

Rubric-based Learner Modelling via Noisy Gates Bayesian Networks for Computational Thinking Skills Assessment

Giorgia Adorni, Francesca Mangili, Alberto Piatti, Claudio Bonesana, and Alessandro Antonucci

Abstract—In modern and personalised education, there is a growing interest in developing learners’ competencies and accurately assessing them. In a previous work, we proposed a procedure for deriving a learner model for automatic skill assessment from a task-specific competence rubric, thus simplifying the implementation of automated assessment tools. The previous approach, however, suffered two main limitations: (i) the ordering between competencies defined by the assessment rubric was only indirectly modelled; (ii) supplementary skills, not under assessment but necessary for accomplishing the task, were not included in the model. In this work, we address issue (i) by introducing dummy observed nodes, strictly enforcing the skills ordering without changing the network’s structure. In contrast, for point (ii), we design a network with two layers of gates, one performing disjunctive operations by noisy-OR gates and the other conjunctive operations through logical ANDs. Such changes improve the model outcomes’ coherence and the modelling tool’s flexibility without compromising the model’s compact parametrisation, interpretability and simple experts’ elicitation. We used this approach to develop a learner model for Computational Thinking (CT) skills assessment. The CT-cube skills assessment framework and the Cross Array Task (CAT) are used to exemplify it and demonstrate its feasibility.

Index Terms—Learner modelling; Bayesian networks with noisy gates; Assessment rubrics; Computational thinking skills.

I. INTRODUCTION

INTELLIGENT Tutoring Systems (ITSs) are technological devices that support learning without the mediation of a teacher. They interact directly with the user, providing hints and suggestions that can only be effective if calibrated to the actual user’s competence level. ITSs collect data on a learner’s performance while accomplishing a task and use that data to develop a competence profile based on a predefined model of the learner’s knowledge and behaviour. This profile helps determine the most appropriate intervention. The new

knowledge collected along with the learning activity continuously updates the competence profile, making the interventions more focused. Therefore, the learner model is one of the main factors that contribute to the success of an AI-based educational tool.

A learner model describes mathematically the learner’s competencies, represented by a set of hidden variables, and their relations with the observable actions performed while solving the task. Such competencies combine knowledge, skills, and attitudes expressed in a specific context. Teachers can evaluate student competencies in realistic scenarios explicitly designed for this purpose and then compare their performance with a model of competence specified through an assessment rubric [1]. A rubric for assessing a student’s performance consists of a list of competence components to be evaluated, a qualitative description of possible observable behaviours corresponding to different levels of such components, and a set of criteria for assessing the level of each component. A rubric, therefore, describes the relationship between competencies and observable behaviours of the learner that need to be formally codified.

Several sources of uncertainty and variability may affect the relationship between the non-observable competencies and the corresponding observable actions. Therefore, a deterministic relationship is not capable of accurately modelling it. Instead, a more appropriate approach would be to use probabilistic reasoning to translate qualitative assessment rubrics into a quantitative, standardised, coherent measure of student proficiency. In the literature, *Bayesian Knowledge Tracing* (BKT) [2], *Item Response Theory* (IRT) [3], and *Bayesian Networks* (BNs) [4] are all popular probabilistic approaches to learner knowledge modelling. BNs are a robust framework for modelling dependencies between skills and students’ behaviours in complex tasks. In addition, the graphical nature of the models makes them easily understandable by domain experts. Therefore, experts can easily use them in eliciting the student model [5]. Desmarais [6] reviewed all the most successful ITS experiences since Bloom’s keynote paper and recognised and presented BNs as the most general approach to modelling learner skills. [7] conducted a systematic review of 53 papers about ITS applications from 2007 to 2017. The review explored the characteristics, applications, and evaluation methods of ITSs and found that a significant proportion of the reviewed papers used BN techniques. More recent works using

Manuscript received December 19, 2022; revised February 12, 2023. Date of publication March 6, 2023. Date of current version March 6, 2023.

The paper was presented in part at the International Conference on Software, Telecommunications and Computer Networks (SoftCOM) 2022.

G. Adorni, F. Mangili, C. Bonesana and A. Antonucci are with the Istituto Dalle Molle di Studi sull’Intelligenza Artificiale (IDSIA) USI - SUPSI, Lugano, Switzerland (e-mails: {giorgia.adorni, francesca.mangili, claudio.bonesana, alessandro.antonucci}@idsia.ch). A. Piatti is with the Department of Education and Learning (DFA), SUPSI, Lugano, Switzerland (e-mail: alberto.piatti@supsi.ch).

This research was funded by the Swiss National Science Foundation (SNSF) under the National Research Program 77 (NRP-77) Digital Transformation (project number 407740_187246).

Digital Object Identifier (DOI): 10.24138/jcomss-2022-0169

BNs to model the learner knowledge in the context of ITSs include [8], who developed an ITS to help students acquire problem-solving skills in computer programming, and [9], who developed an automatic assessment method for students' engineering design performance using a BN model. Other recent works, such as [10], [11], support the construction of ITSs based on BNs and further highlight the ever-present interest in BN techniques for ITSs. Building on these results, we focused our method on BN-based learner modelling.

Not all BNs are easy to design, and a deep understanding of BN theory is required. Although BN arcs can be interpreted as a causal model, their definition by experts is not always trivial because of the complexity of the causal relationships involved and the presence of hidden causes. In addition, a significant effort may be necessary to obtain the network structure and parameters through expert knowledge or the availability of an extensive dataset to learn them directly.

Decomposing domain knowledge into individual basic component skills is the choice typically adopted to simplify the model elicitation process. However, as pointed out in [12], "complex skill mastery requires not only the acquisition of individual basic component skills but also practice in integrating such component skills with one another". Further complications can also arise even when the structure of the learning model can be precisely defined. In some cases, eliciting and learning BN parameters can quickly make the computation of inferences unmanageable. The number of parameters and the problem complexity can rapidly increase with the network's number of arcs.

These issues can discourage ITS practitioners from using BN-based learner modelling in their applications when many skills are involved in the learner's actions. A solution to reduce the number of parameters in a BN-based learner model was proposed in our previous paper [13]. We exploit *noisy-OR gates* [14] to reduce the number of parameters to elicit from exponential in the number of parent skills for each observable action to linear. Similar advantages also concern the inference. In [15], we adopted a solution to set up a general approach for translating assessment rubrics into interpretable BN-based learner models with a complexity compatible with real-time assessment.

Learner models based on assessment rubrics are more accessible to teachers, who are typically more familiar with them than probabilistic graphical models. Moreover, rubrics focus on the learners' complex behaviours in specific contexts. Therefore, although we do not explicitly model skill interactions, they can be captured in the hierarchy of complex behaviours identified by the rubric. However, in our previous work [15], this hierarchy of competencies was only indirectly modelled. This produced assessments assigning larger probabilities to higher-level competence, in contrast with the assumption that when a competence of a certain level is possessed, all lower-level competencies are also owned.

In the model presented here, this counter-intuitive behaviour has been eliminated by imposing the constraints codified by the rubric through observed auxiliary nodes that do not modify the network structure and retain the previous model's relative simplicity. Moreover, considering that assessment rubrics are

usually limited to the competence components under assessment, hereafter also called *target skills*, the modelling approach in [15] may have led to oversimplified learner models, unable to grasp the actual causes of a failure which is not always due to the absence of the target skills. It may, for instance, follow from deficiencies in other skills necessary for the specific task, hereafter referred to as *supplementary skills*.

Therefore, this work extends our previous model [15] by adding the possibility of modelling a set of supplementary skills necessary, in conjunction with the skills under examination, to succeed in the assigned tasks. To this goal, it is essential to extend the noisy-OR approach in [13] to combine disjunctive and conjunctive relations between skills since. At the same time, the behaviours in the assessment rubric are mutually exclusive (OR), and supplementary and target skills must be expressed jointly (AND).

To illustrate this approach, we focus on the activity proposed in [16] for the standardised assessment of algorithmic skills along the entire K-12 school path. We compare four learner models based on different assumptions and sets of expert-elicited parameters and apply them to the dataset collected in [16]. Overall, we obtain a general and compact approach to implementing a learner model given a set of competencies of interest and the corresponding assessment rubric. The resulting model has a simple structure and interpretable parameters, allowing for fast inferences with a reasonable effort for their elicitation by experts.

This article is organised as follows:

- Section II-A provides some background about learner modelling based on Bayesian Networks, and noisy gates.
- Section II-B introduces task-specific assessment rubrics.
- Section II-C illustrates how to model a generic assessment rubric by Bayesian Networks.
- Section III presents the case study and, in particular, the specific assessment rubric developed for the Cross Array Task (CAT) from [16] as well as the procedure for translating it into a learner model.
- Section IV discusses the model outcomes based on the dataset of 109 pupils collected in [16].
- Section V summarises the work's findings and contribution.

