# Can machine learning solve the challenge of adaptive learning and the individualization of learning paths? A field experiment in an online learning platform

Tim Klausmann[a], Marius Köppel[a], Daniel Schunk[a], Isabell Zipperle[a,*]

[a] *Johannes Gutenberg University of Mainz, Johann-Joachim-Becher-Weg 31, 55128 Mainz, Germany*

**Abstract**

The individualization of learning contents based on digital technologies promises large individual and social benefits. However, it remains an open question how this individualization can be implemented. To tackle this question we conduct a randomized controlled trial on a large digital self-learning platform. We develop an algorithm based on two convolutional neural networks that assigns tasks to $4,365$ learners according to their learning paths. Learners are randomized into three groups: two treatment groups – a group-based adaptive treatment group and an individual adaptive treatment group – and one control group. We analyze the difference between the three groups with respect to effort learners provide and their performance on the platform. Our null results shed light on the multiple challenges associated with the individualization of learning paths.

*Keywords:* Education, Digitalization, Learning Behavior, Adaptive Learning, Dynamic Difficulty Adjustment, Deep Learning, Neural Networks
*JEL:* C45, C93, I20, I21

## 1. Introduction

Individualizing learning paths instead of teaching along the grade-appropriate curriculum to the average learner in class is a promising approach. It might enhance learning by creating more personalized learning contexts. Whereas the baseline education production function suggests school inputs to be the same for all students (Hanushek, 2020), we propose to improve school inputs and thereby increase education outcomes by individualizing learning materials (comp. Banerjee et al., 2007; Vandewaetere et al., 2011; Escueta et al., 2020). Increased education outcomes benefit the learner herself, but also convey important positive external benefits for society. The theoretical introducing of individualization to the education production seems promising, although the exact form of the production function remains vague. In fact it is far from clear, how this individualization of learning should be done to effectively promote learning and education outcomes. The goal of this paper is to propose a new methodological approach to solve the the individualization puzzle.

Learning paths may be individualized in many different ways. Most classically, individualized learning concepts can be applied in physical classroom teaching by providing individualized materials to students (e.g. Banerjee et al., 2016). However, it demands a lot of flexibility and adoption ability if it is the responsibility of a single educator. With regard to limited time resources, 'teaching to the right level' requires educators to make difficult trade-offs between reducing the learning rate in order to keep the learners with the lowest learning levels on track or teaching along the grade-appropriate curriculum to the average learner in class (e.g. Banerji, 2000). Promising approaches to tackle these trade-offs in classroom settings therefore include the provision of special support staff to the learners with the low learning levels or difficulties to keep pace (e.g. Banerjee et al., 2007) or to track learners according to their learning levels and adjust teaching to the average level of a more homogeneous group (e.g. Duflo et al., 2011). The effective, large-scale implementation in public schooling systems, however, might be challenged by practical issues (Banerjee et al., 2016), such as resisting school bureaucracy or teachers that prefer sticking to their usual teaching practices.

The evolution of digital technologies in increasingly digitalized education systems and recent debates about its potential to increase education outcomes has, however, brought up new possibilities. We can now adjust learning materials and instructions to individual learners

*Corresponding author

*Email addresses:* tim.klausmann@uni-mainz.de (Tim Klausmann), mkoeppel@uni-mainz.de (Marius Köppel), daniel.schunk@uni-mainz.de (Daniel Schunk), isabell.zipperle@uni-mainz.de (Isabell Zipperle)

in and outside the context of classical classroom teaching (Banerjee et al., 2007; Vandewaetere et al., 2011; Bulman and Fairlie, 2016; Escueta et al., 2020; Chen et al., 2020). Digital technologies have decisive advantages, because they enable the individualization of learning paths in real time and at low cost.

The Covid-19 pandemic boosted a sudden shift from traditional classroom teaching to emergency remote teaching. Already before the pandemic in 2019, the implementation of digital education technologies in Germany was stimulated by the 'Digitalpakt Schule' policy with 5 bn EUR (BMBF, 2019). Such extensive government fundings aim at adjusting the education system to increasing digitalization. Learn management systems yield the opportunities for educators to distribute learning materials to individual learners. Moreover, in increasingly many digital applications learners navigate through their own digital learning environment independent of their classmates. Thus, education technology constitutes – in principle – a well-suited and easy to implement setting for learners to work on individually tailored learning content at their own speed. Given its potential for the enhancement of student learning outcomes, adaptive learning and the individualization of learning paths in digital learning environments have been subject to research in education, behavioral and computer sciences (comp., e.g., Sampayo-Vargas et al., 2013; Cornelisz and van Klaveren, 2018; Vanbecelaere et al., 2020; Martin et al., 2020, for a recent systematic literature review). The topic has also been discussed in recent literature in the field of the economics of education literature (comp., e.g., Banerjee et al., 2007; Van Klaveren et al., 2017; Muralidharan et al., 2019; Klausmann and Schunk, 2021). This literature indicates, however, that designing digital learning environments that prove effective in improving learning processes and outcomes, is highly complex. First, there are different approaches to individualization per se (Lee and Park, 2008; Vandewaetere et al., 2011). Second, effectiveness of individualization may depend on the way it is implemented. One way to implement individualization is via micro-adaptive instruction. Micro-adaptive instruction is a dynamic approach accounting for a broader set of temporal individual and situational parameters. This implies that micro-adaptive instruction relies on measures that are derived from the interaction between learner and the learning environment. Common ways of implementing micro-adaptive learning in online learning platforms include, e.g., rule-based or probability based approaches (for an overview see Vandewaetere et al., 2011).

