



Application of reinforcement learning for the optimization of clinch joint characteristics

Christoph Zirngibl¹ · Fabian Dworschak¹ · Benjamin Schleich¹ · Sandro Wartzack¹

Received: 15 October 2021 / Accepted: 10 December 2021 / Published online: 22 December 2021
© The Author(s) 2021

Abstract

Due to increasing challenges in the area of lightweight design, the demand for time- and cost-effective joining technologies is steadily rising. For this, cold-forming processes provide a fast and environmentally friendly alternative to common joining methods, such as welding. However, to ensure a sufficient applicability in combination with a high reliability of the joint connection, not only the selection of a best-fitting process, but also the suitable dimensioning of the individual joint is crucial. Therefore, few studies already investigated the systematic analysis of clinched joints usually focusing on the optimization of particular tool geometries against shear and tensile loading. This mainly involved the application of a meta-model assisted genetic algorithm to define a solution space including Pareto optima with all efficient allocations. However, if the investigation of new process configurations (e. g. changing materials) is necessary, the earlier generated meta-models often reach their limits which can lead to a significantly loss of estimation quality. Thus, it is mainly required to repeat the time-consuming and resource-intensive data sampling process in combination with the following identification of best-fitting meta-modeling algorithms. As a solution to this problem, the combination of Deep and Reinforcement Learning provides high potentials for the determination of optimal solutions without taking labeled input data into consideration. Therefore, the training of an Agent aims not only to predict quality-relevant joint characteristics, but also at learning a policy of how to obtain them. As a result, the parameters of the deep neural networks are adapted to represent the effects of varying tool configurations on the target variables. This provides the definition of a novel approach to analyze and optimize clinch joint characteristics for certain use-case scenarios.

Keywords Mechanical joining process · Clinching · Data-driven Methods · Deep reinforcement learning

1 Introduction

Due to high potentials in reducing the weight of components, the integration of multi-material parts in the field of lightweight design is continuously increasing over the past years [1]. For this purpose, it is often necessary to join thin sheets of different materials (e. g. metals or polymers) and varying thickness ratios. Since these specifications involve a higher manufacturing complexity within the process chain, established thermal joining techniques (e. g. spot welding) often reach their limits. As a solution, the application of mechanical joining procedures offers robust methods to

create advanced lightweight designs by simultaneously requiring reduced joining forces and process times. Especially, the cold-forming process clinching enables the joining of two or more sheets based on a form- and force-fitting connection. Therefore, neither pre-treatments of the surfaces (e. g. the use of chemical substances) nor auxiliary components (e. g. rivets) are necessary to achieve reliable results which makes the process environmentally friendly and cost-efficient. Furthermore, compared to conventional metallurgical joining processes, clinching offers a robust opportunity to join dissimilar metals and varying sheet thicknesses which leads to a wide application in the automotive and appliance industry [2].

However, to reach a high quality of the resulting connection, not only the selection of a suitable joining method but also the strength improvement of the individual joints against shear and tensile loading is important. This mainly involves the adjustment and optimization of particular

✉ Christoph Zirngibl
zirngibl@mfk.fau.de

¹ Engineering Design (KTmfk) Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Martensstraße 9, Erlangen 91058, Germany

clinch tool and process parameters to maximize quality-relevant geometrical joint characteristics, such as the neck and interlock thickness. For this, the implementation of machine learning methods in combination with a genetic algorithm (GA) showed already a high applicability to determine a solution space involving a wide range of possible design alternatives [3]. Since this mainly requires the definition of a sufficient amount of training data, it is often time- and cost-intensive to setup powerful meta-models and thus to achieve a desired prediction accuracy. Therefore, previous contributions already investigated several machine learning methods combined with a varying amount of training data and clinch tool as well as process parameters. However, if the consideration of changing parameter settings (e. g. new material combinations) is required, the performance of the existing meta-models can drop to an insufficient level whereby the results are no longer reliable. Through this, it is often necessary to re-setup the resource- and time-intensive data sampling and meta-modeling process.

As a solution to this problem, the combination of Deep and Reinforcement Learning (RL) showed already high potentials in the identification of optimal solutions in completely different domains (e. g. [4, 5]). Thus, RL offers a promising and potentially applicable semi-supervised machine learning approach to determine clinching tool and process configurations without taking labeled input data into consideration. Therefore, the training of a Deep Reinforcement Agent aims not only at the prediction of joint characteristics, but additionally at learning a policy of how to obtain effects of varying tool configurations on the target variables. Based on this, the transfer and application of the pre-trained instead of a randomly initiated deep neural network to new use-cases (e. g. changing sheet thicknesses) provides the possibility for a immense decrease of the initial training and model-fitting effort. Additionally, the policy can be visualized as route through the solution space and, therefore, improve the understanding of the adaptation to varying clinching use-cases.

Although few contributions already introduced different ways for a data-driven analysis of mechanical joining processes, an approach for the application of a semi-supervised machine learning algorithm to identify optimal clinch tool configurations is not available yet. Motivated by this lack, this paper proposes a novel approach applying Deep and Reinforcement Learning for the optimization of clinch joint characteristics. The contribution is structured as follows. At the beginning, an overview about related works and the applied methods is given. Then, the setup and utilization of a Reinforcement Learner on a selected use-case scenario is explained and discussed in more detail. Concluding, the achieved results are summarized and an outlook provides further working steps.

2 Related work

Concerning the various opportunities for meta-models, the application of an adopted response surface methodology in combination with a Moving Least-Square approximation in [6, 7] enabled the identification of optimized clinching tools (punch and die). As a result, the joints' resistance against tensile loading were significantly improved.

Based on these results, the authors in [8] reached a further increase of the tensile strength (+10%, 623N to 834N) through the setup of response surfaces involving Kriging meta-models.

Roux and Bouchard [9] used an efficient global optimization algorithm in combination with Kriging meta-models to investigate the clinching process. Therefore, not only the identification of a optimal tool configuration but also the reduction of ductile damage effects in the material behavior had to be taken into consideration. As a result, the improvement of the joint against shear (+46.5%) and tensile (+13.5%) loading were achieved.

The application of an artificial neural network (ANN) in [10] enabled the estimation of the joint strength for varying tool configurations including an extensible die. For this, the setup of a Taguchi's L27 design of experiment (DoE) described a simulation plan for five design parameters divided over three levels. Based on the trained and validated ANN, the use of a meta-model assisted genetic algorithm offered the opportunity to define optimal clinching tool configurations considering a varying thickness of upper and lower sheet.