II. METHOD

A. BN-based Learner Models

The structure of a Bayesian Network (BN) over a set of variables is described by a directed acyclic graph \mathcal{G} whose nodes are in one-to-one correspondence with the variables in the set. We call parents of a variable X , according to \mathcal{G} , all the variables connected directly with X with an arc pointing to it. Learner models usually include a set of n latent (i.e., hidden) variables $\mathbf{X} := (X_1, \dots, X_n)$, henceforward referred to as *skill nodes*, describing the competence profile of the learner and some m manifest variables $\mathbf{Y} := (Y_1, \dots, Y_m)$, hereafter called *answer nodes*, describing the observable actions implemented by the learner to answer each specific task. Arcs go from skill nodes to answer nodes modelling how the presence or absence of a specific competence directly affects

the learner's behaviour in a task requiring such competence. Here, we consider only binary skill nodes, where value one, or *true* state, indicates that the pupil possesses the skill, and binary answers nodes, denoting a correct answer or behaviour shown in solving the task.

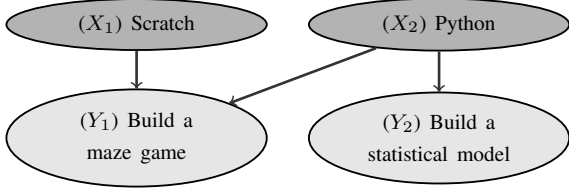


Fig. 1. Example of BN-based learner model.

The relations of a BN-based learner model can be graphically depicted as in the example of Fig. 1. The answer nodes describe whether the learner has been able or not to program, for example, a maze game (Y_1) or a statistical model (Y_2). The skill nodes represent the ability to build this program using a block-based programming language such as Scratch (X_1) or a text-based programming language such as Python (X_2). The second skill can be applied to answer both questions, and therefore X_2 is a parent node for both answer nodes Y_1 and Y_2 . Instead, the first skill can be used to answer just the first question, and therefore there is no direct arc from X_1 to Y_2 .

Once the graph \mathcal{G} structuring the BN is established, the definition of the BN over the $n + m$ variables of the network $\mathbf{V} := (V_1, V_2, \dots, V_{n+m})$, including both skills (\mathbf{X}) and answers (\mathbf{Y}), consists in a collection of Conditional Probability Tables (CPTs) giving the probabilities $P(Y_i = 1 | \text{Pa}(Y_i))$ that Y_i takes value one given all possible joint states of its parent nodes $\text{Pa}(Y_i)$. Let \mathbf{V} take values in $\Omega_{\mathbf{V}}$, the independence relations imposed from \mathcal{G} by the *Markov condition*, i.e., the fact that each node is assumed to be independent of its non-descendants non-parents given its parents, induce a joint probability mass function over the BN variables that factorises as follows [4]:

$$P(\mathbf{V} = \mathbf{v}) = \prod_{v \in \mathbf{v}} P(v | \text{pa}(V)), \quad (1)$$

where $\mathbf{v} = (v_1, v_2, \dots, v_{n+m})$ represents a given joint state of the variables in \mathbf{V} . BN inference consists of the computation of queries based on Eq. (1). In particular, we are interested in *updating* tasks consisting in the computation of the marginal posterior probability mass function for a single skill node $X_q \in \mathbf{X}$ given the observed state \mathbf{y}_E of the answer nodes $\mathbf{Y}_E \subseteq \mathbf{Y}$:

$$P(x_q | \mathbf{y}_E) = \frac{\sum_{\mathbf{v} \in \Omega_{\mathbf{V}} | (x_q, \mathbf{y}_E)} \prod_{v \in \mathbf{v}} P(v | \text{pa}(V))}{\sum_{\mathbf{v} \in \Omega_{\mathbf{V}} | \mathbf{y}_E} \prod_{v \in \mathbf{v}} P(v | \text{pa}(V))}, \quad (2)$$

where $\Omega_{\mathbf{V} | \mathbf{v}'} := \{\mathbf{v} : v_i = v'_i \forall v'_i \in \mathbf{v}'\}$.

According to the above model, multiple parent skills may be relevant to the same answer. The paper in [13] discusses how this can lead to a critical complexity in elicitation and inference and demonstrates how using noisy gates can avoid these issues. In this work, we exploit the disjunctive noisy-OR gates, which shape interchangeable skills and are suitable for modelling assessment rubrics.

1) *Noisy-OR Gates*: This section briefly introduces noisy-OR gates and their use in learner modelling. We refer to [13] and [15] for a more detailed discussion. A typical representation of the noisy-OR network structure, introducing n auxiliary variables (also called inhibitor nodes), is shown in Fig. 2.

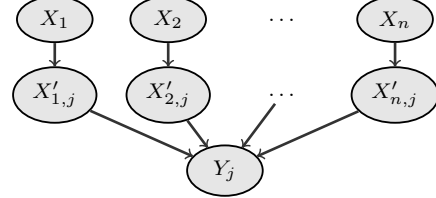


Fig. 2. A noisy gate explicit formulation (adapted from [13]).

To reduce the number of parameters, the noisy-OR defines the state of Y_j as the logical disjunction (OR) of the auxiliary parent nodes, removing the need to specify the CPT of the answer node given the state of its parent nodes. Furthermore, the noisy-OR structure sets the input variable X_i as the unique parent of $X'_{i,j}$ and constraints $X'_{i,j}$ to be zero with probability one when $X_i = 0$. The relationship between skill and answers would be purely logical-deterministic were it not for the noise introduced by the so-called inhibition parameters $\lambda_{i,j} = P(X'_{i,j} = 0 | X_i = 1)$, representing the probability of not expressing skill i in task j . Auxiliary variables can be interpreted as *inhibitors* of the corresponding skills, as they can be in the false state, e.g., $X'_{i,j} = 0$ (with probability $\lambda_{i,j}$) even when the corresponding skill node X_i is in the true state, indicating that, although possessed by the learner, the skill could not be expressed in task Y_j . By defining the probability of a failure in expressing a possessed skill in the specific task j , the inhibition parameter $\lambda_{i,j}$ provides a measure of the task difficulty. If a pair skill-answer has a large inhibition, the state of the answer node tells, in general, little about the state of the skill node, the extreme case of $\lambda_{i,j} = 1$ corresponding to a missing arc in the BN graph between skill i and answer j .

The noisy-OR network induces the following CPT between the n parent skill nodes $\mathbf{X} = (X_1, \dots, X_n)$ and the observable answer node Y_j [14]:

$$P(Y_j = 0 | \mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)) = \prod_{i=1}^n (\mathbb{I}_{\mathbf{x}_i=0} + \lambda_i \mathbb{I}_{\mathbf{x}_i=1}), \quad (3)$$

where \mathbb{I}_A is the indicator function returning one if A is true and zero otherwise. The second term $\lambda_i \mathbb{I}_{\mathbf{x}_i=1}$ represents the noise as it introduces the possibility that a skill X_i that the student possesses is not expressed in task Y_j (this phenomenon is also called *slip* elsewhere in this work). The value of λ_i implying the biggest uncertainty associated with the task-skill pair (Y_j, X_i) is 0.5, whereas the value $\lambda_i = 0$ models the certainty that skill X_i , whenever present, will be expressed in solving task Y_i and, vice versa, $\lambda_i = 1$ model the fact that X_i cannot be expressed in task Y_j .

In [15], a leak node was added to the model to represent the possibility of a random guess, i.e., a correct answer or a behaviour given without mastering any required competencies. The leak is a Boolean variable playing the role of an additional

skill node, X_{leak} , which is set in the observed state $X_{\text{leak}} = 1$ and added to the parents of all answer nodes for which random guessing is possible. The chances of guessing answer Y_j at random are given by parameter $1 - \lambda_{\text{leak},j}$.

To apply the above model, the domain expert (e.g., the teacher) should first list the parentless skill nodes (including, eventually, the leak) X_1, \dots, X_n , the childless answer nodes Y_1, \dots, Y_m and connect by an arc the skills to all answer nodes in which they can be used. Then, the instructor should quantify for each pair of skill-answer nodes, X_i and Y_j , connected by an arc, the value of the inhibition $\lambda_{i,j}$. This results in a total of at most $n \cdot m$ parameters to be elicited. Finally, the expert should state each skill's prior probabilities π_i .

2) *Comparison with BKT*: While the BKT, in its standard implementation, traces the evolution of a single skill over time, our approach focuses on fine-grained skills modelling at the specific moment the assessment is performed. However, a parallel can be drawn between the two. BKT models student knowledge at time t as the (binary) latent variable $X(t)$ of a hidden Markov model [2]. Learning is modelled as the transition of $X(t)$ from state zero (lack of knowledge) to state one (knowledge acquired). The model defines four parameters: (i) the *initial* probability, i.e., the probability that the knowledge has been already acquired at the beginning of the activity; (ii) the *learning* probability, that is, the probability of acquiring the probability between t and $t + 1$; (iii) the *slip* probability of making a mistake when the knowledge is acquired; (iv) the *guess* probability of doing right in the lack of knowledge.