We contribute to this debate by proposing an innovative machine learning (ML) approach for the individualization of learning paths of college-level learners using a self-learning online platform. The general idea of ML algorithms is to make predictions on unseen data based on data they have been trained with.[1] Online learning platforms collect large amounts of user-platform-interaction data that are a main requirement for ML algorithms to operate. More precisely, the platform tracks the interactions of many users with one task. Being trained on these data and informed about a user's characteristics the algorithm is enabled to predict performance on that task for a user that hasn't worked on that task before (for a recent literature review on the application of ML in education research see Chen et al., 2020). Neural networks, that are a common form to implement ML algorithms, have the advantage that they are very flexible and allow for a higher dimensional non-linear correlation between input factors than linear models. They, therefore constitute a methodological improvement to linear models.

Linear models have been applied in the related studies conducted by Van Klaveren et al. (2017), Cornelisz and van Klaveren (2018) and Klausmann and Schunk (2021) who also developed adaptive algorithms in self-learning online platforms. The algorithm applied in Van Klaveren et al. (2017) and Cornelisz and van Klaveren (2018) adapts learning materials to individual learners based on the learner's relative understanding level of a certain topic. Relative understanding of a certain topic is a deterministic metric that they define as the ratio of correct answers and total number of answers given. Their algorithms are designed in a way that a learner gets assigned a more difficult task as soon as this metric for relative understanding crosses a certain threshold. Similarly, the algorithm implemented by Klausmann and Schunk (2021) assigns tasks to learners based on a measure that captures the individual solution probability for each learner-task-combination. Whereas these studies advance the state of knowledge of adaptive learning in the way they conceptualize adaptive learning, the effects of these implemented algorithms on actual learning outcomes remain limited. Van Klaveren et al. (2017) find that the adaptive algorithm compared to a static algorithm increased difficulty of tasks and practice duration, but decreased learning performance of the learners. The results of Cornelisz and van Klaveren (2018) also point to a lower performance of learners that are learning with the adaptive software, but yield no effects on practice intensity, perceived attitudes towards practicing and test scores in summative exams. Klausmann and Schunk (2021) divide adaptive learning into two components. They find that learner increase their learning effort when tasks are assigned at the right level compared to a fixed curriculum, but do not find any additional ben-

[1]The relevance of ML algorithms as a reliable prediction method are currently discussed in the fields of economics (comp., e.g., Athey, 2018) and education sciences (comp., e.g., Lee and Park, 2008; Vandewaetere et al., 2011; Escueta et al., 2020; Chen et al., 2020) is subject to an ongoing debate.

efit from individualization.

In contrast, the ML algorithm applied in our setting provides the opportunity of making dynamic predictions about individual learners' future learning outcomes. Our algorithm is set up as a convolutional neural network. More specifically, we apply two consecutive convolutional neural networks where the prediction of the first neural network is used as an input factor for the second neural network. This enables us to predict individual task difficulties and future learning effort of learners for given tasks. The ML algorithm is pre-trained on more than 5 million historical learner interactions with the platform that are used for assigning tasks to individual learners. During the training phase, the ML algorithm gradually learns from historical platform data, i.e., it learns how likely learners with given behaviors are to solve a given task on the platform. Further, it learns for how long learners are likely to carry on using the platform having faced this given task. Being trained on a sufficient amount of data the algorithm makes predictions about learning outcomes of unfamiliar learners. Therefore, it will match characteristics of the unfamiliar learners to those of the learners whose data it has been trained on. We can use the predictions of the algorithm to assign tasks to a learner. These tasks are ought to be solved with a predicted probability and motivate the learner to keep on learning for a maximal amount of time.

In a randomized controlled framework we treat two different groups of learners by letting the algorithm assign them tasks of differing degrees of individualization based on the predictions of task difficulty and future learning effort. One treatment group is assigned with tasks based on the average predictions for the learning group (group-based adaptive treatment group), whereas the second treatment group receives tasks based on individual predictions (individual adaptive treatment group). We compare the two treatment groups and the control group with respect to learning effort (number of solved tasks) and performance (number of correctly solved tasks) on the platform. Although we find null results for the treatment effects between treated and untreated learners, our contribution to the literature is twofold. First, we approach the known challenge of individualizing learning with a ML algorithm that we have specifically designed for this purpose and context, and that is retrained in real-time with current learning data. Second, by shedding light on what aspects of individualization do not work (based on a conclusive treatment-control comparison), we demonstrate once again that improving learning outcomes by individualizing learning paths is a challenging exercise, and we also prepare the ground for future more elaborate approaches that follow up on our work.[2]

## 2. Design and prediction algorithm

*Setting.* We use data from a large online learning platform named pruefungs.tv. The platform provides college-level students with learning videos, exercises, and mock exams for self-learning. Students use the platform to prepare for the final high-stake standardized exam of their degree. We implement our intervention in a section providing multiple choice tasks to overall $4,365$ learners. The pool of tasks that are presented to the learners within this section is created by trained educators and tailored to the content of the learners' courses. Learners can attempt each task on the platform only once a day.