In comparison to this, Eshtayeh et al. [11] introduced a procedure for the use of a Taguchi-based Grey method including an analysis of variance (ANOVA) for the estimation of several clinch joint characteristics such as the neck, interlock and bottom thickness. For the achievement of meaningful results, the authors setup a Taguchi's L27 orthogonal array involving a notion of signal-to-noise (S/N) ratio. This enabled the determination of optimized joints based on the previous identification of impact values of the investigated geometrical tool parameters on the resulting neck, interlock and bottom thickness.

Wang et al. [12] used parameterized Bezier curves to describe and optimize clinching tool contours. For this reason, the direct communication between a finite element model and a genetic algorithm enabled the consistent transfer of data within the particular optimization steps. Based on this, it is possible to determine optimal tool contours through the connected adaption of shape control points and the measurement of the resulting joints' resistance against tensile loading.

The authors in [13] analyzed the impact of varying tool geometries as well as process parameters on the formation

of clinch joint characteristics (tensile force; neck and interlock thickness). Subsequently, the combination of the response surface method with a non-dominated sorting genetic algorithm (NSGA-II) enabled the definition of improved clinch joint parameter settings.

Besides the application of a genetic algorithm to maximize the interlock and neck thickness, Schwarz et al. [14] described a novel approach for the improvement of the clinching tool geometries by taking a principle component analysis (PCA) into consideration. For this, the identification of statistical eigenmodes and the setup of meta-models entirely based on the PCA provided a functional relationship between the detected joint contours and the particular input parameter sets. As a result, the performance of several analysis iterations enabled the determination of an optimal tool contour to obtain improved clinch joint properties.

In comparison to the data-driven analysis of clinch joint, few contributions investigated the application of machine learning algorithm, especially artificial neural networks, for the prediction of joint characteristics in the field of self-piercing riveting. For example, Oh and Kim et al. [16, 17] introduced a data-driven approach to estimate the cross-sectional shape of the punch forces' scalar input using supervised deep-learning algorithms (convolutional neural network and generative adversarial network). Therefore, the models enabled the generation of segmentation images of the particular cross-sections showing a high prediction accuracy of 92.22% and 91.95%. As a result, the geometrical shape of a self-piercing riveting process can be estimated for any material combination applying deep-learning algorithms and all relevant material properties. Furthermore, Karathanasopoulos et al. [18] investigated the ability to predict particular joint characteristics using neural network modeling. In this context, the authors setup and trained the machine-learning technique to successfully estimate the occurrence of joint formation for a chosen set of geometrical tool and process parameters.

In summary, the related contributions introduced a few procedures for the analysis and optimization of mechanical joining characteristics, such as the neck and interlock thickness. Therefore, the sampling of data in combination with the setup of an intelligent DoE mainly formed the basis for the training of machine-learning algorithms and the following fitting of meta-models as well as the application of genetic algorithms (e. g. NSGA-II). Thereby, a highly varying number of data and parameters were taken into account. As an example, Table 1 depicts an overview of applied DoEs and the considered amount of clinching factors and sampling resources.

While the application of a GA mainly defines a fitness function for a given scenario and a specific problem, the training of a Deep RL Agent aims not only at the estimation of target variables, but also at learning an optimal, or

Table 1 Overview of related works regarding their selected DoE and investigated number of samples and factors

Authors	DoE	$No._S$ ¹	$No._F$ ²
Oudjene et al. [6]	L_{18} Taguchi	18	8
Lambiase et al. [10]	L_{27} Taguchi	27	5
Eshtayeh et al. [11]	L_{27} Taguchi	27	11
Drossel et al. [15]	LHD ³	300	11
Wang et al. [12]	BBD ⁴	130	9

¹ $No._S$ = Number of samples

² $No._F$ = Number of factors

³ LHD = Latin Hypercube design

⁴ BBD = Box–Behnken design

nearly-optimal, policy of how to obtain them. Thus, it is possible to efficiently operate in a defined environment by choosing an optimal Agent's strategy for a given state in order to reach a specific goal. The resulting pre-trained deep neural network offers high potentials to reduce the initial training effort for the method adaption on new use-cases. Hence, this contribution aims to evaluate whether the setup of an optimization approach involving a Reinforcement Learner is feasible for the determination of optimal clinching tool and process configurations. Moreover, possible further steps for the consideration of versatile process chains (e. g. multiple sheet thickness and material combinations) are also identified and described in more detail.

3 Research questions

While several previous works considered a genetic algorithm (e. g. NSGA-II) in combination with labeled data, this contribution applies a Deep Reinforcement Learning algorithm to achieve the definition of an optimal clinching tool configuration based on a meta-model assisted data sampling process in order to answer three research questions (RQ). Following the methodical section and fitting of meta-models, the setup and training of an Agent answers the question whether Deep Reinforcement Learning is applicable for the identification of optimal clinch tool configurations taking a dimensionality of eight clinching parameters into consideration (RQ1). In comparison to a GA, Reinforcement Learning is expected to require a higher training effort to find optimal joint parameters for one clinch process. In order to estimate the difference, this contribution investigates how extensive the learning effort for RL would be to achieve a stable policy which continuously finds optimal solutions (RQ2). Subsequently, the capabilities and limitations are discussed to answer whether the application of Reinforcement Learning shows potentials for the design of clinch joint characteristics

in versatile process chains and which aspects are necessary to achieve a robust procedure (RQ3).

4 Methodical approach

To probe the feasibility of Reinforcement Learning for the design of a clinching procedure the subsequent experiment is conducted. First, multiple supervised meta-models are trained based on a limited number of clinch simulations to estimate the quality-relevant target variables neck, interlock and bottom thickness as well as the joining force. Second, the most suitable meta-model is chosen by R^2 . Third, the selected meta-models of neck and interlock are used as the meta-model assisted environment for the Reinforcement Learner. Fourth, the Reinforcement Learner is trained multiple times using multiple random seeds to initiate the initial state and the Deep Q-Networks (DQNs). Therefore, a DQN combines Reinforcement Learning with a class of artificial neural networks known as deep neural networks [5]. The initial state defines the starting and reset point of the Agent during the training. For the DQN the random seed, which is used to initialize a pseudorandom number generator, defines the random starting weights of the DQN. Accordingly, the impact of the initial state and the initial weights will be probed. Fifth, multiple indicators are presented to visualize the learning behavior. Sixth, the chosen meta-model is used for brute-force solving of the solution space. Solving every single state the global maxima of the reward is revealed and compared to the predictions of the Reinforcement Learner. Seventh, the determination of the best (top 20) designs in combination with the meta-model based estimation of the joining force and bottom thickness enables the designer to select an individual and most suitable tool and process configuration for a given use-case.

