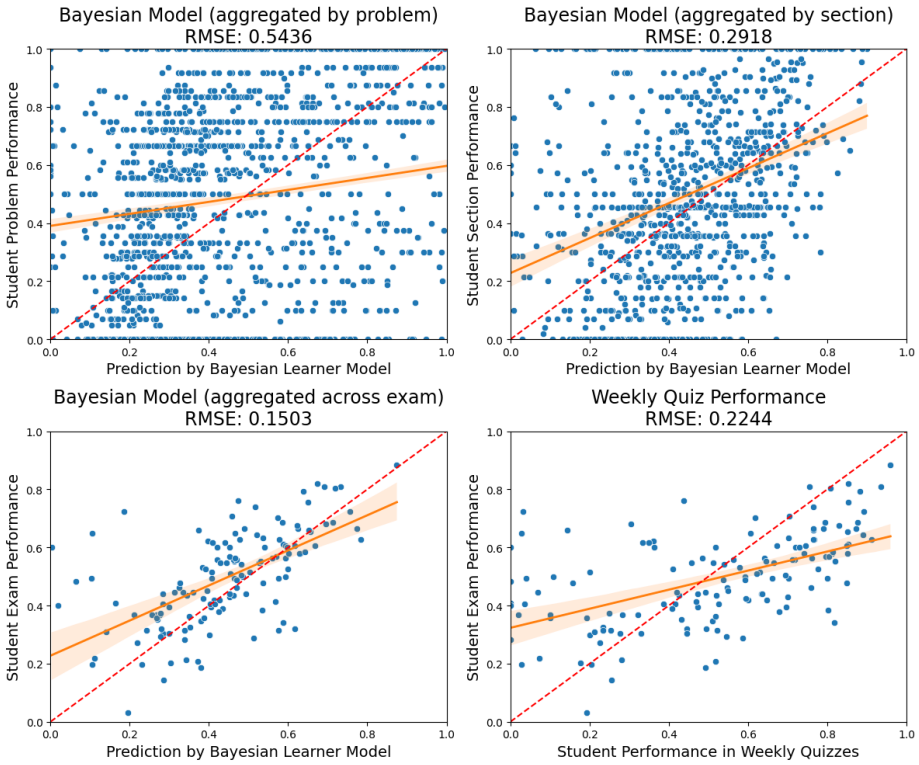

Fig. 5. Bayesian model performance vs. exam performance aggregated on a per-problem/section/exam basis; quiz performance vs. overall exam performance. Reference lines as above.

Interpreting the results we can see in Figure 5, it quickly becomes clear that although aggregating by section or even by individual problem gives a lot more data points, the average quality of the prediction goes down when deviating from the standard method of aggregating over the entire exam.

There are certainly a greater number of total data points available for the more fine-grained aggregation methods, however this is more than offset by the fact that across only a limited set of concepts, the average error in the prediction noticeably increases. Notably, aggregating by either problem or section leads to a greater root mean square error than the baseline comparison of quiz performance has.

It follows that, for the greatest accuracy, all analysis should be aggregated across the entire exam, giving an opportunity for the individual errors to middle out.

Note that the "layering", the visible accumulation of data points on the same y-value in the graphs for aggregation by problem and (to a lesser extent) by section, does not indicate a problem with the underlying data or with the evaluation algorithm. This effect is caused by the fact that when looking at individual problems, there are only so many points to be given out in the first place and hence there are only so many different values for how many points the student could reach. Some of them (e.g. half of the points) even appear particularly often, leading to the appearance of horizontal "lines" of data points.

3.3 Further Experiment: Additional Cognitive Dimensions

In the evaluation thus far, only the cognitive dimension of `Remember` was considered as it was empirically determined to give the most accurate results across different models.

Interestingly, this is generally *not* the dimension that's most frequently annotated. A total of 3441 available learning objects in the formulation model have any cognitive dimensions annotated whatsoever. Out of those 3441 learning object (all of them are problems), the frequency of annotated individual dimensions can be found below.