*Intervention.* We run a randomized controlled trial in the learning intensive time between July and November 2021 that constitutes the period prior to the final exam .[3] We randomly assign students to one of three groups that differ in the way they get tasks assigned by our ML algorithm: a control group ($n = 1,454$), a group-based adaptive treatment group ($n = 1,455$), and an individual adaptive treatment group ($n = 1,456$).

The control group receives tasks randomly drawn from the pool of all tasks available to the learner on the respective day. The group-based adaptive treatment group receives tasks based on the common learning path of the learning group, thus the average predictions of all learners on the platform for task difficulty and future learning effort. The individual adaptive treatment group receives tasks based on the learning path of the individual learner, thus based on the individual predictions of the ML algorithm. As an example, Figure 1 visualizes the task assignment for a given learner in one of the two treatment groups on a specific day. Tasks available on that day (blue dots in Figure 1) are assigned to that learner according to pre-tested probability distributions that optimize task difficulty and maximize future learning effort. Thus, the ML algorithm predicts certain point within the distributions and draws tasks that are closest to the predicted points. The heat-map shows the joint two-dimensional distribution that the algorithm draws the tasks from that the learner get assigned with. The lighter the area in Figure 1, the more likely it is that a task from this area is drawn.

Task difficulty, the first dimension, is the probability of solving a task correctly (y-axis in Figure 1). The optimization of task difficulty follows a distribution pretested by Klausmann and Schunk (2021) within the same learning platform. The recommended distribution is a logit-normal distribution ($\mu = -0.85$ and $\sigma = 0.8$) with a predicted average error probability of 30 percent.

[2]Code for the used ML models can be found at `https://doi.org/10.5281/zenodo.6337835`

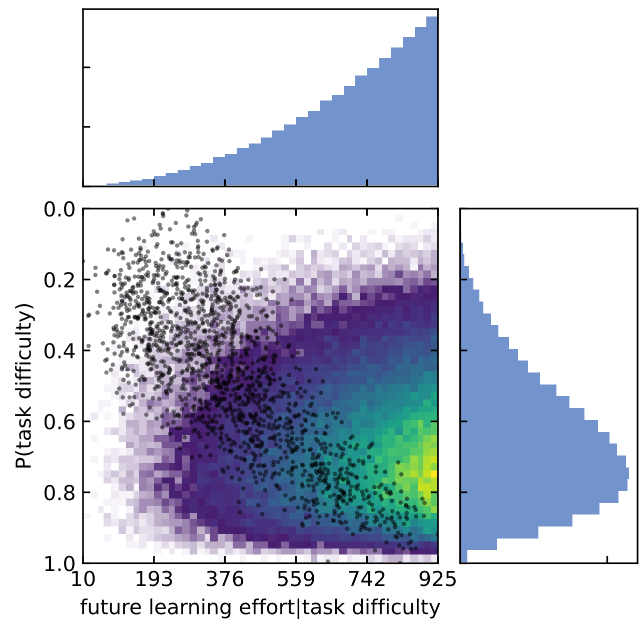
[3]We preregistered this study as Klausmann et al. (2021) and received ethics approval by the joint ethics commission in economics by Goethe University in Frankfurt and the University of Mainz.

Future learning effort, the second dimension, is approximated by the number of tasks that learners complete in the future after being assigned with a certain task (x-axis in Figure 1). For the optimization of learning effort we assume that more effort is better. Thus, in this dimension we draw from a cube root distribution.

*Prediction Algorithm.* Our ML algorithm predicts task difficulty and future learning effort for a given task on the platform to individually assigning tasks to learners.[4]. Our two consecutively applied convolutional neural networks are pre-trained with historical data on learner-platform interactions in the same learning platform, collected over two years directly prior to the intervention. The networks are trained based on pre-determined input features capturing learner characteristics and learning behavior meaning that they learn how these characteristics relate the learning outcomes of our interest. More precisely, the input features that we use to train our networks are learner-specific measures recorded by the platform (e.g. the number of tasks a learner solved before the current interaction with the platform, the number of tasks a learner solved correctly before the current interaction, the last time a learner was logged into the platform, etc.). Based on these features, the first network is trained to predict the probability that a learner answers a given task correctly. This information is concatenated to the input features to train the second network which predicts for the given task the number of tasks the learner will take in the future. Our reasoning for applying the two networks consecutively and using the first network's prediction of solving probability as an input to the second network is motivated by motivation theory. Accordingly, pursuit of success and effort expenditure is linked to the perceived difficulty of a task approached (e.g. Atkinson, 1957; Brehm and Self, 1989; Karabenick and Youssef, 1968). Thus, we assume that solving probability is an important predictor of future learning effort. Being confronted with a task that is too easy might impede future effort because it might not be considered crucial whereas being confronted with a too difficult task might deteriorate motivation.