In our model, the probability of the *slip* may vary depending on the pair skill i and task j and is represented by the inhibition $\lambda_{i,j}$. The *guess* probability depends on the task and is equal to $1 - \lambda_{\text{leak},j}$. The *initial* probability of a skill X_i is defined by its prior probability π_i . Notice, however, that since our approach, differently from BKT, does not model the learning process, the concept of initial probability here is meant to describe our initial knowledge of the learner competence profile rather than the probability that the skill is initially acquired. For the same reason, no *learning* probability is defined in our model.

B. Assessment Rubrics

There are several possible approaches to identifying the knowledge components to be included in a learner model. In this article, we follow the one introduced in [15] and start from a rubric defined for assessing a given competence through a specific task or family of similar tasks [17], [18].

A task-specific assessment rubric consists of a two-entry table where each row corresponds to a component of the given competence, described in the light of the given task. In contrast, each column corresponds to a competence level in ascending order of proficiency. For each combination of component and level, the rubric provides a qualitative description of the behaviour expected from a person with the given level in the given component. Identifying a person's competence level consists in matching the learner's behaviours while solving a given task with those described in the assessment rubric.

For instance, Table I shows the task-specific assessment rubric for an example focused on assessing the student's ability

TABLE I
EXAMPLE OF A TASK-SPECIFIC ASSESSMENT RUBRIC WITH A SINGLE COMPETENCE COMPONENT, THE ABILITY TO DESIGN ALGORITHM CONTAINING LOOPS, AND TWO COMPETENCE LEVELS.

	X_1	X_2
Loops	Develop an iterative algorithm using a block-based programming language	Develop an iterative algorithm using a text-based programming language

to use iterative instructions in algorithms. This competence has two levels depending on the tools used by the learner: a visual programming language (X_1) or a textual programming language (X_2). By checking how the learner produced the algorithm, the teacher can see whether he applied any of the methods in the rubrics and assign him the corresponding competence level.

Generally speaking, assessment rubrics define an ordering between competence levels and sometimes between competence components, as for the case study about computational thinking introduced in [15] and discussed in Section III. Here, a competence level or component is considered higher than another if the former implies the latter, meaning that a learner with the higher competence can also perform all the tasks that require the lower. In practice, the competence level matching the student's behaviours for a given component does not always correspond to the actual learner's state of knowledge. It is also possible that the person possesses a higher level but is underperforming.

In the case of a task composed of similar sub-tasks, i.e., tasks sharing the same assessment rubric, it is, therefore, possible to observe behaviours matching different competence levels in the various sub-tasks. In the following subsection, we illustrate how this uncertainty can be considered and how an overall assessment based on a full battery of tasks can be produced by modelling the learner competence profile with the BN-based approach described in Section II-A.

C. Modelling Assessment Rubrics by Bayesian Networks

Considering a task-specific assessment rubric, as defined in Section II-B, it is possible to derive a learner model, as presented in Section II-A, hereafter referred to as *baseline model*. For each cell (c, r) of an assessment rubric with R rows and C columns, we introduce a latent binary competence variable X_{rc} taking value one for a learner mastering the corresponding competence level and zero otherwise. Moreover, for each task t , in a battery of T similar tasks, and each competence variable X_{rc} , we define an observable (manifest) binary variable Y_{rc}^t taking value one if the behaviour described in the assessment rubric's cell (r, c) was applied successfully by the learner in solving task t and zero if he failed using it.

In this work, we improve the baseline model in two ways. Firstly, we explicitly impose the ordering of competence levels encoded by the rubric. Secondly, we include in the model task-specific supplementary skills which can be combined with each other and with the competencies of the rubric through arbitrary logic functions.

1) *Ordering of competences:* In the baseline model of Section II-A, it was indirectly accounted for the partial ordering between variables by setting as parents of answer node Y_{rc}^t the skill node X_{rc} and all skill nodes corresponding to higher competence levels. The network was quantified through noisy-OR relations, as described in Section II-A1. This structure assumes that an observed behaviour can be explained as the student mastering the corresponding competence level or a higher one if he is underperforming, thus not exploiting his full potential, but cannot be achieved through a lower level.

As mentioned above, we interpret the (partial) ordering between competencies defined by the assessment rubric as implication constraints, meaning that possessing a particular skill X_i implies that the learner possesses also his inferior competencies. While exploited to design the network structure, this hierarchy of competencies is not strictly imposed by the above baseline model, giving rise to posterior inferences that are usually inconsistent.

To solve this issue, we enrich the model by adding an auxiliary variable D_{ik} for each relation $X_i \implies X_k$ defined by the rubric. A constraint node D_{ik} is always in the observed state one and has X_i and X_k as parent nodes. The desired implication constraint is then implemented by choosing a CPT for D_{ik} such that $P(D_{ik} = 1|X_i = 1, X_k = 0) = 0$. The addition to the network of each observed node D_{ik} changes the prior probabilities of X_i and X_k , initially set to π_i and π_k . Let

$$\begin{aligned} p_{00} &= P(D_{ik} = 1|X_i = 0, X_k = 0) \\ p_{01} &= P(D_{ik} = 1|X_i = 0, X_k = 1) \\ p_{11} &= P(D_{ik} = 1|X_i = 1, X_k = 1), \end{aligned} \quad (4)$$

be the non-null parameters in the CPT of D_{ik} . After updating with the evidence $D_{ik} = 1$, one has

$$\begin{aligned} P(X_i = 1|D_{ik} = 1) &= \frac{p_{11}\pi_j\pi_k}{K}, \\ P(X_k = 1|D_{ik} = 1) &= \frac{p_{11} + p_{01}\pi_j\pi_k}{K}, \end{aligned} \quad (5)$$

with $K = p_{11}\pi_j\pi_k + p_{01}(1 - \pi_j)\pi_k + p_{00}(1 - \pi_j)(1 - \pi_k)$.

In this work, we simply assume $p_{00} = p_{01} = p_{11}$ and adopt uniform prior probabilities $\pi_i = \pi_k = 0.5$. Applying them to Eq.(5) give $P(X_i = 1) = 1/3$ and $P(X_k = 1) = 2/3$. This result follows from the fact that skill X_i can only be possessed jointly with X_k , whereas X_k can also be owned when $X_i = 0$.

Under the assumption $p_{00} = p_{01} = p_{11} = p_*$, the prior over the superior skill X_i can be interpreted as the conditional probability of having it given that the learner possesses the inferior skill X_k since

$$P(X_i = 1|X_k = 1, D_{ik} = 1) = \frac{\pi_i\pi_j p_*}{\pi_i\pi_j p_* + (1 - \pi_i)\pi_j p_*} = \pi_i \quad (6)$$

2) *Supplementary Skills:* While the assessment rubric details the components of the competence of interest and their interactions with the specific task and available tools, it does not necessarily include all the skills required to solve the task successfully.

For instance, considering the assessment rubric proposed in Table I, to develop an iterative algorithm with a text-based programming language successfully, the learner might also need knowledge about the different types of statements, e.g., while, repeat, for, do until and so on. Ignoring such supplementary skills might be misleading in an automatic assessment system, as failures due to the lack of one of them would not be recognised as such and, eventually, be attributed to the absence of the competence components under assessment. Therefore, if not adequately modelled, the lack of unmodelled supplementary skills would translate into an unfairly negative evaluation of the competencies of interest.

To produce fairer assessments, we extend the model by an additional layer of auxiliary nodes combined with a logic function to allow for the inclusion of a suitable set of supplementary skills.

Fig. 3 shows an example of the structure of the extended network. Supplementary skills are described by additional skill nodes S_1, \dots, S_m , which are grouped into sets of interchangeable skills (in the case of the example we have just one set). Each of these groups is connected through a noisy-OR to a node in the layer of auxiliary latent nodes, hereafter referred to as group nodes G_1, \dots, G_l , representing the success or failure in applying the type of competence described by each group to the specific task Y . Finally, the group nodes are connected to the answer node through a logic AND or any other logic function suitable for the particular task.

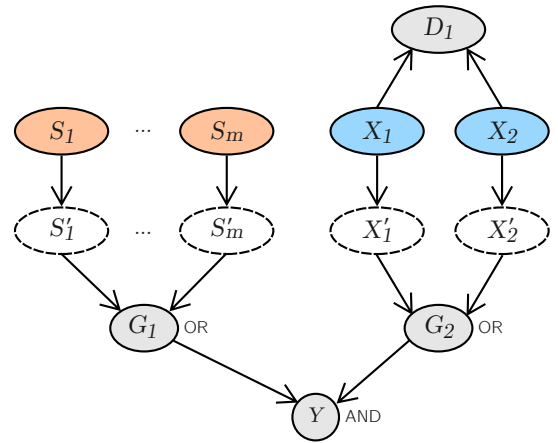


Fig. 3. Example of a BN network modelling a task-specific assessment rubric with two cells, represented by skills X_1 and X_2 (on the right), m supplementary skills grouped in a single set (on the left), and the constraint $X_2 \implies X_1$, represented by the auxiliary variable D_1 (on the top right).