`Remember`	$\approx 20.052\%$ (n=690)
`Understand`	$\approx 43.534\%$ (n=1498)
`Apply`	$\approx 34.554\%$ (n=1189)
`Analyse`	$\approx 00.465\%$ (n=16)
`Evaluate`	$\approx 00.174\%$ (n=6)
`Create`	$\approx 01.221\%$ (n=42)

Many more learning objects than the aforementioned 3441 *use* concepts, of course. Used (referenced) concepts (not just annotated objectives) from the exam *are* included in the value for a student's learner model. However, they only change in the learner model if they are annotated as an objective of a learning object the student interacts with. Hence the prevalence of skipped concepts that are used in the exam but do not appear in any learner models.

Including other cognitive dimensions beyond `Remember` decreases the predictiveness significantly, as Table 1 shows.

	Trivial	Intuitive	Bayesian	Random	Cog. Dim.
Problem	0.44263	0.54327	0.54365	0.45798	R
Section	0.38103	0.32793	0.29181	0.41769	R
Exam	0.26657	0.20186	0.15035	0.30695	R
Problem	0.49418	0.50191	0.53602	0.45567	R/U/A
Section	0.44279	0.33668	0.29704	0.40331	R/U/A
Exam	0.36854	0.24318	0.17772	0.33641	R/U/A
Problem	0.53351	0.49011	0.46474	0.45347	all
Section	0.48353	0.41323	0.38091	0.41757	all
Exam	0.41846	0.33508	0.29438	0.33524	all

Table 1. Rooted mean square error values for all modes of aggregation and all learner models, including random values, considering cognitive dimensions.

3.4 Further Experiment: Grade Prediction

One of the most persistently suggested (by students and research staff alike) potential learning services is the pre-exam grade prediction. Specifically *not* in the sense of average accuracy overall that we have looked at above but the accuracy of the prediction which discrete grade step a student will achieve.

In German tertiary education, grades are usually given not as a precise decimal but rounded to the nearest of the following available discrete grade steps: $1.0, 1.3, 1.7, 2.0, \ldots 3.7, 4.0, 5.0$. The best possible grade one can hope to achieve is a grade of 1.0, whereas 4.0 is usually the worst grade that still passes. 5.0 is used to represent a student not passing the assessment at all.

Understandably, students are sometimes less interested in the precise accuracy of their learner model to the n^{th} significant digit, and more interested in whether or not the model can accurately predict which grade they are likely to get in the exam. Table 2 shows a comparison of the various different models and data sources and their relative predictive success for the exam in question. The first four rows describe the results for all 140 students who took part in the exam. The last three rows show the same data again, calculated only on the basis of those students that passed the exam.

In this, a "grade step difference" is the number of grade steps between the actual exam grade and the learner model prediction. Negative values indicate that the performance was better than the prediction and positive values indicate that the performance was worse than the prediction. Say a learner achieved a grade of 1.7 in the exam and the model predicted a grade of 2.3, this would correspond to a grade step difference of -2. An "exact" prediction is a prediction with a grade step difference of 0, a "near-miss" is a prediction with a grade step difference of either 1 or -1. The "Pass/Fail" row indicates for how many exams the model correctly predicted if the student would pass the exam or not. For the grade step difference, only grade step *differences* were considered. Hence an

	Trivial	Intuitive	Bayesian	Quiz	Random
Pass / Fail	$\frac{78}{140} \approx 55.7\%$	$\frac{92}{140} \approx 65.7\%$	$\frac{103}{140} \approx 73.6\%$	$\frac{98}{140} = 70,0\%$	$\frac{79}{140} \approx 56.4\%$
Exact	$\frac{70}{140} = 50\%$	$\frac{69}{140} \approx 49.3\%$	$\frac{73}{140} \approx 52.1\%$	$\frac{50}{140} \approx 35.7\%$	$\frac{47}{140} \approx 33.6\%$
Near Miss	$\frac{16}{140} \approx 11.4\%$	$\frac{22}{140} \approx 15.7\%$	$\frac{28}{140} = 20\%$	$\frac{22}{140} \approx 15.7\%$	$\frac{16}{140} \approx 11.4\%$
Avg. GSD	1.429	1.214	0.979	1.879	2.793
Exact	$\frac{0}{69} = 0\%$	$\frac{7}{69} \approx 10.1\%$	$\frac{10}{69} \approx 14.5\%$	$\frac{6}{69} \approx 8.7\%$	$\frac{4}{69} \approx 5.8\%$
Near Miss	$\frac{15}{69} \approx 21.7\%$	$\frac{18}{69} \approx 26.1\%$	$\frac{25}{69} \approx 36.2\%$	$\frac{12}{69} \approx 17.4\%$	$\frac{14}{69} \approx 20.3\%$
Avg. GSD	2.884	2.159	1.783	2.623	3.333