During the training phase prior to the intervention we perform a five-fold cross-validation on the available data on pre-intervention learner-platform interactions to identify the optimal hyper-parameters for the networks. Whereas the first network predicting solving probability of a given task was trained in a classification setup, the second network predicting future effort was trained in a regression fashion. We evaluate the performance of both networks by using the area under the receiver operating characteristic space (AUC). We consider the upper 10% of predicted future tasks to be the positive class which allows for an upper bound approximation on this prediction. We report an average performance of an AUC of 0.70 over all cross-validation splits for the most optimal hyper-parameters.

Figure 1: Adaptive task assignment




*Note:* The graph displays task assignment to an example learner at a specific day. The y-axis visualizes task difficulty. The x-axis visualizes future learning effort. The blue dots represent available tasks on the specific day. Tasks are assigned according to a draw from a two-dimensional distribution. The distribution that we draw from for the first dimension, task difficulty, is shown in the histogram on the right. The distribution that we draw from for the second dimension, future learning effort, is shown in the histogram on top. The heatmap shows the joint two-dimensional distribution. We draw tasks using a nearest neighbor approach. This means that the algorithm predicts a certain point in the two distributions. Finally the task that - in terms of task difficulty and future learning effort - is closest to the predicted point is assigned to the learner.

We use these evaluated optimal hyper-parameters during the intervention. To assign tasks to learners in the group-based adaptive treatment group, we calculate the mean difficulty and number of future tasks over all learners on the platform using the predictions of the two networks. Moreover, for the individual adaptive treatment group we use individual predictions for task assignment. During the intervention, we fully retrain the whole networks each night with the constantly incoming data on the learner-platform interactions in order to

[4] A sketch of the network architecture is displayed in A.3. Technical specifications of the applied neural networks are displayed in Table A.1

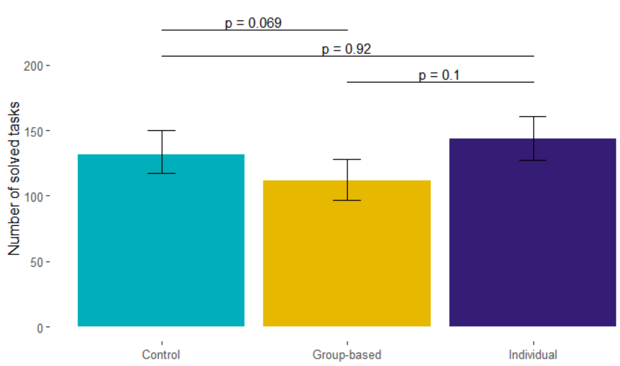
dynamically adapt our algorithm.

## 3. Results

Overall, we do not find that learners in both of our treatment groups who get tasks assigned by our ML algorithm provide more effort or perform better on the platform. Further, on most days both neural networks do not perform better than random assignment and in fact much worse than during our cross-validation phase before the intervention.

*Data.* We collect data for each learner-task interaction with the multiple choice task section of the learning platform. A learner-task interaction implies that a learner faces a multiple choice task and attempts to solve it. More specifically, we track (i) which task a learner faces, (ii) when it is faced, and (iii) whether it is solved correctly. Thus, we track effort (number of solved tasks) and performance (number of correctly solved tasks) of all learners during the intervention. On average, a learner on the platform faced 129 tasks and solved 77 tasks correctly during the time of intervention. Moreover, we record the performance of the neural networks by collecting predictions and comparing them with realized outcomes on a daily basis.

Figure 2: Treatment effect of effort

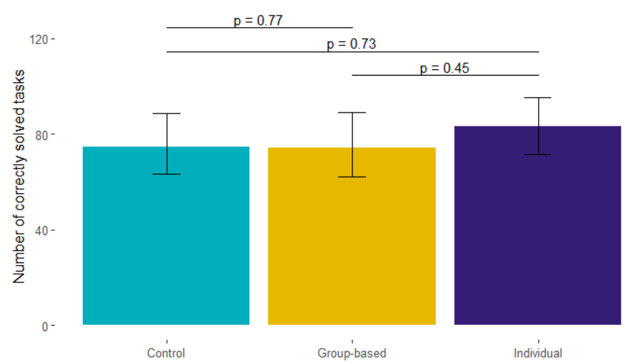



*Note:* We do not find that any kind of adaptivity changes learning effort on the platform. The bars correspond to the control group and the two treatment groups. The y-axis depicts number of tasks that learners solved during the intervention. We report bootstrapped confidence intervals. The brackets display pairwise non-parametric Wilcoxon tests.

*Treatment effect.* Figure 2 shows the treatment effect regarding effort. Comparing the control group to the group-based adaptive treatment group we find that learners in the treatment group solved 20 tasks less ($p > 0.05$, 132 vs. 112 tasks on average). Comparing the control group to the individual adaptive treatment we find that learners in the treatment group solved 11 tasks more ($p > 0.05$, 132 vs. 143 tasks on average). Also, the difference between both treatments is insignificant ($p > 0.05$). Figure A.1 depicts that this pattern is replicated when splitting by subject. Accordingly, our main result reflects insignificant treatment effect in the single subjects the learners can choose from on the platform.