4.1 Numerical clinching process

Since the clinching process involves highly nonlinear deformations and large element distortions, the software LS-DYNA offers an appropriate environment for the generation of a finite element model. Therefore, the simulation model was validated in [19] showing a sufficient similarity in comparison to experimental microsections by analyzing the qualitatively relevant characteristics interlock, neck and bottom thickness as well as the sheet metal contour. Resulting, the FE model represents a 2D-axisymmetric structure including the following components: die, punch, blank holder, upper and lower sheet. The generation of the clinch joints base on a similar sheet combination involving the aluminium alloy EN AW-6014 temper T4, which is mainly used for automotive exterior parts, with a nominal thickness of each 2.0mm, a Poisson's ratio $\nu=0.3$ and a Young's modulus of 70 GPa.

Moreover, the implementation of an automatic (periodic) 2D r-adaptive remeshing method enables the generation of accurate simulation results. Besides the geometrical dimensions and the force of the blank holder, the joining velocity (2 mm S^{-1}) of the punch remains constant. The generated FE model does not consider the influence of material behavior and the impact of damage during the joining process. Additionally, the Coulomb friction law is used in all implemented contact formulations (tools / sheet metals, upper sheet / lower sheet).

Through the parameterized setup of the joining tool geometries (die and punch), based on [20], it is possible to automatically transfer data between the DoE dataset and the FEM. This ensures an efficient and consistent sampling process of several numerical clinch joints.

Then, based on this results, an algorithm automatically detects and determines quality-relevant geometrical (neck, interlock and bottom thickness; see Fig. 1) and process (joining force F_j) target variables [20]. As a summary, Fig. 2 shows a schematic illustration of the clinching process before and after the joining of the upper and lower sheet including all parameterized geometrical tool variables.

4.2 Definition of solution space and meta-modeling techniques

Since investigations in such a large solution space using a RL may lead to an unforeseeable number of required simulation runs, the clinching process is represented by a meta-model. However, this approach is only valid if the generated meta-model indicates a sufficiently high estimation accuracy ($R^2 \geq 0.8$) in order to predict the particular clinch joint characteristics (neck, interlock and bottom thickness; joining force). This enables the fast estimation of the joint characteristics for a given parameter set instead of running repeatedly a new numerical clinching simulation and thus provides an efficient and time-reducing training of the Agent. In addition, it is crucial to determine all relevant clinching geometrical and process parameters at the beginning. For this purpose, expert knowledge in combination with existing design principles were taken into consideration to define factors and

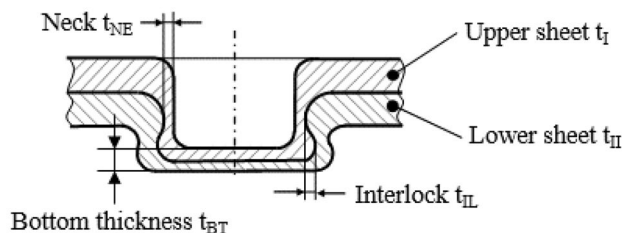


Fig. 1 Schematic illustration of the joints' geometrical characteristics based on [21]

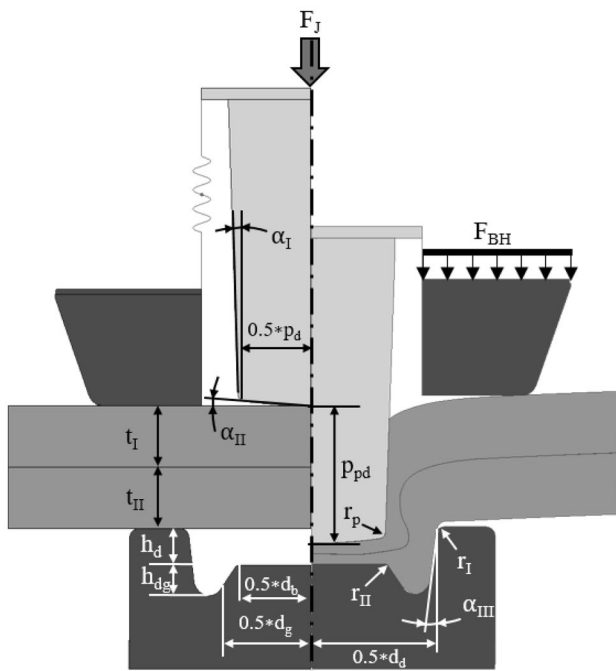


Fig. 2 Schematic illustration of the clinching process including relevant geometrical and process parameters

Table 2 Input parameters and the relating min./max.-spaces

Input parameter	Abbr.	Unit	Range	
			Min.	Max.
Punch diameter	d_p	mm	4.5	6.0
Punch corner radius	r_p	mm	0.1	0.6
Die diameter	d_d	mm	7.5	8.5
Die depth	h_d	mm	0.8	1.8
Die groove depth	h_{dg}	mm	0.5	1.3
Die bottom diameter	d_b	mm	3.5	4.8
Die groove diameter	d_g	mm	5.6	7.0
Punch penetration depth ¹	p_{pd}	mm	2.8	3.6

¹ corresponds to a thickness reduction of 70–90% percent of the total sheet thickness

suitable boundary conditions for the following generation of an intelligent design of experiment. Thus, Table 2 shows an overview of the selected geometrical and process parameters including the relating minimal and maximal ranges.