When supplementary skills can be directly assessed through observing specific learner behaviours or by purposed questions, additional answer nodes, children only of the relevant supplementary skills, can be added to the network.

III. A CASE STUDY ON K-12 COMPUTATIONAL THINKING SKILLS

In this section, we use the case study about Computational Thinking (CT) skill assessment already introduced in [15] to illustrate the proposed approach. However, our methodology

can be applied analogously to each task for which an assessment rubric can be defined.

CT assessment is an important field of research [19] due to its relevance in evaluating the effectiveness of CT teaching and learning activities on an individual or class level and, on a larger scale, the impact of curricular and educational system policies on the development of CT skills of a given population.

A. The Cross Array Task

The Cross Array Task (CAT), proposed in [16], is an unplugged activity designed to assess the development of algorithmic skills, i.e., the ability to describe a complex procedure through a set of simpler instructions, in pupils aged from 3 to 16 years.

The authors of [16] carried out an experimental study from March to April 2021, collecting data from 109 students (51 girls and 58 boys) in eight classes from three public schools in Southern Switzerland.

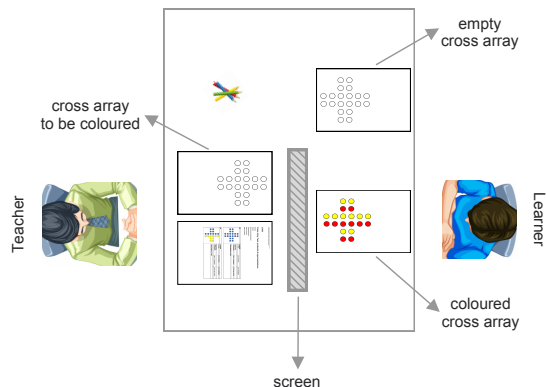


Fig. 4. CAT experimental settings (adapted from [16]).

The data collection setting is illustrated in Fig. 4. In the activity, each student was provided with 12 coloured cross arrays, visible in the right part of Fig. 4 and at the top of Fig. 5, characterised by different levels of complexity and based on different types of regularities, and requested to give the teacher instructions, either by voice (symbolic artefact) or with the help of an empty cross array schema (embodied artefact), to reproduce the same colouring patterns on a blank cross array. The pupil and the teacher are initially separated by a screen that does not allow the student to see what the other is doing. Upon request by the pupil, this barrier is removed to allow him/her to see how his commands are implemented (visual feedback).

B. Modelling the CAT Assessment Rubric

The assessment rubric for this case study, shown in Table II, is the same as presented in our previous work [15].

The instruction sequences built by the pupils, called algorithms, are ranked into three categories corresponding to the assessment rubric's competence components (rows). Each row represents the ability of the pupil to solve a CAT schema using, respectively, *zero-dimensional (0D)* algorithms – where the dots of the schema are described point by point –, *one-dimensional (1D)* algorithms – where structures, such as

rows, diagonal, squares etc. are also used to illustrate the coloured pattern –, and *two-dimensional (2D)* algorithms – where repetitions and loops on dots or structures are also used to describe the schema.

The degree of autonomy of the pupil and the tools used to accomplish the task have been hierarchically ordered and determine the competence levels in the columns of the rubric. Specifically, from the highest (right) to lowest (left), such levels correspond to the ability of the pupil to solve a CAT schema using: *voice (V)* – the pupil can communicate all the necessary instructions using only the voice –, *voice & schema (VS)* – the pupil, in combination with the voice, uses an empty cross array schema to illustrate his instructions through gestures, e.g., by pointing with fingers the dots to be coloured –, *voice, schema & feedback (VSF)* – the student also asks to remove the screen to have visual feedback of how the teacher is colouring the schema.

As introduced in Section II-C, the columns of an assessment rubric provide the competence levels in increasing order from left to right. Sometimes, as in this case study, this is true also for the rows, where competence components are ordered from the lower (0D at the top) to the highest (2D at the bottom). This follows from the assumption that mastering algorithms of higher complexity imply also mastering simpler ones. The same is valid for communication tools.

Summing up, we can conclude that a competence level X_{rc} is higher than $X_{r'c'}$ whenever $c > c'$ and $r \geq r'$, or $c = c'$ and $r > r'$. When, instead, $c > c'$ but $r < r'$, neither skill can be said to dominate the other.

From the CAT assessment rubric in Table II we define nine target skills to be assessed, representing the ability to develop an algorithm using elementary operations communicated either by a symbolic language (X_{13} – 0D-V), an embodied language (X_{12} – 0D-VS) or supported by visual feedback (X_{11} – 0D-VSF); elementary operations and structures communicated either by a symbolic language (X_{23} – 1D-V), an embodied language (X_{22} – 1D-V) or supported by visual feedback (X_{21} – 1D-VSF); elementary operations, structures and repetitions communicated either by a symbolic language (X_{33} – 2D-V), an embodied language (X_{32} – 2D-VS), or supported by visual feedback (X_{31} – 2D-VSF).

TABLE II
DEFINITION OF THE CAT ASSESSMENT RUBRIC.

		Competence levels		
		VSF $c = 1$	VS $c = 2$	V $c = 3$
Competence components	0D ($r = 1$)	X_{11}	X_{12}	X_{13}
	1D ($r = 2$)	X_{21}	X_{22}	X_{23}
	2D ($r = 3$)	X_{31}	X_{32}	X_{33}

Accordingly, with the method described in Section II-C, a latent skill node X_{rc} is included in the BN learner model for each of the nine target skills of the rubric. The hierarchy of competencies is then modelled by nine latent binary variables $D_{rc,r'c'}$, as described in Section II-B, encoding the implication $X_{rc} \implies X_{r'c'}$ for each pair of consecutive skills in the

hierarchy, i.e., such that $(r = r' + 1) \wedge (c = c')$ or $(r = r') \wedge (c = c' + 1)$.

Also, the BN network includes an observable answer node Y_{rc}^t for each skill in the rubric and each task $t = 1, \dots, 12$ in the sequence of 12 similar tasks administered during the CAT experiments. Observing $Y_{rc}^t = 1$ means that the pupil has solved the t -th CAT schema using an algorithm of complexity corresponding to the c -th row of the rubric, and requesting the help in the r -th column. By way of example, a student solving the t -th schema using a 0-dimensional algorithm and with voice, empty schema and feedback (0D-VSF) result in the observed node $Y_{11}^t = 1$.

In principle, all answer nodes should be explicitly observed through specific interactions with the pupil. However, this was not possible for the case of the dataset collected by [16] since the pupils were left free to choose their preferred approach. Therefore, to make such a dataset compatible with our model, in [15] we encode the collected answers as follows: a task t solved at level c^* by an algorithm with complexity r^* was translated into $Y_{rc}^t = 1$ for all competence levels rc lower than or equal to r^*c^* , thus assuming that, if requested, the pupil would have been able to implement solutions requiring a lower competence level than the one used. Similarly, we set all answer nodes $Y_{rc}^t = 0$ for all higher levels, leaving those not directly comparable unobserved. As an example, Table III (i) illustrates the case in which the pupil has generated as a solution for task t a one-dimensional algorithm using only the empty schema and the voice (1D-VS).

The choice made in our previous work [15] also contributed to stress the skills ordering. Since the ordering is modelled by explicit constraints imposed through the auxiliary variables $D_{rc,r'c'}$, such a choice would be unnecessary and detrimental, as it would artificially multiply the number of observations. Therefore, in the constrained model, a task t solved at level c^* by an algorithm with complexity r^* would be better translated into the single observation $Y_{r^*c^*}^t = 1$. However, in the specific experimental setting adopted in [16], since pupils were always allowed to try solving the task with the lowest competence level (0D-VSF), a failure could only be observed for that level, with the consequence that only answer nodes Y_{11}^t can be directly observed in the false state $Y_{11}^t = 0$. To work around this problem, we set the answer nodes just above the one observed in the true state, i.e., $Y_{r^*(c^*+1)}^t$ and $Y_{(r^*+1)c^*}^t$ to the false state, leaving all other nodes unobserved. As an example, Table III (ii) shows how the answer encoding changes in the case of a 1D-VS solution to task t for the constrained model.

Finally, having observed that, depending on the specific CAT schemes, other skills than those in the assessment rubric may be necessary, especially for the algorithm complexities 1D and 2D, by analysing the CAT schemes structures and characteristics, we identify ten supplementary skills, divided into three groups, to be added to the skill nodes in the network: (S_1) paint dot – group 1; (S_2) fill empty dots – group 2; (S_3) paint monochromatic rows or columns – group 2; (S_4) paint monochromatic squares – group 2; (S_5) paint monochromatic diagonals – group 2; (S_6) paint monochromatic l-shaped patterns – group 2; (S_7) paint monochromatic zigzags – group 2; (S_8) paint polychromatic rows or columns – group 3; (S_9)

TABLE III
ANSWER ENCODING, ASSUMING A PUPIL HAS GENERATED A 1D-VS SOLUTION FOR THE t -TH SCHEMA: $Y_{22}^t = 1$. SYMBOL \emptyset INDICATES THAT THE ANSWER NODE IS NOT OBSERVED.