Figure 3 shows the treatment effect regarding performance. Similarly to learning effort, we do not find any significant differences between all three groups ($p > 0.05$). In this case, learners in the group-based treatment group and in the control group solved on average 74 tasks correctly. Learners in the control group and learners in the individual adaptive group differ by 9 correctly solved tasks. As for the treatment effect on effort, Figure A.2 shows that this result replicates in the treatment effects of performance by subject.

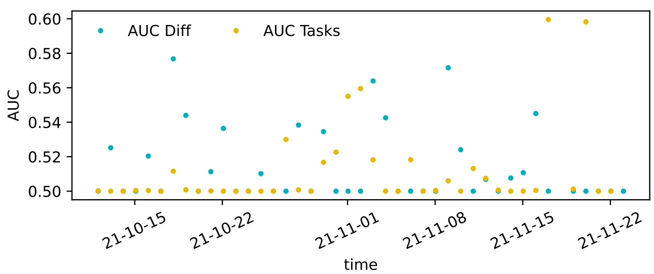
Figure 3: Treatment effect on performance




*Note:* We do not find that any kind of adaptivity changes learning performance on the platform. The bars correspond to the control group and the two treatment groups. The y-axis depicts number of tasks that learners solved correctly during the intervention. We report bootstrapped confidence intervals. The brackets display pairwise non-parametric Wilcoxon tests.

In Figure 4, we display the performance (AUC) of the two networks during the high learning period prior to the final exam of the students. Contrasting the performance evaluation of the trained networks before the intervention, the performance of the two networks during the intervention is significantly lower. Most of the time each of the models does not show a better performance compared to random guess. The number of days that both models show a reasonable performance are even more limited.

Figure 4: Model performance




*Note:* Model performance (AUC) of the two networks during the learning intensive period directly prior to the learners' final exams. The blue dots show the performance of the first network that is predicting the task difficulty of a learner while orange dots show the performance of the network that is predicting future learning effort.

## 4. Discussion and avenues for further research

Overall, we identify three possible reasons to explain the absent results and identify potential avenues for future research.

First, we might find null results due to low power (for our power calculation see Klausmann et al., 2021). During the training period our network was trained on a large data set including more than 5 million learner-platform interactions. Using this data pool, we reach high precision in the predictions of the ML algorithm. However, a general characteristic of self-learning platforms is that individual learners do not use the platform regularly. So, while we had enough data for training our network, we might have collected too few data for the ML algorithm to predict task-specific outcomes with a high precision for individual learners that used the platform during our intervention period. As assigning the right tasks to individual learners might require a large amount of data on the individual learner, this constitutes a central challenge for making accurate predictions within a short time horizon for individual learners in in our setting and in self-learning environments in more general. Future research should therefore ensure to collect a sufficiently large amount of data on the individual learner.

Second, null effects might stem from our set-up of the network itself. In line with motivation theory, we assume that individual future learning effort might depend on individual task solving probabilities. Thus, we are confident that interlinking a neural network predicting task solving probability with a network predicting future learning effort is a reasonable approach to individualization. However, we train each network globally on all learner-platform interactions that we observe prior to our intervention. Using the same network for each learner might have impeded learning of the algorithm about the individual. Future research should therefore start improving the algorithm by training individual networks for each learner or other ML algorithms using the feature representation of the pre-trained network. This implies, that the networks are again trained on the historical data of all learner-platform interactions, but branched out for each learner during the intervention period. More concretely, this means that the networks are simply re-trained with data of the individual learner only which would produce specialized networks following the 'journey' on the learning platform. Moreover, the algorithm in our set-up makes event-based predictions meaning that tasks are assigns tasks to a learner based on historical learner-task-interactions but not learner-task sequence-interactions. Including this time dimension could be a possible way to further develop the algorithm. It is also conceivable to improve outcomes by using a different form of ML algorithms. We applied a convolutional neural network as it is applicable in a flexible way and fulfills the requirements for making predictions in large data on large data sets. Other approaches could be to apply random forest or booting methods.

Last, our results may be influenced from the choice of our input features. We train our ML algorithm using data on learner-platform interactions that measure learning behavior on the platform to predict future learning outcomes. However, learning is a complex process. Applying a ML algorithm to implement individualization allows us to stay agnostic about the true underlying theoretical model of learning, thus, how the algorithm learns depends on our self-determined input features that we feed into the neural network. While we used primarily data on platform behavior and achieve high prediction accuracy in doing so during the training phase, we were not able to include more comprehensive data capturing the individual learner and the educational environment of the individual learner as input features which might be crucial to enhance model performance even more. Future research should therefore work on developing a more integrated idea of input features that might influence educational outcomes in the online platform learning setting in an important way and select input features accordingly.

## 5. Conclusion

Individualization of school inputs – in theory – constitutes a specification of the education production function as propsed by Hanushek (2020). It provides learners with the opportunity to learn contents that are tailored to their individual needs at exactly the pace they need to thoroughly process the learning materials.