Since the setup of meta-models involving experimental studies can be highly time- and cost-intensive, the generation of the training database is entirely based on the previously introduced numerical clinching process. Furthermore, the utilization of an intelligent design of experiment (Latin Hypercube Design) provides a multi-dimensional distribution of near-randomly sample values including a space-filling design configuration. This enables, for instance, the

avoidance of spurious correlations within the n-dimensional factor space [22]. The resulting data set, including 400 design points and eight dimensions, provides the basis for the following fitting and training of several machine learning algorithms. Therefore, the approximation of relationships between the factors and the resulting clinch joint characteristics offers the opportunity to obtain a best-fitting meta-model. Thus, this contribution involves a Linear and Polynomial (2nd degree) regression model as well as a Support Vector Machine and an Ensemble Learner (Random Forest). Through this, different types of meta-modeling algorithms can be investigated based on their structural settings. For example, a Random Forest is composed of several Decision Trees while a Linear regression model bases on the fitting of a linear function. For the determination of a best-fitting meta-model a performance score (Coefficient of Determination R^2) enables a meaningful comparison of the particular algorithms.

4.3 Setup of a deep reinforcement learner

Following [23], the setup of a Reinforcement Learner comprises the environment and the Agent which communicate in a repetitive loop for every time step. Representing the meta-model assisted clinching simulation, the environment returns a interlock and neck thickness as feedback for given input parameters (see Table 2). In terms of Reinforcement Learning, a tuple of input parameters is referred to as a state of the environment. Interlock and neck thickness are equally important, but conflicting targets (increasing one factor leads to the decrease of the other parameter) rating the state with a reward. Therefore, an equally weighted sum is used to rate the rewards. To assure that both rewards are taken into account, the rewards are normalized using Pareto-Normalization. The individual rewards for neck and interlock are linear monotonically increasing functions as the goal is to maximize both values at the same time. To investigate the learned behavior, crossing zero is used as a benchmark. Therefore, a neck thickness of 0.53mm and an interlock of 0.36mm will be individually rewarded with 0. Values for neck and interlock lower than these will result in negative rewards. Thus, it becomes possible to check individual rewards as well as the total reward, which is the equally weighted sum over all individual rewards. In order to find highly rewarded states in the 8-dimensional solution space, the Agent is allowed to take actions by adjusting every input parameter by increasing, decreasing or keeping it the same. For this purpose, the individual rewards have been picked according to a policy that represents a path through the solution space leading from a given initial state to a desired state as a chain of actions based on the Agent’s decisions. Among other approaches, value-based Deep Q-Learning utilizes artificial neuronal networks (DQNs: evaluation and target

network) to choose actions for states [5]. Here, networks have an input neuron for each state parameter (e. g. 8), an output neuron for every possible action (e. g. 6561) referred as Q-Values and the identical architecture for hidden layer(s) (e. g. 1x256, fully connected). Applying a state to the evaluation network results in 6561 Q-Values among which the maximum is chosen as action for the given state. This is labeled as the green prediction loop in Fig. 3. Therefore, the schematic setup of the Deep Reinforcement Learner is based on [23].

Therefore, finding a suitable policy is equal to training the evaluation network for the given task. Weights are updated using experience tuples (state, action, reward, next state) from the history, which is referred as training and labeled red in Fig. 3. Here, a temporal difference method is used to calculate the loss between the Q-Values of the actual state ($Q(s_t)$) and the discounted, best next state ($\gamma \max Q(s_{t+1})$). First, the evaluation network is cloned into the target network by copying the weights. Second, the evaluation network is used to predict $Q(s_t)$ and the target network to predict $Q(s_{t+1})$. Third, the loss is calculated and used as objective for the stochastic gradient descent optimizer (e. g. ADAM) of the evaluation network's back propagation to update the weights.

In order to streamline the Agent's training, strategic decisions have to be made prior. First, as the used approach is dependent on the stored experiences, it has to be assured that the Agent is allowed to explore the solution space by randomly picking actions. If the Agent would exploit the DQN right from the start, the Agent would easily stick to optima close to the initial state. To balance this exploration a ϵ -Greedy-Strategy according to [24] is embedded.

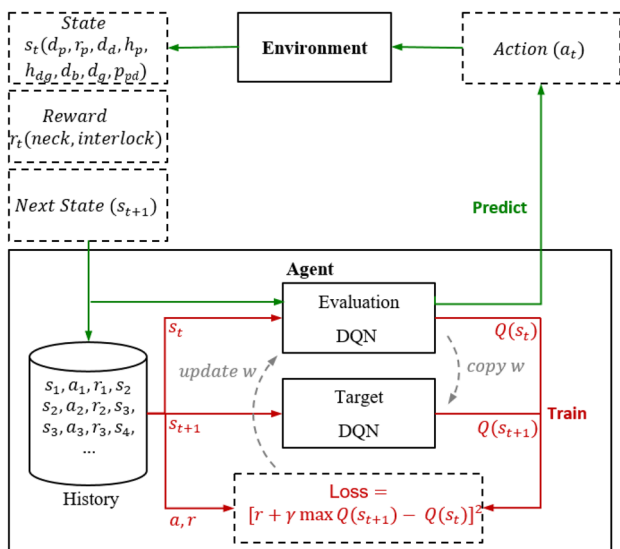


Fig. 3 Schematic setup of the Deep Reinforcement Learner for clinch simulations

Moreover, the Agent could get lost or stuck in the solution space by sticking to a local maximum. Therefore, a reset to the initial state is performed every 1000 time steps. Lastly, the training start is delayed by 300 time steps. Second, to reduce the impact of a single experience, memory batches are used in the training loop. Moreover, the update rate of the weights is regulated by the learning rate. Balancing learning rates can be a delicate task as a too high learning rate leads to instability of the training while a too low value leads to slow learning and a higher possibility to stick with sub-optimal solutions. Third, the discount rate models the foresightedness of the DQN. In summary, Table 3 states the used hyperparameters, which have been estimated from experience gathered from this and related use-cases.

5 Results

The subsequent sections describe the results of the preliminary meta-modeling procedure and the following implementation of a framework for the Reinforcement Learner. Firstly, the training as well as the estimation performance of different meta-modeling techniques is presented. Then, the setup and fitting of the Reinforcement Learner for the identification of an optimal parameter configuration is shown in more detail.

5.1 Data sampling and meta-modeling

The creation of the training database was carried out by the automated adaption and execution of the numerical clinching models and the subsequent measuring of the neck and interlock thickness based on [20].

The final size of training data involves 389 samples whereby eleven design points had to be removed (loss ratio of 2.75%) due to design failure or error terminations. Afterwards, the setup and fitting of the machine learning methods enabled the definition of a best-fitting technique according on the achieved mean R^2 values and the relating standard deviations. Therefore, Fig. 4 depicts the algorithms and performance scores.