		Competence levels		
		VSF $c = 1$	VS $c = 2$	V $c = 3$
(i) Unconstrained BN-based learner model				
Competence components	0D ($r = 1$)	1	1	\emptyset
	1D ($r = 2$)	1	1	0
	2D ($r = 3$)	\emptyset	0	0
(ii) Constrained BN-based learner model				
		Competence levels		
		VSF $c = 1$	VS $c = 2$	V $c = 3$
Competence components	0D ($r = 1$)	\emptyset	\emptyset	\emptyset
	1D ($r = 2$)	\emptyset	1	0
	2D ($r = 3$)	\emptyset	0	\emptyset

paint polychromatic diagonals or zigzags – group 3; (S_{10}) repetition of a pattern – group 3.

Note that, the first group of supplementary skills is made of the variable S_1 alone, which is necessary to implement a 0D algorithm. The second group includes all the monochromatic structures and is associated with 1D algorithms. The last group, which combines all polychromatic structures and the ability to repeat a structure, identifies the skills without which it would not be possible to develop a 2D algorithm.

From the annotations collected during the experimental study in [16] it was possible to extract direct observations about using each supplementary skill in each task. Consequently, answer nodes $Y_{S_i}^t$ were added to the network for each task $t = 1, \dots, 12$ and each supplementary skill S_i with $i = 1, \dots, 10$. Each schema can be solved using one or more supplementary skills, but using all of them is not always possible. Answer nodes $Y_{S_i}^t$ take the value one if the pupil has used the i -th supplementary skill in the solution of CAT schema t , and zero otherwise.

As described in Section II-C, a noisy-OR combines the variables in the same group into the group auxiliary nodes G_i , with $i = 1, \dots, 4$, where G_4 combines the target skills X_{rc} . In contrast, the relation between the group nodes and the target skills is conveyed through the logical AND.

C. Summary Score Metric

To evaluate pupils' competence level in a specific CAT schema, we use the CAT score, a metric provided by [16], taking values from 0 to 4, as shown in Table IV. Unsolved CAT schemes, which were not considered in [16], are here assigned a score of -1 .

When using the BN-based learner model, a summary metric of the inferences at the end of the 12 tasks, hereafter referred to as the BN-based CAT score, was obtained as the sum of the marginal posterior probabilities of all target skill nodes estimated by the model and can be interpreted as the expected number of competence levels mastered by the student.

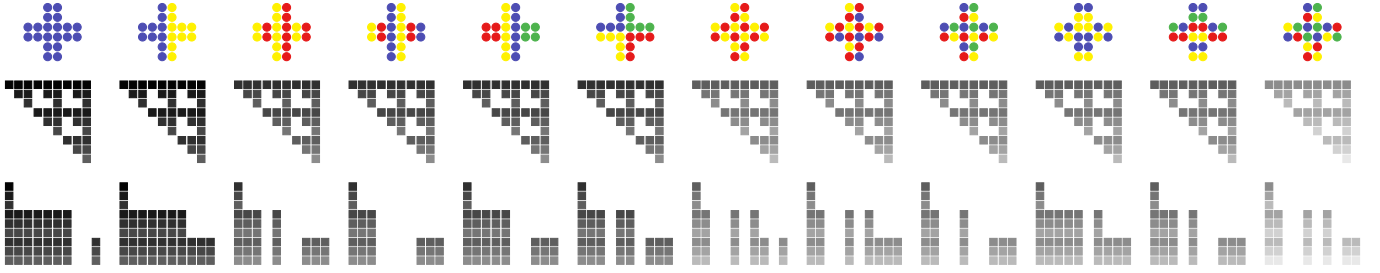


Fig. 5. The 12 CAT schemes T (top); the values of the inhibition parameters λ_{rc}^t for the target skill nodes (centre); the value of the inhibition parameters $\lambda_{S_i}^t$ for the supplementary skill nodes (bottom). The inhibition parameters for both the target and supplementary skill are depicted as a matrix of nine rows representing the answers, and as many columns as the number of modelled skills. The strength of the skill-answer relation has eleven levels, from 0.1 to 0.6 with a step of 0.05. Darker shades of grey mean lower skill-answer inhibition probabilities and white squares denote non-relevant skills.

TABLE IV
THE CAT SCORE METRIC USED IN [16] TO GRADE A PUPIL ON A SINGLE TASK (ADAPTED FROM [15]).

	VSF	VS	V
0D	0	1	2
1D	1	2	3
2D	2	3	4

As a baseline summary measure used for checking the consistency of this score with the expert-based assessment in [16], we define the *CAT score* as the average of the individual scores assigned to the 12 CAT schemes solved by each student according to Table IV.

D. Parameters' Elicitation

Having defined the structure of the model for the CAT activity, it is necessary to set the values of the prior probabilities π_* , and the 12 inhibitors λ_*^t , $t = 1, \dots, 12$, for both the target and supplementary skills. As in [15], uniform prior probabilities, i.e., $\pi_{rc} = 0.50$, have been assigned to each skill. However, when conditioning given the constraints nodes $D_{rc,r'c'} = 1$, their probabilities, before the observation of any answer node, become $\pi_{11} = 0.95$, $\pi_{12} = 0.8$, $\pi_{13} = 0.5$, $\pi_{21} = 0.8$, $\pi_{22} = 0.5$, $\pi_{23} = 0.2$, $\pi_{31} = 0.5$, $\pi_{32} = 0.2$, $\pi_{33} = 0.05$. Concerning the inhibition parameters, we consider two different models. Similarly to what was done in [15], in the first *baseline* model, hereafter referred to as *Model B*, all inhibitors are set to the same value. In contrast, a domain expert has elicited parameters in the second *enhanced* model, hereafter referred to as *Model E*.

Model B may look trivial and unrealistic, but it allows one to understand better the effect of the constraints resulting from ordering the skills and supplementary skills on the model inferences. The constant value of λ was chosen equal to 0.2, except for the leak node, associated with all answer nodes and modelling a guess probability of 0.1, resulting in $\lambda_{leak} = 0.9$.

Model E is an enhanced version of the baseline model designed to address the increasing difficulties of the 12 tasks and the challenges students may encounter applying their skills to different schemes. The expert elicitation process involved grouping the 12 schemes into eight categories of increasing difficulties based on their characteristics: (i) T1, (ii) T2, (iii)

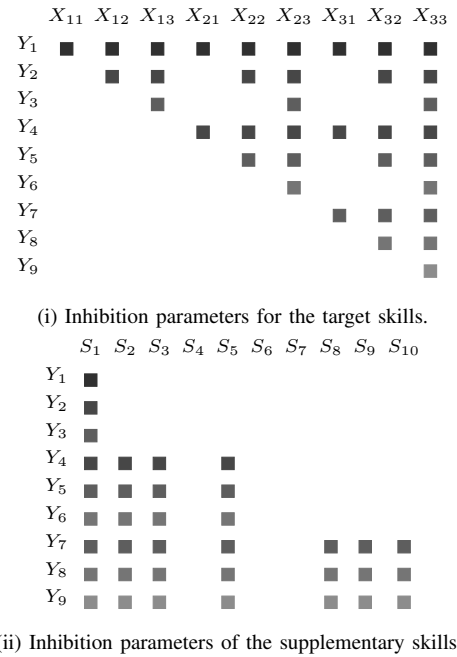


Fig. 6. Inhibition parameters λ_{rc}^{T3} used in the ECS model for schema T3 (Zoom on schema T3 of Fig. 5). The parameters are divided into two sets: the target skills (top) and the supplementary ones (bottom). The supplementary skills S_4 (paint monochromatic squares), S_6 (paint monochromatic ls), and S_7 (paint monochromatic zigzags) are represented as empty columns because they cannot be used to solve task T3.

T3, T4, (iv) T5, T6, (v) T7, T8, T9, (vi) T10, (vii) T11, (viii) T12. The expert assumed all tasks could be solved with 0D, 1D, and 2D algorithms. Moreover, given a schema t and a manifest variable Y_{rc}^t , the same inhibition probability was assumed for all relevant skills, meaning that all have the same probability of successfully being applied in solving scheme t with level rc . In the proposed method, the inhibitor parameter λ_{rc} is used to model the probability of failing a task of a particular difficulty level rc , assuming the student has the necessary skills to solve the task. When a task is more complex or less help is available to the student, the value of rc increases, which means that the inhibitor parameter also increases. This is because when the student possesses the necessary skills to solve a difficult task, the probability of failing is higher than for a simpler task. Similarly, the inhibitor

parameter λ_{rc}^t is assigned to a particular scheme t and is used to model the difficulty of implementing a solution of level rc for that scheme. A high value of λ_{rc}^t means that it is difficult to implement a solution of level rc for that scheme. In other words, the inhibitor parameter λ_{rc}^t provides a measure of the difficulty of implementing a particular solution for a given scheme at a given level of complexity.