In the light of ongoing digitalization of our environment, digital learning applications are increasingly entering the education sector providing a well-suited environment to implement individualization in an easy and scalable way. To make use of these benefits, we conduct a field experiment on adaptive learning in a self-learning online platform that prepares college-level students for their final high-stake exams. We apply a ML algorithm and thus take a micro-adaptive approach (comp. Lee and Park, 2008; Vandewaetere et al., 2011) to individualize learning paths of learners in the platform. Informed by motivation theory, we use two consecutive neural networks to predict task difficulty and future learning effort for learner-task interactions and assign individualized tasks to learners in our adaptive treatment groups.

Our contribution once again discloses the complexity of individualizing learning paths. In line with the findings of Cornelisz and van Klaveren (2018), Van Klaveren et al. (2017), Klausmann and Schunk (2021), we do not find that our adaptive treatment increases learning effort or performance. Whereas ML algorithms theoretically are a powerful approach to the challenge of individualizing learning paths (comp. Chen et al., 2020), the concrete set-up as well as the setting the algorithm is applied in are decisive. This paper has intended to provide ground for future research on adaptive digital learning, by using a large scale field experimental approach to shed light on the challenges associated with and some preconditions for the successful implementation of algorithms that enable the individualization of learning paths


### Acknowledgments

We would like to thank Johannes Laudenberg, Johannes Schulte, and Stephan Hansen. Katharina Hartinger, Erik Lehmann, Erik Sarrazin and Steffen Viete provided valuable feedback. There are no conflicts of interest; this research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. All errors in fact or interpretation are ours.


### References


Susan Athey. The impact of machine learning on economics. In Ajay Agrawal, Joshua Gans, and Avi Goldfarb, editors, *The economics of artificial intelligence: An agenda*, pages 507–547. University of Chicago Press Chicago, 2018.

John W Atkinson. Motivational determinants of risk-taking behavior. *Psychological Review*, 64(6p1):359, 1957.

Abhijit Banerjee, Shawn Cole, Esther Duflo, and Leigh Linden. Remedying education: Evidence from two randomized experiments in India. *The Quarterly Journal of Economics*, 122(3):1235–1264, 2007.

Abhijit Banerjee, Rukmini Banerji, James Berry, Esther Duflo, Harini Kannan, Shobhini Mukherji, Marc Shotland, and Michael Walton. Mainstreaming an effective intervention: Evidence from randomized evaluations of 'teaching at the right level' in India. *Technical report*, 2016.

Rukmini Banerji. Poverty and primary schooling: Field studies from Mumbai and Delhi. *Economic and political weekly*, pages 795–802, 2000.

BMBF. Verwaltungsvereinbarung digitalpakt schule 2019 bis 2024, 2019.

Jack W Brehm and Elizabeth A Self. The intensity of motivation. *Annual Review of Psychology*, 40(1):109–131, 1989.

George Bulman and Robert Fairlie. Technology and education: Computers, software, and the internet. In *Handbook of the Economics of Education*, pages 239–280. Elsevier, 2016.

Xieling Chen, Haoran Xie, Di Zou, and Gwo-Jen Hwang. Application and theory gaps during the rise of artificial intelligence in education. *Computers and Education: Artificial Intelligence*, 1:100002, 2020.

Ilja Cornelisz and Chris van Klaveren. Student engagement with computerized practising: Ability, task value, and difficulty perceptions. *Journal of Computer Assisted Learning*, 34(6):828–842, 2018.

Esther Duflo, Pascaline Dupas, and Michael Kremer. Peer effects, teacher incentives, and the impact of tracking: Evidence from a randomized evaluation in Kenya. *American Economic Review*, 101(5):1739–74, 2011.

Maya Escueta, Andre Joshua Nickow, Philip Oreopoulos, and Vincent Quan. Upgrading education with technology: Insights from experimental research. *Journal of Economic Literature*, 58(4):897–996, 2020.

Eric Hanushek. Education production functions. In *The Economics of Education*, pages 161–170. Elsevier, 2020.

Stuart A Karabenick and Zakhour I Youssef. Performance as a function of achievement motive level and perceived difficulty. *Journal of Personality and Social Psychology*, 10(4):414, 1968.

Tim Klausmann and Daniel Schunk. Shedding light on the effects of adaptive learning: A randomized field experiment in an online learning platform. unpublished, 2021.

Tim Klausmann, Marius Köppel, Daniel Schunk, and Isabell Zipperle. A field experiment on adaptive learning applying a machine learning algorithm. *AEA RCT Registry. May 26*, 2021. URL https://doi.org/10.1257/rct.7637.

Jung Lee and O Park. Adaptive instructional systems. In *Handbook of Research on Educational Communications and Technology*, pages 469–484. Citeseer, 2008.

Florence Martin, Yan Chen, Robert L Moore, and Carl D Westine. Systematic review of adaptive learning research designs, context, strategies, and technologies from 2009 to 2018. *Educational Technology Research and Development*, 68(4):1903–1929, 2020.

Karthik Muralidharan, Abhijeet Singh, and Alejandro Ganimian. Disrupting education? Experimental evidence on technology-aided instruction in India. *American Economic Review*, 109(4):1426–60, 2019.