Table 3 Summary of used hyperparameters

Name	Symbol	Value
Learning rate	α	0.0005
Discount rate	γ	0.8
Decay rate	ϵ	0.1
Batch size		128
Train Start		300
Time Steps		1000
Episodes		200

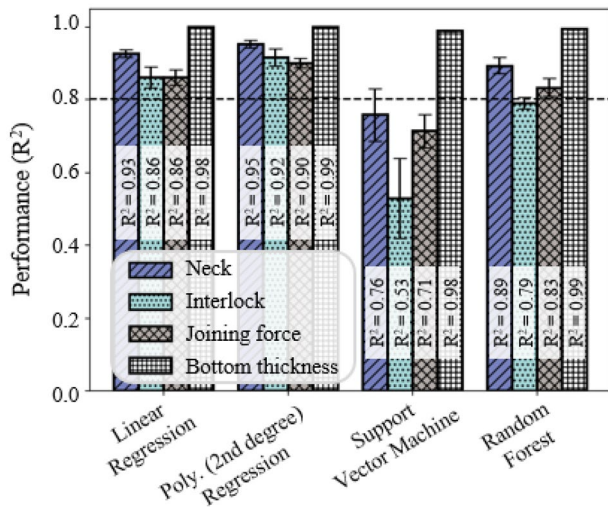


Fig. 4 Overview of the applied machine learning algorithms and the resulting R^2 performances

One can see, that the Linear as well as the Polynomial (2nd degree) regression model reached the highest R^2 values for the estimation of all investigated target variables whereas the nonlinear model indicated a slightly higher accuracy in combination with a lower standard deviation. Furthermore, while the application of the Ensemble Learner (Random Forest) showed a sufficient performance, the Support Vector Machine achieved only poor results especially for the prediction of the interlock thickness. Thus, the pre-trained and best-fitted polynomial (2nd degree) meta-models of neck and interlock represent the clinching process for the setup of the Reinforcement Learner. Furthermore, due to highly demonstrated prediction accuracies, the subsequent estimation of the bottom thickness and joining force can also be carried out by fitting the quadratic regression model. Resulting, a sufficient and detailed description of the clinching process is provided in the following steps.

5.2 Results of the reinforcement learner

Figures 5 and 6 visualize the gained, individual reward for interlock and neck thickness over the time steps. The benchmarks of 0.53mm for neck thickness and 0.36mm for interlock are indicated as a dashed grey line. One can see, that both benchmarks are reached after approximately 100,000 time steps independent of the initial clinching parameters and the weights of the DQNs caused by the random seed. Regardless of this, each experiment was run 200,000 steps because the number of episodes and steps must be defined before the start and it is not clear from the outset that only 100,000 steps are needed.

Figure 7 demonstrates the superordinate objective of the Reinforcement Learner which is the total reward

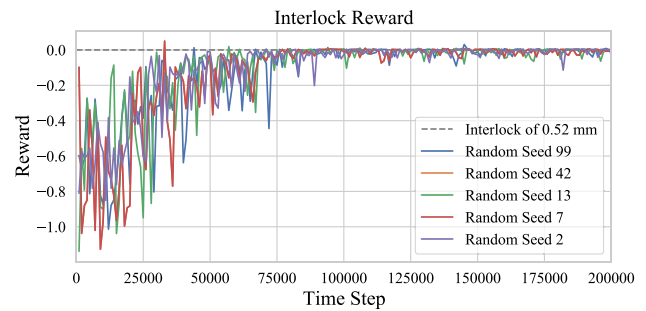


Fig. 5 Illustration of the Interlock Reward as first, individual reward for each time step and Random Seed

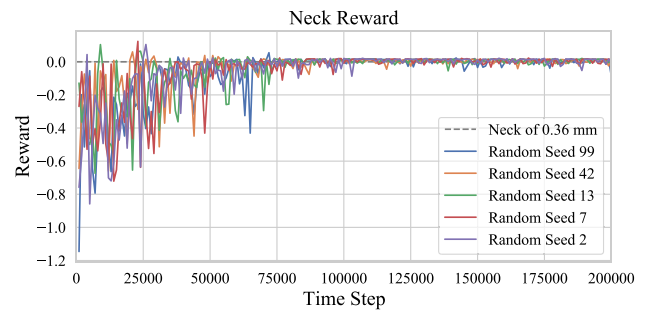


Fig. 6 Illustration of the Neck Reward as second, individual reward for each time step and Random Seed



Fig. 7 Illustration of the Total Reward for each time step and Random Seed

as a equally weighted sum over the individual rewards. As well as for the individual rewards, the best possible reward of 0.013, which is indicated as a dashed grey line, is reached after 100,000 iterations. To estimate the best possible reward, the complete solution space was solved using the meta-model. The top 20 combination including the relating clinching parameters can be found in appendix 1. In this context, the first two design combinations (bold typed) achieved positive rewards for both target variables and thus can be considered as the best optimization results. There, 0.0013 represents the global maximum of