While the students are generally expected to use 2D algorithms to solve the tasks optimally, there may be cases where a simpler 2D solution may be optimal. Nonetheless, in the current implementation, the first two tasks are designed to serve as starting points for students, introducing them to the activity. They are expected to be solved using simpler 1D algorithms. However, this particular case was not included in our model. This could have been described by setting high inhibitor values to indicate that certain 2D solutions are more difficult to implement than others, making them less likely to be chosen by students.

Our succinct elicitation setup allows summarising both the BN topology and its parameter values graphically (see the monochromatic rows at the bottom of Fig. 5) and explained more in detail in Fig. 6 for CAT schema T3.

The underlying BN has been implemented within the CREMA Java library [20], which supports the specifications of noisy gates and inference based on these parametric CPTs. The network size allowed for exact inferences using the Variable Elimination algorithm [21].

IV. RESULTS

To evaluate the performance of our models, we processed the answers provided by all 109 pupils included in the study in [16], obtaining for each model the posterior probabilities of the nine target skills and eventually for the supplementary skills. From the posterior probabilities of the target skills, we derived the BN-based CAT score, and the original CAT score from [16], as defined in Section III. Fig. 7 illustrates the correlation between these two scores for the following four models: the baseline model from [15] (Model B), the baseline model with constraints (Model BC), the one which also includes the supplementary skills (Model BCS) and finally the enhanced model including both constraints and supplementary skills (Model ECS). In the scatterplot, the BN-based CAT score, originally in the $[0, 9]$ range, has been rescaled in the $[0, 4]$ range to obtain a more direct graphic comparison. In all cases the Pearson correlation coefficient, indicated in Fig. 7 as ρ , is very high, confirming consistency between the BN-based assessment and the one performed by experts.

Table V shows the inference time of each model for all 12 tasks and all 109 students, as well as the estimated inference time to provide an assessment for a single student in all tasks and for a single student in a single task. It is important to note that the inference times for a single student in a single task may vary depending on the complexity of the task. However, these average inference times provide a rough estimate of the performance of the four models in terms of inference time. The total inference times of the four models can vary significantly, with Model B and Model BC being particularly fast at under

TABLE V
INFERENCE TIMES FOR ALL MODELS.

Model	Total inference time (sec)	Inference time per student (sec)	Inference time per student per task (sec)
Model B	29.615	0.272	0.023
Model BC	28.940	0.266	0.022
Model BCS	316.555	2.904	0.242
Model ECS	306.517	2.812	0.234

30 seconds, while Model BCS and Model ECS have a much longer inference time, taking over 5 minutes each. However, it's worth noting that when considering the assessment time for a single student, all models perform very quickly and can be considered suitable for real-time applications.

Besides the overall consistency of the summary metrics, the probabilistic assessments derived from the BN-based models provide more detailed information about student competence profiles in the form of posterior probabilities for each competence level and, when included, for the supplementary skills.

When looking at the posterior probabilities for the individual students, it is possible to understand their competence profile better and demonstrate the model interpretability. Moreover, by comparing competence profiles issued by the four models considered, it is possible to discern relevant differences between them and recognise the effects of the different improvements introduced in this work. To this goal, we summarise in Table VII the answers provided by four representative pupils, in Table VI the BN-based CAT scores versus the CAT score, and in Tables VIII and IX the corresponding posterior probabilities inferred by the models for the target and the supplementary skills, respectively. The subset of students we use to demonstrate the performance of the models has been selected by choosing some pupils with interesting situations.

TABLE VI
COMPARISON OF THE BN-BASED CAT SCORES WITH THE CAT SCORE FROM [16] OF ALL MODELS FOR A REPRESENTATIVE SUBSET OF PUPILS.

Student	CAT score	BN-based CAT score
21	3.3	2.23 (Model B)
		1.65 (Model BC)
		1.98 (Model BCS)
		2.00 (Model ECS)
33	0.75	2.00 (Model B)
		0.00 (Model BC)
		1.33 (Model BCS)
		1.47 (Model ECS)
81	1.75	2.90 (Model B)
		0.07 (Model BC)
		1.62 (Model BCS)
		1.82 (Model ECS)
92	2.5	1.77 (Model B)
		1.42 (Model BC)
		1.59 (Model BCS)
		1.79 (Model ECS)

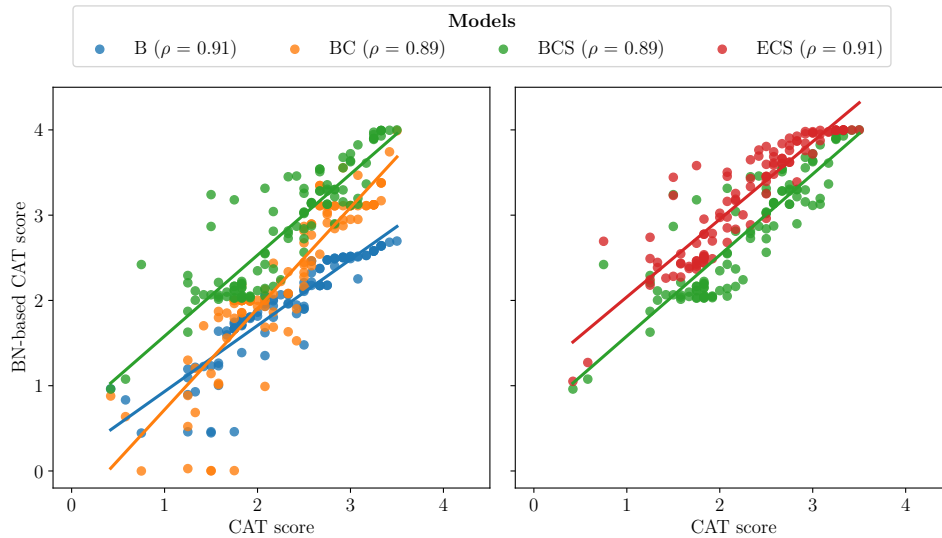


Fig. 7. Scatterplot of the BN-based CAT scores versus the CAT score from [16] for the four models under comparison: *B* — baseline proposed in [15] —, *BC* — baseline model with constraints —, *BCS* — baseline model with constraints and supplementary skills — and *ECS* — enhanced model with constraints and supplementary skills. The legend gives the Pearson correlation coefficient ρ between the two metrics.

TABLE VII
ANSWER TO THE 12 SCHEMES, EXPRESSED AS TARGET AND SUPPLEMENTARY SKILLS, FOR A REPRESENTATIVE SUBSET OF PUPILS.

Student	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12
21	1D-V S ₂	1D-V S ₂ ; S ₆	2D-V S ₃ ; S ₁₀	1D-V S ₃	1D-V S ₃ ; S ₄	1D-V S ₆	2D-V S ₈ ; S ₁₀	2D-V S ₁ ; S ₅ ; S ₁₀	2D-V S ₁ ; S ₁₀	1D-V S ₁ ; S ₄	1D-V S ₁	1D-V S ₁ ; S ₅
33	1D-V S ₂	1D-VS S ₂ ; S ₆	1D-VS S ₃	1D-VSF S ₃	1D-VS S ₃	1D-VS S ₁ ; S ₃	1D-VS S ₅	fail	fail	fail	fail	fail
81	1D-V S ₂	1D-V S ₂ ; S ₆	1D-V S ₃	1D-VS S ₃	1D-V S ₃ ; S ₄	1D-V S ₆	2D-VSF S ₁ ; S ₅ ; S ₁₀	0D-VS S ₁	2D-V S ₁ ; S ₁₀	fail	fail	fail
92	1D-V S ₂	1D-V S ₂ ; S ₆	1D-V S ₃	1D-V S ₃	1D-V S ₃ ; S ₄	1D-V S ₆	0D-V S ₁	0D-V S ₁	0D-VSF S ₁	1D-VS S ₄ ; S ₅	2D-V S ₁ ; S ₁₀	0D-V S ₁

Pupil 21 is a high-performing student, in terms of CAT and BN-based scores, who consistently used high-level artefacts for all CAT schemes and primarily used 2D algorithms.

Pupils 33 and 81 cannot be considered high-performing since they failed to complete some of the CAT schemes. Pupil 33 solved only the first seven schemes, always using 1D algorithms and almost always relying on the VS artefact. In comparison, student 81 was successful in the first nine schemes where he applied different algorithms and artefacts, but mostly the 1D-V. For both students 33 and 81, the BN-based CAT scores predicted by the four models vary significantly, indicating that the models may be producing different predictions of their abilities. The difference between the original and BN-based CAT scores is inconsistent across the models. For both students, the largest difference between the original and BN-based CAT scores is observed in Model B, which predicts a much higher score for both students. On the other hand, Model BC predicts a meager BN-based CAT score close to 0 for both students, indicating that this model may not be the most accurate for these particular students. This suggests that other models may be better suited for predicting their performance on the CAT.