Sandra Sampayo-Vargas, Chris Cope, Zhen He, and Graeme Byrne. The effectiveness of adaptive difficulty adjustments on students' motivation and learning in an educational computer game. *Computers and Education*, 69:452–462, 2013.

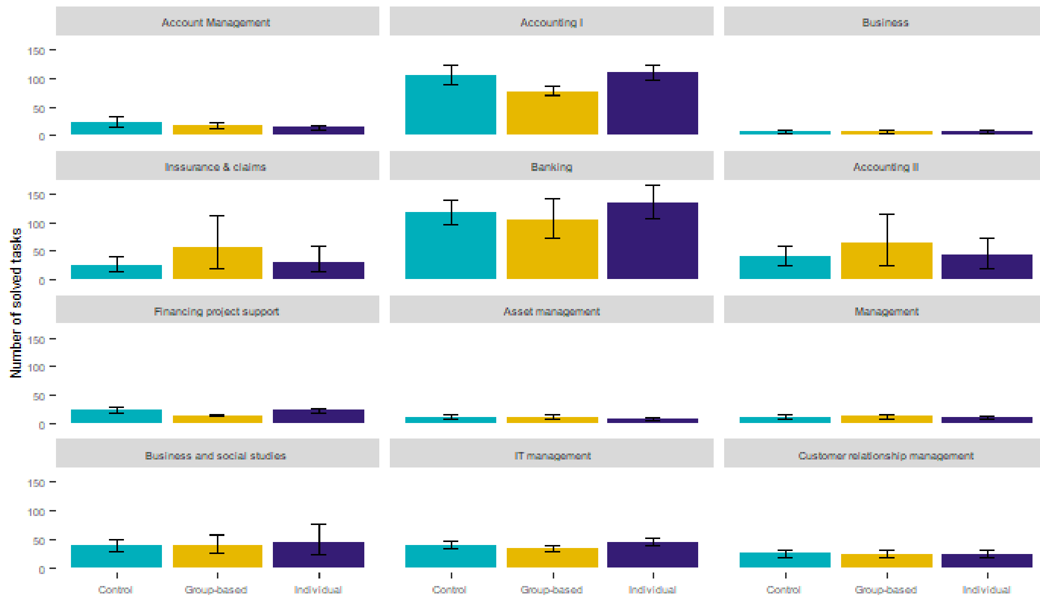
Chris Van Klaveren, Sebastiaan Vonk, and Ilja Cornelisz. The effect of adaptive versus static practicing on student learning-evidence from a randomized field experiment. *Economics of Education Review*, 58:175–187, 2017.

Stefanie Vanbecelaere, Katrien Van den Berghe, Frederik Cornillie, Delphine Sasanguie, Bert Reynvoet, and Fien Depaepe. The effectiveness of adaptive versus non-adaptive learning with digital educational games. *Journal of Computer Assisted Learning*, 36(4):502–513, 2020.

Mieke Vandewaetere, Piet Desmet, and Geraldine Clarebout. The contribution of learner characteristics in the development of computer-based adaptive learning environments. *Computers in Human Behavior*, 27(1): 118–130, 2011.

