Journal of Manufacturing Processes 73 (2022) 395–408

Contents lists available at ScienceDirect

Journal of Manufacturing Processes

journal homepage: www.elsevier.com/locate/manpro

Transfer learning with artificial neural networks between injection molding processes and different polymer materials

Yannik Lockner [*,1], Christian Hopmann [1], Weibo Zhao [1]

Institute for Plastics Processing (IKV), RWTH Aachen University, Seffenter Weg 201, 52074 Aachen, Germany

ARTICLE INFO

Keywords:
Artificial neural networks
Injection molding
Transfer learning
Sparse data
Material variation
Process setup

ABSTRACT

Finding appropriate machine setting parameters in injection molding remains a difficult task due to the highly nonlinear process behavior. Artificial neural networks are a well-suited machine learning method for modelling injection molding processes, however, it is costly and therefore industrially unattractive to generate a sufficient amount of process samples for model training. Therefore, transfer learning is proposed as an approach to reuse already collected data from different processes to supplement a small training data set. Process simulations for the same part and 60 different materials of 6 different polymer classes are generated by design of experiments. After feature selection and hyperparameter optimization, finetuning as transfer learning technique is proposed to adapt from one or more polymer classes to an unknown one. The results illustrate a higher model quality for small datasets and selective higher asymptotes for the transfer learning approach in comparison with the base approach.

1. Introduction

In high-wage countries, rising customer demands and increasingly smaller tolerances for product quality provoke the need for technology innovation of manufacturing processes for businesses in order to prevail on the international market [1]. In the plastics processing industry, especially for the injection molding process, the reduction of production costs for the maximization of profit margins is a crucial factor for many companies. Subject to discussion for already a long time is the setup of an injection molding process, i. e. the most effective search for suitable machine setting parameters during the consecutive production process. While manufacturers commonly use expert knowledge by their employees to setup a new process with a subjectively optimized choice of machine setting parameter values [2], other enterprises recognize benefits of a data-based process model and analysis.

Machine learning methods and modelling techniques, such as artificial neural networks (*ANNs*), have provided evidence of their capability to help solve various tasks in the industrial sector, including quality prediction and optimization, error diagnosis and scheduling [3–6]. ANNs are especially applicable for modelling tasks that involve nonlinear relations such as the thermoplastic injection molding process with its complex visco-elastic material behavior and nonlinear relationships between several quality, process and machine parameters [7,8]. Once fitted to a specific process, the resulting surrogate model can be exploited by a subsequent algorithm. Given optimal values for the quality parameters of the model, i. e. parameters of the produced part of the modelled injection molding process, the algorithms vary the model's input parameters in order reproduce the optimal values as closely as possible. Evolutionary algorithms, such as genetic algorithms or particle-swarm algorithms are frequently employed in industrial use cases [9,10]. This two-step optimization process offers an objective and standardized method for a process setup which can substantially support shop-floor and production personnel for manufacturing companies.

A commonly invoked impediment still to overcome before a broad attractiveness for practical implementation can be raised is the required amount of training data, i. e. process data recordings of different settings. As the model's prediction quality is depending on the amount of data samples available and the generation of data samples takes machine time, material, energy and educated personnel, companies rightly scrutinize the advantages of machine learning over expert knowledge, e. g. for the use case injection molding process setup [11]. A possible mitigation of this obstacle is so-called *transfer learning* (**TL**): Data from already known processes, which are particularly similar to the new process, are used to substitute data of the new process and therefore

* Corresponding author.
E-mail address: yannik.lockner@ikv.rwth-aachen.de (Y. Lockner).
[1] IKV Institute for Plastics Processing (IKV) in Industry and Craft at RWTH Aachen University

https://doi.org/10.1016/j.jmapro.2021.11.014
Received 7 May 2021; Received in revised form 7 October 2021; Accepted 11 November 2021
Available online 23 November 2021

reduce the costs for data generation. The degree of data availability is expected to benefit from recent years' development in the field of "plastics industry 4.0", offering a multitude of solutions for data acquisition in production [12]. However, structural process influences in injection molding are manifold [13] and an omnipotent artificial intelligence (AI) for the injection molding, therefore, requires an unavailable amount of data. A possible step in-between can be the development of strategies for machine-bound models, which can easily transfer knowledge between processes of varying geometries and materials to accelerate a process setup in injection molding.

While an evaluation of transfer learning for different geometries and a simulation database has been conducted successfully [14], this paper focuses on the evaluation of transfer learning for the modelling of injection molding processes with a single part geometry but varying material as a further influence on the injection molding process. Section 2 defines the state of the art for usage of artificial neural networks in the plastics processing industry. Furthermore, the concept of transfer learning is further introduced and preliminaries are explained. Section 3 describes the part geometry and materials used for the simulations, as well as the data sampling strategy and describes the specific transfer learning approach for the injection molding process. In Section 4, the results of a hyperparameter optimization and the transfer learning experiments are depicted and discussed and general implications are deduced and interpreted. The paper concludes in Section 5 and offers possible prospects for the advancements of injection molding by transfer learning.