Pupil 92's performance was strong, as he successfully com-

pleted all 12 tasks using different skill levels. He solved the first six schemes with the 1D-V skill, reduced the algorithm's complexity in the following ones, changed artefact for some of the more complex tasks, and applied the highest level skill, 2D-V, in a tricky schema. Regarding the BN-based scores, all four models predicted a lower BN-based CAT score for student 92 than the original CAT score, although the differences were not as large as those observed for students 33 and 81. This suggests that student 92 is a relatively strong performer overall, but there is potential for improvement in his performance.

For all students, the baseline model assigns posterior probabilities equal or very close to one to the most used skill levels but fails to recognise that they also possess lower level skills, to which rather small probabilities, eventually equal to zero, as for the worst performing students 33 and 81, are assigned.

On the one hand, when the ordering between skills is explicitly imposed, this problem is solved: the probabilities of lower skills increase, and those of higher skills decrease. This may lead to an excessive penalisation of higher skills, as in the case of pupils 33 and 81, where, as a consequence of the repeated failures in applying even the lowest competence level, Model BC decides for the total absence of the competencies under examination, returning a posterior probability of zero,

TABLE VIII
POSTERIOR PROBABILITIES $P(X_{rc} = 1|Y)$ OF ALL MODELS, FOR A REPRESENTATIVE SUBSET OF PUPILS.

Student	Model	X_{11}	X_{12}	X_{13}	X_{21}	X_{22}	X_{23}	X_{31}	X_{32}	X_{33}
		0D-VSF	0D-VS	0D-V	1D-VSF	1D-VS	1D-V	2D-VSF	2D-VS	2D-V
21	Model B	0.50	0.51	0.67	0.51	0.57	0.96	0.59	0.83	0.80
	Model BC	1.00	1.00	1.00	1.00	1.00	1.00	0.69	0.38	0.07
	Model BCS	1.00	1.00	1.00	1.00	1.00	1.00	0.97	0.95	0.92
	Model ECS	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
33	Model B	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
	Model BC	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Model BCS	1.00	1.00	0.52	1.00	1.00	0.05	0.59	0.30	0.00
	Model ECS	1.00	1.00	0.69	1.00	1.00	0.39	0.63	0.33	0.03
81	Model B	0.03	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
	Model BC	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Model BCS	1.00	1.00	1.00	1.00	1.00	1.00	0.91	0.21	0.03
	Model ECS	1.00	1.00	1.00	1.00	1.00	1.00	0.95	0.67	0.44
92	Model B	0.55	0.40	0.41	0.33	0.13	1.00	0.46	0.05	0.00
	Model BC	1.00	0.99	0.99	0.76	0.70	0.68	0.13	0.00	0.00
	Model BCS	1.00	1.00	1.00	1.00	1.00	1.00	0.59	0.19	0.01
	Model ECS	1.00	1.00	1.00	1.00	1.00	1.00	0.79	0.58	0.41

TABLE IX
POSTERIOR PROBABILITIES $P(S_i = 1|Y)$ OF MODEL BCS AND MODEL ECS, FOR A REPRESENTATIVE SUBSET OF PUPILS.

Student	Model	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}
		paint dot	fill empty dots	rows columns	squares	diagonal	1	zigzag	rows columns poly.	diagonals zigzag poly.	repetitions
21	Model BCS	1.00	1.00	1.00	1.00	1.00	1.00	0.40	1.00	0.26	1.00
	Model ECS	1.00	1.00	1.00	1.00	1.00	1.00	0.52	1.00	0.38	1.00
33	Model BCS	1.00	1.00	1.00	0.42	1.00	1.00	0.38	0.15	0.16	0.13
	Model ECS	1.00	1.00	1.00	0.43	1.00	1.00	0.42	0.21	0.22	0.19
81	Model BCS	1.00	1.00	1.00	1.00	1.00	1.00	0.36	0.34	0.31	1.00
	Model ECS	1.00	1.00	1.00	1.00	1.00	1.00	0.39	0.41	0.35	1.00
92	Model BCS	1.00	1.00	1.00	1.00	1.00	1.00	0.34	0.31	0.31	1.00
	Model ECS	1.00	1.00	1.00	1.00	1.00	1.00	0.36	0.35	0.35	1.00

even for the skills successfully used by the students in several schemes. These inferences look too severe for these situations, where an expert would rather attribute the errors to the specific difficulties of the failed tasks rather than the total lack of algorithmic skills.

On the other hand, when the supplementary skills are included in the assessment (Models BCS and ECS), this issue is solved, and the result of the posterior inference is consistent with the hierarchy of competencies defined by the rubric and the observations collected. In this case, the model understands that the failure follows from lacking the supplementary skills necessary to solve specific schemes with more complex algorithms and not from the lack of target skills.

For instance, according to models BCS and ECS, pupil 21 is likely to miss monochromatic zigzags (S_7) and polychromatic diagonals and zigzags (S_9), justifying the failure in applying the possessed 2D competence in schemes related to these supplementary skills.

Finally, employing more elaborate models, such as the ECS one, may, in some cases, reward the ability to apply high-level skills in more complex tasks, i.e., those assigned with higher inhibition probabilities, such as for pupil 92 who managed to solve scheme T11, a difficult one according to the parameters' elicitation in Fig. 5, using a 2D-V skill and thus 2D algorithms are given a much higher probability by Model ECS than by Model BCS.

A. Discussion

Recent literature shows that BNs are among the most common tools for modelling student knowledge in ITS [7], however, in most works, a great deal of effort is devoted to building an ad-hoc network and eliciting its parameters [22], [23] or collect data from which they could be learned [8], [10], [11]. This effort and technical complexity might be one reason why most ITSs in the literature concern primarily technical domains. For instance, [24] shows that most ITSs

focus on computer science education. If the reason for this phenomenon is that experts in this field are more comfortable using mathematical models, our proposal to base BN model elicitation on assessment rubrics, a popular educational tool for eliciting the relation between skills and behaviours, opens up the possibility of bringing this method closer to other, less technical disciplines.

Our study estimated the performance of four different models (Model B, Model BC, Model BCS and Model ECS) in assessing the algorithmic competence of 109 pupils based on their answers to 12 tasks. The models were evaluated by comparing the BN-based CAT score derived from the posterior probabilities of the target skills with the original CAT score obtained from the expert assessment. The results show a high correlation between the two scores in all models, indicating consistency between the BN-based assessment and the one performed by experts. The probabilistic assessments derived from the BN-based models provided more detailed information about student competence profiles in the form of posterior probabilities for each competence level and supplementary skills. The analysis of the models' posterior probabilities for the individual students highlighted relevant differences between the four models showing the effects of the improvements introduced in the study. The model interpretability was demonstrated by comparing competence profiles issued by the four models considered. The study found that Model B assigned posterior probabilities close to one of the most used skill levels but failed to recognise that the students also possessed lower-level skills, thus violating the ordering between competencies assumed by the assessment rubric. Instead, this issue was solved when the ordering between skills was explicitly imposed (Model BC). The result of the posterior inference was consistent with the hierarchy of competencies defined by the rubric. Finally, including the supplementary skills in the assessment (Model BCS) and refining the model's parameters (Model ECS) produced more detailed competence profiles and the evaluation quality more consistent with the observations collected and their interpretation by the experts.

Our study has demonstrated the effectiveness and efficiency of our approach to student knowledge modelling. Specifically, we found that the noisy-OR network structure can model student knowledge with high accuracy and minimal effort once the assessment rubric is defined. We first evaluated the baseline model, defined by only two parameters (i.e., *slip* and *guess* probabilities) assumed to be constant across questions and skills. Even with this minimal parameterisation, we found that the inferences about the learner's knowledge were consistent with the expert assessment. This suggests that the model is effective even when the number of parameters is small and can serve as a starting point for subsequent refinements. We have illustrated this refinement process in the enhanced model, which used different values of the inhibition parameters to describe the different task complexities while keeping the elicitation cost limited thanks to some simplifying assumptions. This refinement was shown to improve the quality of the assessment. Further refinements could be introduced by removing some of the assumptions.

Our approach is highly practical and efficient for real-world

educational settings. The initial effort required for elicitation is almost equivalent to that used to define the rubric, and subsequent refinements can be made with little additional exertion. Findings suggest that our approach can facilitate accurate and efficient student knowledge modelling for various educational contexts and teachers with a non-technical background.

V. CONCLUSION

This article is an extension of the work initially presented in the SoftCOM conference paper [15], which proposed a procedure for deriving a learner model for automatic skill assessment directly from the competence rubric of any set of tasks. This study aims to solve the two main limitations of the previous approach. Firstly, although partially implied by the structure of the model and implicitly defined in the assessment rubric, the ordering between competence levels was not strictly enforced in [15]. Secondly, in the previous modelling approach, either disjunctive or conjunctive gates must be used, while sometimes it would be helpful to combine both or implement more general relations between competencies. Supplementary skills, complementary to those in the rubric and not under assessment but necessary for the accomplishment of the task, were not included in the model. In this article, we provide a method to fulfil both needs. The first one is met by introducing dummy observed nodes in the network without changing its structure. The second was achieved by designing a network with two layers of gates, the first performing the disjunctive operation through noisy-OR gates and the second executing the conjunctive one by a logical AND.