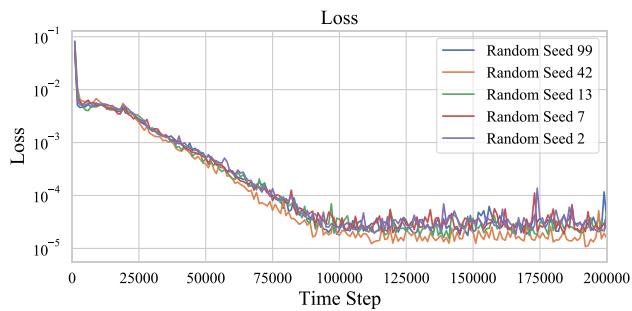


Fig. 8 Illustration of the Loss for each time step and Random Seed

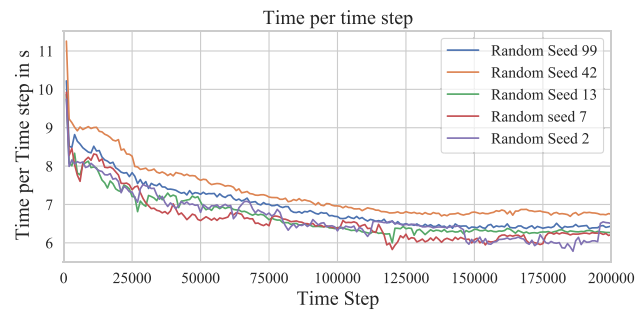


Fig. 10 Illustration of the time per time step

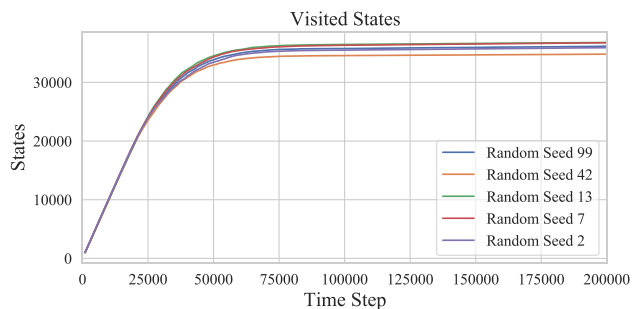


Fig. 9 Illustration of the Visited States over all calculated time steps

the solution space of the possible rewards. The additional estimation of the joining force and bottom thickness provides the opportunity to identify the most suitable solution for the given use-case. For example, while the joining forces varied strongly (range: 41.7–64.3 kN), the bottom thicknesses showed only slightly different values (range: 0.5154–0.5485 mm).

Regarding the results of the Reinforcement Learner, Fig. 8 illustrates the loss as indicator for the update rate of the DQN in logarithmic scaling. One can see, that even in the logarithmic scaling the loss rapidly drops until 100,000 iterations. Afterwards only minimal adaptations in the range of 1×10^{-4} to 1×10^{-5} occur. Figure 9 counts the visited states which indicate the number of unique simulations a Reinforcement Learner needs in

order to find the policy. Independent of the random seed the Reinforcement Learner probes approximately 35,000 unique states. Lastly, Fig. 10 illustrates the time per time step which is necessary for one training loop of the Reinforcement Learner. Note that the simulation time is not included here. Also note that the time is hardware depended and should only provide an assessment of the loop time using the GPU of an ordinary desktop computer (NVIDIA Quadro M2000).

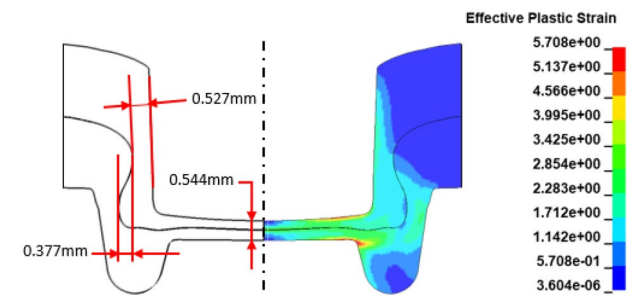


Fig. 11 Overview of resulting clinch joint and the particular quality-relevant geometrical characteristics

6 Discussion

Reflecting RQ1, Fig. 7 illustrates that the Reinforcement Learner converges to a reward of 0.0 which has been identified as a benchmark for clinch tool parameters that simultaneously result in a neck thickness of 0.53mm and an interlock of 0.36mm. The individual rewards demonstrated in Fig. 5 and 6 emphasize that both objectives are equally considered by the Reinforcement Learner. In combination with the completely solved solution space this indicates that the Reinforcement Learner has achieved a policy that continuously leads to good or even the best possible solution in the large 8-dimensional solution space after 100,000 time steps independent of the initial state and the initial weights of the DQN caused by the various random seeds. Later peaks in the reward most likely occur due to the reset of Learner every 1000 time steps, which places it into the initial state causing lower rewards. This is also indicated by Fig. 8 which demonstrates the marginal adaptation of the DQN after 100,000 time steps. Concerning the top20 results in appendix 1, the optimal solution can be achieved for the parameter combination: $d_p=6.0$ mm, $r_p=0.3$ mm, $d_d=8.5$ mm, $h_d=1.0$ mm, $h_{dg}=1.5$ mm, $d_b=4.8$ mm, $d_g=5.6$ mm and $p_{pd}=3.6$ mm. Therefore, Fig. 11 gives an overview of the relating clinch joint and the particular

quality-relevant geometrical characteristics. One can see, that the prediction of the trained Reinforcement Learner and the actual numerical results show nearly similar effects (neck: +0.013 mm, interlock: +0.004 mm).

Thus, the Reinforcement Learner is feasible to find near optimal or optimal tool parameters in this large solution space for a clinching procedure and the application is robust as the initial state and initial weights of the DQN do not inflict any results. Additionally, the meta-model-based estimation of the bottom thickness differs only slightly in comparison to the generated clinch joint geometry (-0.003mm).

Referring to RQ2, the learning effort can be estimated observing Figs. 9 and 10. As one can see from Fig. 10, the update time for the DQN is negligible compared to the necessary time for one clinch simulation (approximately 15 minutes). As a new and unique state requires one clinch simulation, the primer indicator for the learning effort is the number of visited states which is illustrated in Fig. 9. Independent from initial state and weights of the DQN, the Reinforcement Learner requires approximately 35,000 individual states to converge to a suitable policy. Accordingly, it becomes obvious that the Reinforcement Learner need a tremendous amount of clinch simulations and the training effort exceeds a genetic algorithm by far. In addition to that, tuning the hyperparameters is not included in this estimation. The used hyperparameters are derived from previous investigations and experiences made in preceding case studies. Therefore, it must be admitted that the actual effort for a RL for one clinch simulation would be even higher.

Leading over to RQ3, potentials of Reinforcement Learning for versatile process chains emerge from the transferability of the achieved policy. Unlike genetic algorithms, the outcome of the Reinforcement Learner is a trained DQN which can be easily used as pre-trained setup for consecutive tasks by replacing the randomly instantiated weights of the DQN with the previously trained weights. As the DQN predicts actions for given states which are clinching tool configurations, similar states are likely to occur in closely related clinch processes. Moreover, actions which are modifications of the tool configurations might be valid as well. To justify the extensive effort, the Reinforcement Learner needs to be general applicable for multiple clinching processes in terms of sheet thickness and material combinations. Therefore, state parameters for the description of the sheet metals should be included. This might allow the Reinforcement Learner to notice and react to new combinations more appropriately by fine-tuning the pre-trained weights. In addition to that, a related approach for chip design by Mirhoseini et al. [25] has shown that Reinforcement Learning can lead to versatile process chains by using the same DQN for multiple chip placement procedures. The paper also emphasizes the significance of state augmentation, meaning the adding of information to the environment's state parameters. This

leads from the here presented pure value-based to a model-based approach by adding supplemental dimensions to the state parameters. For example relating information about the tool (die or punch) or the type (radius, diameter, depth) to the value of the state parameter. This might enable the DQN to gain intrinsic insights about the clinching procedure and improves the generalizability of the DQNs further.