Table 2. Grade prediction results of several models, including quiz performance and random noise for reference. Here, GSD is short for "Grade Step Difference".

underestimation of two grade steps does not cancel out an overestimation of two grade steps somewhere else.

The results broadly line up with our expectations. The Bayesian model significantly outperforms all other sources of model data, including the intuitive model and the students' quiz performance, even getting to an average grade difference of below 1.0 in the case of all exam participants. With 72.1% and 50.7% at-most-near-miss predictions respectively, it is also performing well enough to be useful in practice on an absolute and not just a comparative level.

It should be noted that actually offering a "grade prediction" learning service to students would be inadvisable. Not only is even the best-performing model only able to correctly predict if a student will pass at all in three out of four cases[2], it would also in all likelihood disincentivize some students from further study if the model already predicts they will "pass anyway". Furthermore, in some cases (such as a student focusing their attention on a different course because of a strong prediction but then still failing the actual exam) such a learning service might even open the university up to legal trouble.

A better way to use the learner model in exam preparation would be to direct students towards topics their learner model is still weak in and away from topics they have already mastered to make the best possible use of their time.

4 Conclusion & Future Work

We have presented and evaluated the learner model provider of the ALeA system. The ALeA approach is special in that it is **semantic**, i.e. the competence model is a function from symbols to competence evaluations, and the update functions are based on declarative models of the meaning and didactic function of learning objects rather than their syntactic or surface forms. The learner

[2] This shortcoming could be mitigated by only offering the service to students if they have interacted enough with the system that we have a high confidence in their learner model.

model makes crucial use of this, and indeed, both successful models use the semantic dependency relation for updates.

In a way, the ALeA system takes the high road to learner modeling and adaptive learning support needing considerably more investment in establishment of a domain model and the semantic/didactic annotation of learning objects than other approaches. The advantage is that that learner models are conceptually relatively simple, reusable over a wide range of (annotated) learning objects, and – as we could show in the evaluation in this paper – surprisingly accurate.

It seems that the underlying stochastic approach of the Bayesian learner model works sufficiently well that we can concentrate on

1. using more structure from the domain or competency models. Additionally to the (vertical) domain dependency relation the intuitive and Bayesian models already use, we could use "horizontal" dependency between the cognitive dimensions of the domain model, e.g. usually – though there are counter-examples in CS – we can only apply a concept if we understand it.
2. using additional learner interaction data that can be interpreted into learner model updates. But if we want to glean learner model updates from e.g. the complete learner click-stream, we will have to interpret the learner intentions and interpret them as didactic events or intents, which in term can be transformed into meaningful learner model updates.
3. unlocking new information streams that can be interpreted didactically, e.g. video data captured from the camera of the device used by the learner, and interpreted for gaze information (eye-tracking) and facial micro-expressions (emotion analysis); see [KK24] for first results.

One thing to understand for the latter two is that to understand raw click or eye-tracking data – i.e. spatially encoded data of interactions with content on the screen, we have to tie it back to the underlying symbols, which the annotated learning objects in the ALeA context allow us to do, since we are generating the course materials from the semantically annotated media ourselves and can therefore instrument it to reveal semantic features.