The approach has been illustrated, and its feasibility demonstrated by implementing the automatic assessment of the Cross Array Task (CAT) [16]. Four models based on noisy gates Bayesian networks (BNs) have been developed and evaluated. By comparing the BN-based CAT score obtained for the 109 pupils performing the CAT to the original scores assigned by the experts, we have shown the consistency of the four models developed with the expert assessment. Moreover, by analysing the more detailed individual student assessment provided by the posterior probabilities, we have shown that by adding constraints and supplementary skills to the original baseline model, as well as by refining its parameters, it was possible to improve the quality of the assessment, its consistency the student behaviours, and the coherence between the model outcomes and the partial ordering between competencies defined by the assessment rubric, increasing the modelling tool's flexibility and allowing for a more accurate result. Outcomes also confirmed that changes from our previous modelling approach do not compromise the model compact parametrisation, its simple elicitation by experts and its interpretability, derived from using noisy gates BNs [13].

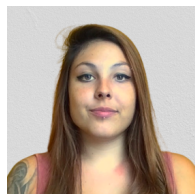
The proposed approach offers a promising practical, and efficient student knowledge modelling solution. By allowing for a straightforward translation of assessment rubrics into a flexible and scalable mathematical model, this approach enables more widespread use of Intelligent Tutoring and Assessment Systems (ITAS) that can interact with learners automatically and in real time. The combination of assessment rubrics and

the noisy-OR network structure makes this approach useful for teachers seeking to enhance student learning and performance, especially those without a technical background.

While the presented work focuses mainly on expert-based elicitation, ongoing work explores the possibility of improving the model by updating its parameters based on the evidence collected during user interaction.

REFERENCES

- [1] P. Dawson, "Assessment rubrics: towards clearer and more replicable design, research and practice," *Assessment & Evaluation in Higher Education*, vol. 42, no. 3, pp. 347–360, 2017.
- [2] A. T. Corbett and J. R. Anderson, "Knowledge tracing: Modeling the acquisition of procedural knowledge," *User modeling and user-adapted interaction*, vol. 4, no. 4, pp. 253–278, 1994.
- [3] S. E. Embretson and S. P. Reise, *Item response theory*. Psychology Press, 2013.
- [4] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.
- [5] E. Millán, J. L. Pérez-de-la Cruz, and E. Suárez, "Adaptive Bayesian networks for multilevel student modelling," in *International Conference on Intelligent Tutoring Systems*. Springer, 2000, pp. 534–543.
- [6] M. C. Desmarais and R. S. d Baker, "A review of recent advances in learner and skill modeling in intelligent learning environments," *User Modeling and User-Adapted Interaction*, vol. 22, no. 1, pp. 9–38, 2012.
- [7] E. Mousavinasab, N. Zarifasanaiey, S. R. Niakan Kalhori, M. Rakhshan, L. Keikha, and M. Ghazi Saedi, "Intelligent tutoring systems: a systematic review of characteristics, applications, and evaluation methods," *Interactive Learning Environments*, vol. 29, no. 1, pp. 142–163, 2021.
- [8] D. Hooshyar, R. B. Ahmad, M. Yousefi, M. Fathi, S.-J. Horng, and H. Lim, "Sits: A solution-based intelligent tutoring system for students' acquisition of problem-solving skills in computer programming," *Innovations in Education and Teaching International*, vol. 55, no. 3, pp. 325–335, 2018.
- [9] W. Xing, C. Li, G. Chen, X. Huang, J. Chao, J. Massicotte, and C. Xie, "Automatic assessment of students' engineering design performance using a bayesian network model," *Journal of Educational Computing Research*, vol. 59, no. 2, pp. 230–256, 2021.
- [10] L. Wu, "Student model construction of intelligent teaching system based on bayesian network," *Personal and Ubiquitous Computing*, vol. 24, no. 3, pp. 419–428, 2020.
- [11] E. U. Rodriguez-Barrios, R. A. Melendez-Armenta, S. G. Garcia-Aburto, M. Lavoignet-Ruiz, L. C. Sandoval-Herazo, A. Molina-Navarro, and L. A. Morales-Rosales, "Bayesian approach to analyze reading comprehension: A case study in elementary school children in mexico," *Sustainability*, vol. 13, no. 8, p. 4285, 2021.
- [12] Y. Huang, J. Guerra-Hollstein, J. Barria-Pineda, and P. Brusilovsky, "Learner modeling for integration skills," in *Proceedings of the 25th Conference on user modeling, adaptation and personalization*, 2017, pp. 85–93.
- [13] A. Antonucci, F. Mangili, C. Bonesana, and G. Adorni, "Intelligent tutoring systems by bayesian nets with noisy gates," *The International FLAIRS Conference Proceedings*, vol. 35, May 2022. [Online]. Available: <https://journals.flvc.org/FLAIRS/article/view/130692>
- [14] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.
- [15] F. Mangili, G. Adorni, A. Piatti, C. Bonesana, and A. Antonucci, "Modelling assessment rubrics through bayesian networks: a pragmatic approach," in *2022 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. IEEE, 2022, pp. 1–6. [Online]. Available: <https://ieeexplore.ieee.org/document/9911432>
- [16] A. Piatti, G. Adorni, L. El-Hamamsy, L. Negrini, D. Assaf, L. Gambardella, and F. Mondada, "The CT-cube: A framework for the design and the assessment of computational thinking activities," *Computers in Human Behavior Reports*, vol. 5, p. 100166, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2451958821001147>
- [17] M. Castoldi, *Valutare le competenze*. Carocci, 2009.
- [18] A. Jonsson and G. Svingby, "The use of scoring rubrics: Reliability, validity and educational consequences," *Educational research review*, vol. 2, no. 2, pp. 130–144, 2007.
- [19] E. Poulakis and P. Politis, *Computational Thinking Assessment: Literature Review*. Cham: Springer International Publishing, 2021, pp. 111–128.
- [20] D. Huber, R. Cabañas, A. Antonucci, and M. Zaffalon, "Crema: A java library for credal network inference," in *Proceedings of the 10th International Conference on Probabilistic Graphical Models*, ser. Proceedings of Machine Learning Research, M. Jaeger and T. D. Nielsen, Eds., vol. 138. Skørping, Denmark: PMLR, 23–25 Sep 2020, pp. 613–616. [Online]. Available: <https://proceedings.mlr.press/v138/huber20a.html>
- [21] M. Chavira and A. Darwiche, "Compiling bayesian networks using variable elimination," in *IJCAI*, vol. 2443, 2007.
- [22] A. Ramírez-Noriega, R. Juárez-Ramírez, and Y. Martínez-Ramírez, "Evaluation module based on bayesian networks to intelligent tutoring systems," *International Journal of Information Management*, vol. 37, no. 1, Part A, pp. 1488–1498, 2017.
- [23] M. Eryılmaz and A. Adabashi, "Development of an intelligent tutoring system using bayesian networks and fuzzy logic for a higher student academic performance," *Applied Sciences*, vol. 10, no. 19, p. 6638, 2020.
- [24] A. A. Soofi and M. U. Ahmed, "A systematic review of domains, techniques, delivery modes and validation methods for intelligent tutoring systems," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 3, 2019.



Giorgia Adorni is a PhD candidate at the Swiss AI Lab IDSIA, joint institute USI-SUPSI, with a background in Computer Science and Informatics. Her research interests include Artificial Intelligence, Machine Learning, Computer Science Education, Educational Robotics, Learning Technologies, and Human-Robot Interaction.



Francesca Mangili is a Senior Researcher at IDSIA, the Swiss AI Lab affiliated with the two Universities USI and SUPSI. She teaches courses on probability and applied statistics at SUPSI and has authored more than 40 peer-reviewed publications. Her main research interests are in the area of statistics, probabilistic reasoning and machine learning with applications in various fields among which the medical and educational ones stand out.



Alberto Piatti is director of the Department Education and Learning of the University of Applied Sciences and Arts of Southern Switzerland and professor in STEM Education. He has studied mathematics at the Swiss Federal Institute of Technology in Zürich and holds a PhD in Economics from the University of Lugano.



Claudio Bonesana is a Machine Learning Engineer and Researcher at IDSIA, the Swiss AI Lab affiliated with the two Universities USI and SUPSI. His main research interests are in the area of machine learning, distributed systems, and graphical models for AI and their application to real world problems.



Alessandro Antonucci is a Senior Lecturer-Researcher at IDSIA, the Swiss AI Lab located in Lugano affiliated with the two Universities of Southern Switzerland. He is the author of more than 100 peer-reviewed publications and an Area Editor of the International Journal of Approximate Reasoning. His research interests mostly focus on probabilistic graphical models for AI reasoning and machine learning and their application to explainability and causality theory.