7 Conclusion and outlook

Summary: The presented contribution introduced a novel approach for the data-driven identification of optimal clinching tool and process configurations using Reinforcement Learning. Therefore, the investigation of an equal sheet metal combination (material: EN AW-6014 T4, thickness: each 2.0mm) served as a representative example for the generation of clinch joints. Furthermore, since the application of a suitable policy mainly requires a high amount of data, the training of an Agent is assisted by a previously generated meta-model. For this purpose, the setup of a parameterized FEM in combination with the selection of an intelligent DoE enabled the automated generation of a database and thus the following selection of a best-fitting meta-modeling technique. Then, in order to identify an optimal clinch tool and process configuration in the 8-dimensional solution space, a value-based Deep Q-Learning algorithm is used to utilize artificial neural networks and to learn the value of a realized action in a specific state. In summary, the setup and use of a Reinforcement Learner identified the subsequent results:

- It is possible to train a Reinforcement Learner for the identification of an optimal clinch tool and process configuration in a 8-dimensional solution space considering a required neck thickness of 0.53mm and an interlock of 0.36mm.
- However, the Reinforcement Learner currently needs a very high amount of input clinching simulations (35,000 individual states) to converge to a suitable policy which exceeds the training effort of a genetic algorithm by far.
- But, unlike a genetic algorithm, the trained policy has the potential to be applied as a pre-trained Reinforcement Learner for new clinching setups (similar material and sheet thickness combination) with changing requirements on the neck and interlock thickness by replacing the randomly instantiated weights of the DQN with the previously trained weights.

Outlook: To get a higher applicability of the Reinforcement Learner both the ability to describe multi-material joints and the clear reduction of the training effort have

to be improved in future works. Therefore, the inclusion of further parameters to the environment, e. g. sheet metal data such as the ultimate tensile strength or the thickness of lower and upper sheet, provides the opportunity to consider dissimilar material combinations by fine-tuning the pre-trained weights. Moreover, the implementation of the joining force and bottom thickness in the environment of the Reinforcement Learner will provide the opportunity to identify optimal design combinations for specific manufacturing conditions.

Additionally, the adding of supplemental dimensions to the environments' state parameters leads to a changeover from a value-based to a model-based approach and thus enables to increase the transferability from one clinch process to consecutive ones.

Acknowledgements This work was Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research 635 Foundation) "TRR 285 B05" Project-ID 418701707.

Author Contributions C. Zirngibl: Conceptualization, Methodology, FE-Simulation, Formal analysis, Investigation, Writing - Original Draft, Visualization, F. Dworschak: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing - Original Draft, Visualization, B. Schleich: Conceptualization, Methodology, Formal analysis, Writing - Review & Editing, Supervision, S. Wartzack: Conceptualization, Writing - Review & Editing, Supervision, Project administration, Funding acquisition.

Funding Open Access funding enabled and organized by Projekt DEAL.

Declarations

Conflict of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

A Top20 Results

Index	d_p	r_p	d_d	h_d	h_{dg}	d_b	d_g	p_{pd}	t_{IL} [mm]	t_{NE} [mm]	reward _{IL}	reward _{NE}	Reward total	F_J [N]	t_{BT} [mm]
873874	6	0.3	8.5	1	1.5	4.8	5.6	3.6	0.36386	0.53088	0.011005	0.002786	0.013791	60349.10	0.5410
581374	5.5	0.2	8.5	1	1.5	4.8	5.6	3.6	0.36763	0.52209	0.002118	0.005504	0.007623	50428.12	0.5282
716374	5.75	0.2	8.5	1	1.5	4.8	5.6	3.6	0.381165	0.517199	-0.008494	0.015267	0.006773	56177.15	0.5356
716379	5.75	0.2	8.5	1	1.5	4.8	5.88	3.6	0.368199	0.519692	-0.000933	0.005914	0.004981	56181.66	0.5361
288874	5	0.1	8.5	1	1.5	4.8	5.6	3.6	0.368226	0.519233	-0.002326	0.005934	0.003609	42253.12	0.5154
851374	6	0.2	8.5	1	1.5	4.8	5.6	3.6	0.393652	0.513074	-0.021002	0.024275	0.003273	62816.24	0.5430
851379	6	0.2	8.5	1	1.5	4.8	5.88	3.6	0.380445	0.515848	-0.01259	0.014749	0.002159	62987.88	0.5434
851194	6	0.2	8.5	1	1.25	4.8	5.6	3.6	0.357533	0.526994	0.00707	-0.005339	0.001731	64399.76	0.5473
851384	6	0.2	8.5	1	1.5	4.8	6.16	3.6	0.368402	0.51789	-0.006398	0.006061	-0.000337	63369.86	0.5439
869379	6	0.3	8.25	1	1.5	4.8	5.88	3.6	0.365435	0.518286	-0.005199	0.003921	-0.001278	64239.81	0.5462
887374	6	0.4	8	1	1.5	4.8	5.6	3.6	0.363171	0.518553	-0.004387	0.002287	-0.0021	63672.57	0.5485
869374	6	0.3	8.25	1	1.5	4.8	5.6	3.6	0.380708	0.514269	-0.017378	0.014938	-0.00244	63718.69	0.5458
734374	5.75	0.3	8.25	1	1.5	4.8	5.6	3.6	0.370441	0.516237	-0.01141	0.007532	-0.003878	57296.97	0.5384
738874	5.75	0.3	8.5	1	1.5	4.8	5.6	3.6	0.351167	0.534642	0.014801	-0.019116	-0.004315	54142.38	0.5337
599374	5.5	0.3	8.25	1	1.5	4.8	5.6	3.6	0.359127	0.518978	-0.003101	-0.001888	-0.004989	51765.32	0.5310
423874	5.25	0.1	8.5	1	1.5	4.8	5.6	3.6	0.383647	0.512422	-0.02298	0.017058	-0.005922	46654.40	0.5228
716384	5.75	0.2	8.5	1	1.5	4.8	6.16	3.6	0.356396	0.521453	0.001469	-0.0078	-0.00633	56396.50	0.5365
288879	5	0.1	8.5	1	1.5	4.8	5.88	3.6	0.356859	0.520211	0.000214	-0.006798	-0.006585	41669.89	0.5159
581379	5.5	0.2	8.5	1	1.5	4.8	5.88	3.6	0.354905	0.524308	0.004355	-0.011026	-0.006671	50265.50	0.5287
446374	5.25	0.2	8.5	1	1.5	4.8	5.6	3.6	0.353049	0.527765	0.007849	-0.015044	-0.007195	45569.15	0.5208