Last but not least, we are on the verge of broadening our data from one semester (WS24/25) to three (the regular exams and retake exam for SS25 and WS25/26 have just been written and are being corrected as we write). This would give us a five-fold increase in data which we will include in the final version, which will also be accompanied by a published pseudonymized data set.

ALeA is open source, and the `LMP` described in this paper is at [Bet25], where the evaluation scripts can also be found.

References

[AAH14] Mohammad Alshammari, Rachid Anane, and Robert J. Hendley. "Adaptivity in E-Learning Systems". In: *2014 Eighth International Conference on Complex, Intelligent and Software Intensive Systems*. 2014, pp. 79–86. DOI: 10.1109/CISIS.2014.12.

[AB14] Mohammad Alotaibi and Susan Bull. "Discussion around Individual Open Learner Models: Understanding or Copying?" In: *2014 IEEE 14th International Conference on Advanced Learning Technologies*. 2014, pp. 82–83. DOI: 10.1109/ICALT.2014.34.

[Ai1] *AI-1 Course Home — ALeA*. URL: https://alea.education/FAU/ai-1/WS25-26 (visited on 04/08/2026).

[ALeA] *ALeA: Adaptive Learning Assistant*. URL: http://alea.education (visited on 07/16/2024).

[Ber+23] Marc Berges et al. "Learning Support Systems based on Mathematical Knowledge Managment". In: *Intelligent Computer Mathematics (CICM) 2023*. Ed. by Catherine Dubois and Manfred Kerber. Vol. 14101. LNAI. Springer, 2023. DOI: 978-3-031-42753-4. URL: https://url.mathhub.info/CICM23ALEA.

[Bet25] Jonas Lambda Totoro Betzendahl. *ALeA Learner Modelling Software Repository*. Available at https://gitos.rrze.fau.de/voll-ki/fau/system/alea-lm. 2025. URL: https://gitos.rrze.fau.de/voll-ki/fau/system/alea-lm.

[Bij25] Hildo Bijl. *Tracking student skills real-time through a continuous-variable dynamic Bayesian network*. 2025. arXiv: 2501.10050 [stat.ML]. URL: https://arxiv.org/abs/2501.10050.

[Bis06] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer New York, 2006.

[Blo56] B. S. Bloom, ed. *Taxonomy of educational objectives: The classification of educational goals: Handbook I, cognitive domain*. New York, Toronto: Longmans, Green, 1956.

[Bö+25] Felix Böck et al. "Learner models: design, components, structure, and modelling". In: *User Modeling and User-Adapted Interaction* 35 (2025). DOI: 10.1007/s11257-025-09434-4. URL: https://doi.org/10.1007/s11257-025-09434-4.

[Ful+07] Ursula Fuller et al. "Developing a computer science-specific learning taxonomy". In: *SIGCSE Bulletin* 39.4 (2007), pp. 152–170.

[Gun19] Gregory Gundersen. *Conjugacy in Bayesian Inference*. 2019. URL: https://gregorygundersen.com/blog/2019/03/16/conjugacy/.

[KK24] Andrea Kohlhase and Michael Kohlhase. "Towards Automated Competency Estimation for Math Education – An Eye Tracking and Emotion Analysis Study". In: *MathUI 2024: The 15th Workshop on Mathematical User Interfaces*. Ed. by Kazuhisa Nakasho and Jan Frederik Schaefer. 2024. URL: https://kwarc.info/kohlhase/papers/mathui24-dbloom.pdf.

[Loh+23] Dominic Lohr et al. "The Y-Model – Formalization of Computer-Science Tasks in the Context of Adaptive Learning Systems". In: *2023 IEEE German Education Conference (GeCon)*. IEEE, 2023. URL: https://url.mathhub.info/23GeCon.

[PMV22] Héctor Miguel Figueroa Pucheta, Viviana Yarel Rosales Morales, and Yesenia Hernández Velázquez. "User Modeling for Collaborative Adaptative Mobile Educational Applications: A Systematic Review of the Literature". In: *2022 IEEE Mexican International Conference on Computer Science (ENC)*. 2022, pp. 1–7. DOI: 10.1109/ENC56672.2022.9882919.