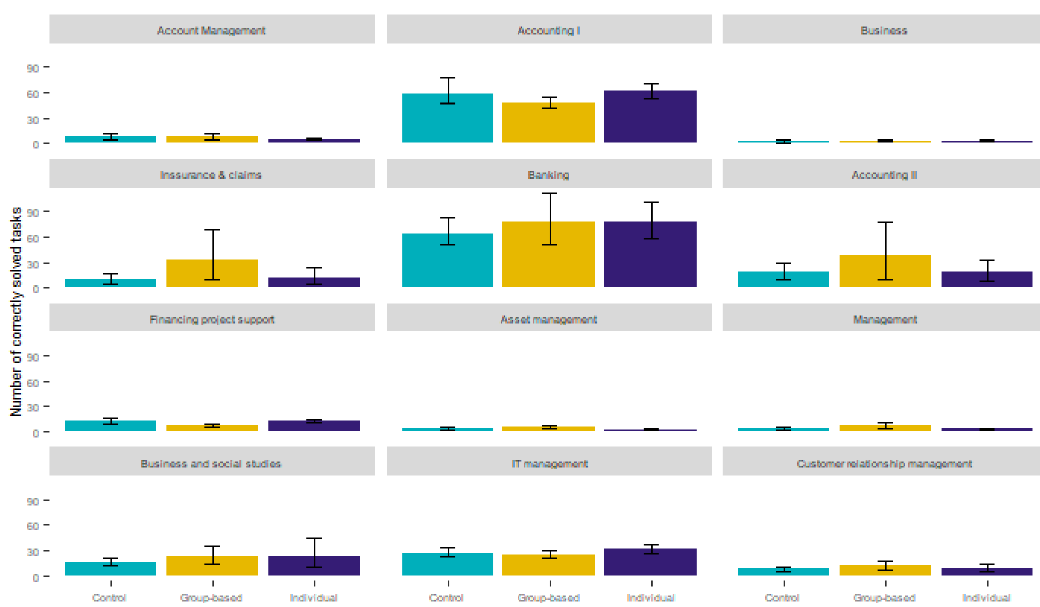
## Online Appendix

Figure A.1: Treatment effect of effort by subject



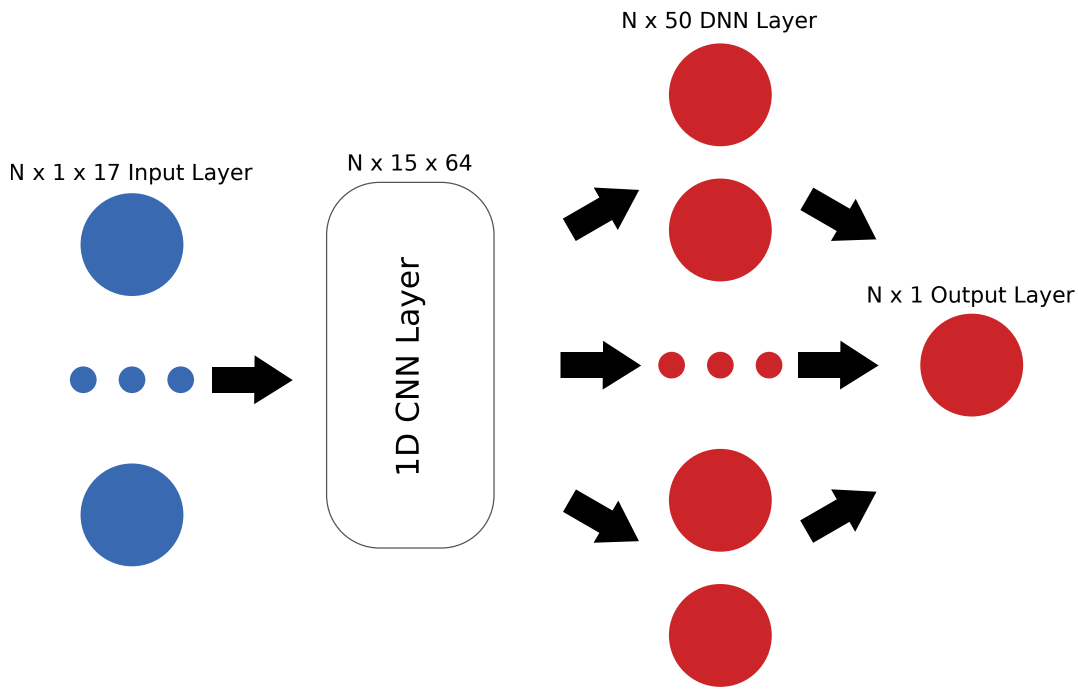

*Note:* If we split by subject the overall pattern of the treatment effect on effort replicates. The bars correspond to the control group and the two treatment groups. The y-axis depicts number of tasks that learners solved during the intervention. We report bootstrapped confidence intervals. The brackets display pairwise non-parametric Wilcoxon tests.

Figure A.2: Treatment effect of performance by subject




*Note:* If we split by subject the overall pattern of the treatment effect on performance replicates. The bars correspond to the control group and the two treatment groups. The y-axis depicts number of tasks that learners solved during the intervention. We report bootstrapped confidence intervals. The brackets display pairwise non-parametric Wilcoxon tests.

Figure A.3: Sketch of the used network architecture




*Note:* Best architecture used for the two convolutional networks. The convolutional layers are 1D layers which take an 1D input vector and map it to an 2D output vector. The N in the dimensions are the used batch size during training.

Table A.1: Overview of the hyperparameters used for training the two convolutional networks. The best overall performance is indicated in boldface. The permutation layer represents a fully connected layer with the identical number of neurons as the input size. The task of this layer is to reorder the input features before they are fed into the convolutional layers.

| Parameter | Values difficulty Network | Values effort Network |
|---|---|---|
| convolutional activation | **Relu** | **Relu** |
| dense activation | **Relu**, tanh | **Relu**, tanh |
| permutation layer | **false**, true | **false**, true |
| # convolutional layers | **1**..6 | **1**..6 |
| # convolutional filters | 16, 32, **64**, 128 | 16, 32, **64**, 128 |
| # kernel | 2, **3** | 2, **3** |
| # dense layers | **1**..3 | **1**..3 |
| # neurons per dense layer | 25, **50**, 100, 150 | 25, **50**, 100, 150 |
| $L_2$ weight regularization | **0.0**, 0.0001 | **0.0**, 0.0001 |
| cost function | **mean-squared-error** | **cross-entropy** |
| batch size | 50, 100, **200**, 500 | 50, 100, **200**, 500 |
| learning rate | **0.001** | **0.001** |
| learning rate decay rate | **1** | **1** |
| learning rate decay steps | **1000** | **1000** |
| optimizer | **Adam**, Nadam, SGD | **Adam**, Nadam, SGD |
| epoch | **10**, 50, 100 | **10**, 50, 100 |
| validation size | **0.0**, 0.1 | **0.0**, 0.1 |
| early stopping look-back | 3, 6, **no** | 3, 6, **no** |
| pre-processing | **quantile-transformer** | **quantile-transformer** |