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Gude M, Meschut G, Liberwirth H, Zäh H et al. (2015) FOREL-Studie - Chancen und Herausforderungen im ressourceneffizienten Leichtbau für die Elektromobilität. Dresden, ISBN 978-3-00-049681-3
- Feldmann K, Schöppner V, Spur G (2014) Handbuch Fügen, Handhaben. München, Carl Hanser Verlag, Montieren. 978-3-446-42827-0
- In: Zirngibl C, Schleich B, Wartzack S (2020) Potentiale datengestützter Methoden zur Gestaltung und Optimierung mechanischer Fügeverbindungen. In: Proceedings of the Symposium DfX 31:71–80. <https://doi.org/10.35199/dfx2020.8>
- Silver D, et al. (2016) Mastering the game of go with deep neural networks and tree search. *Nature* 529 (7587). <https://doi.org/10.1038/nature16961>
- Mnih V et al (2015) Human-level control through deep Reinforcement learning. *Nature* 518(7540):529–533. <https://doi.org/10.1038/nature14236>
- Oudjene M, Ben-Ayed L (2008) On the parametrical study of clinch joining of metallic sheets using the Taguchi method. *Eng Struct* 30(6):1782–1788. <https://doi.org/10.1016/j.engstruct.2007.10.017>
- Oudjene M, Ben-Ayed L, Delamézière A, Batoz J-L (2009) Shape optimization of clinching tools using the response surface methodology with moving least-square approximation. *J Mater Process Technol* 209(1):289–296. <https://doi.org/10.1016/j.jmatprotec.2008.02.030>
- Lebaal N, Oudjene M, Roth S (2012) The optimal design of sheet metal forming processes: application to the clinching of thin sheets. *Int J Comput Appl Technol* 43(2):110–116. <https://doi.org/10.1504/IJCAT.2012.046041>
- Roux E, Bouchard P-O (2013) Kriging metamodel global optimization of clinching joining processes accounting for ductile damage. *J Mater Process Technol* 213(7):1038–1047. <https://doi.org/10.1016/j.jmatprotec.2013.01.018>
- Lambiase F, Di Ilio A (2013) Optimization of the clinching tools by means of integrated FE modeling and artificial intelligence techniques. *Procedia CIRP* 12:163–168. <https://doi.org/10.1016/j.procir.2013.09.029>
- Eshtayah M, Hrairi M (2016) Multi objective optimization of clinching joints quality using grey-based taguchi method. *Int J Adv Manuf Technol* 87(1–4):1–17. <https://doi.org/10.1007/s00170-016-8471-1>
- Wang M, G-q X, Li Z, Wang J (2017) Shape optimization methodology of clinching tools based on Bezier curve. *Int J Adv Manuf Technol* 24(1):2267–2280. <https://doi.org/10.1007/s00170-017-0987-5>
- Wang X, Li X, Shen Z, Ma Y, Liu H (2018) Finite element simulation on investigations, modeling, and multiobjective optimization for lynch joining process design accounting for process parameters and design constraints. *Int J Adv Manuf Technol* 96:3481–3501. <https://doi.org/10.1007/s00170-018-1708-4>
- Schwarz C, Kropp T, Kraus C, Drossel W-G (2020) Optimization of thick sheet clinching tools using principal component analysis. *Int J Adv Manuf Technol* 106:471–479. <https://doi.org/10.1007/s00170-019-04512-5>
- Drossel WG, Israel M (2013) Sensitivitätsanalyse und Robustheitsbewertung beim mechanischen Fügen. EFB-FB Nr. 323, EFB e.V., Hannover. ISBN: 978-3-86776-419-3
- Oh S, Kim HK, Jeong T-E, Kam D-H, Ki H (2020) Deep-Learning-Based Predictive Architectures for Self-Piercing Riveting Process. *IEEE*. <https://doi.org/10.1109/ACCESS.2020.3004337>
- Kim HK, Oh S, Cho K-H, Kam D-H, Ki H (2021) Deep-Learning Approach to the Self-Piercing Riveting of Various Combinations of Steel and Aluminum Sheets. *IEEE*. <https://doi.org/10.1109/ACCESS.2021.3084296>
- Karathanasopoulos N, Pandya KS, Mohr D (2021) Self-piercing riveting process: Prediction of joint characteristics through finite element and neural network modeling. *J Adv Join Process*. <https://doi.org/10.1016/j.jajp.2020.100040>
- Bielak ChR, Böhnke M, Beck R, Bobbert M, Meschut G (2021) Numerical analysis of the robustness of clinching process considering the pre-forming of the parts. *J Adv Joining Processes* 3. <https://doi.org/10.1016/j.jajp.2020.100038>
- Zirngibl C, Schleich B, Wartzack S (2021) Approach for the automated and data-based design of mechanical joints. *Proc Des Soc* 1:521–530. <https://doi.org/10.1017/pds.2021.52>
- DVS-EFB 3420:2021-04, Clinching – basics
- Siebertz K, van Bebber D, Hochkirchen T (2017) *Statistische Versuchsplanung - Design of Experiments (DoE)*. (Wiesbaden: Springer), <https://doi.org/10.1007/978-3-662-55743-3>.
- Sutton R and Barto A (2018) *Reinforcement learning: An introduction*. MIT press (2). ISBN: 978-0-26203-924-6
- Tokic M (2010) Adaptive ϵ -Greedy Exploration in Reinforcement Learning Based on Value Differences. In: Dillmann R., Beyerer J., Hanebeck U.D., Schultz T. (eds) *KI 2010: Advances in Artificial Intelligence*. KI 2010. Lecture Notes in Computer Science, vol 6359. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-16111-7_23
- Mirhoseini A, Goldie A, Yazgan M et al (2021) A graph placement methodology for fast chip design. *Nature* 594:207–212. <https://doi.org/10.1038/s41586-021-03544-w>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.