2. State of the art

ANNs can predict conditional parameters from a set of provided values of input parameters by internal processing (forward propagation). This requires an a-priori training of the model to a conditional probability, represented by the provided training samples. During this training, the weighted discrepancy between the model's prediction and the true value of the presented data sample is propagated back through the ANN's layered structure to adapt the neuron weights and minimize the prediction error. This concept of backpropagation has been developed by *Rumelhart* et al. with the help of gradient descent [15].

The model quality after training, therefore, is unavoidably depending on the quality of data, i. e. in terms of accuracy regarding the measured values, but also highly sensitive to the choice of hyperparameters. All characteristics of a neural networks and the corresponding training process can be regarded as hyperparameters, whereas some are commonly included into a hyperparameter search, i. e. a sequential variation of hyperparameter values to determine their suitability for the model and the dataset. Examples are activation functions, optimizers, amount of hidden layers and their respective cardinal number of neurons, neuron weight initialization strategies or the maximal amount of epochs [16]. Some of these are used in this paper for a hyperparameter search.

2.1. Applications of machine learning in injection molding

Tsai and *Luo* implemented a combination of an artificial neural network and a genetic algorithm to minimize the warpage of injection molded optical lenses [17]. Mold temperature, cooling time, packing pressure and packing time were identified as significant influencing parameters of the part quality by ANOVA and used as input parameters to the ANN. The proposed approach was able to improve the form accuracy in experimental trials by 13.36%.

Iniesta, *Alcaraz* and *Borbón* combined a modelling unit with an optimization unit to determine optimized injection molding machine parameters [18]. As an optimization unit served an artificial neural network with melt temperature, mold temperature, packing pressure, packing time and cooling time as input and the warpage of part dimensions as quality parameter. After 10.000 training epochs, the training loss with mean-squared error as loss function was lower than 10^{-6}.The required training data was generated by the simulation software Moldflow, Autodesk Inc., San Rafael, CA, USA. The optimization unit was an artificial bee colony that exploited the ANN, determining the set of input parameters that resulted in minimized warpage. This combined approach was found to be superior to the optimization tool included in Moldflow.

Zhang et al. used an ANN to map the relationships between machine setting parameters and quality parameters, in this case the warpage and the clamping force of the mold [19]. The motivation for the second quality parameter is the use of a smaller injection molding machine by optimizing the clamping force, which can lead to significant cost savings and prevent scheduling bottlenecks. It has been shown that the parameters required for predicting both warpage and clamping force the ANN was superior to the comparison methods, Kriging and RMS.

Nagorny et al. utilize Long-Short Term Memory (LSTM) neural networks to correlate raw process input signals of the screw position, injection pressure, in-mold pressure sensor and in-mold temperature sensor as well as thermographic images to predict the width of the considered part [20]. The LSTM approach scored an R^2-evaluation of the model quality with raw input signals input is 0.92.

Jain and *Bhuyan* focused on the comparison of different optimization techniques for warpage, among them genetic algorithms and an approach of human behavior based optimization (HBBO) [21]. Data was generated by the software Moldflow according to a Taguchi orthogonal array test plan varying holding pressure, fiber aspect ratio, mold temperature, fiber content, melt temperature and injection time. A backpropagation neural network served as a substitute model of the injection molding process. The result showed that the warpage could be minimized by utilization of HBBO. The modelling of the injection molding process with warpage as a quality parameter implicates that an ensuing second algorithm has to exploit the fitted model in order to optimize the machine parameters for the process.

Other research with artificial neural networks as surrogate model and a following optimization with evolutionary algorithms has been conducted for several quality parameters such as mechanical properties [22–24], shrinkage or warpage [17–21,25] [26], surface roughness [10], gate location [27] or a multitude of quality parameters [2,28–30]. However, each previously described research required purposely generated data to train their surrogate models. While a complete negligence of structured data for the ANN to learn the process correlations is unfeasible, amending the database with samples from other similar processes might be able to reduce the necessary amount of new data. The concept of transfer learning will be introduced in the following.

2.2. Transfer learning between processes

Transfer learning defines the endorsed fit of a machine learning model for a specified target assignment A_T by leveraged knowledge from one or many source assignments $A_{S,\,i}$ [31]. If the data domains $D_{s,\,i}$ and D_T are equal by definition, the assignment difference is reduced to the respective tasks $T_{s,\,i}$ and T_T. A marginal distribution $P(X)$ and a specific input parameter space X define a data domain D. A task T consists of an output parameter space Y and prediction model $f(X)$ that maps any $x \in X$ onto $y \in Y$. For the prediction of any quality parameter, i. e. label, the prediction model needs to learn the conditional probability $P(Y|X)$. This is ensured through the training of $f(X)$ with a dataset $M = \{(x_i, y_i) \mid i = I\}$ with I being the amount of labeled data in M.

It is intuitively assumable that the success of transfer learning is depending on the similarity of the assignments. Insufficient similarity can lead to negative transfer, provoking a deterioration of the prediction quality in the target domain after knowledge transfer [32]. However, positive transfer, hence the amelioration of learning in the target assignment, can manifest in three different effects, as described by *Torrey* and *Shavlik* [33]:

1. The quality of model is significantly higher in comparison to an untrained model.
2. A faster adaption of the model to the target domain by the provision of a specific amount of training samples or regarding the trained epochs can be observed.
3. The model lastly exhibits a greater generalization capability through transfer learning.

The success of transfer learning can be verified by the observation of any of those effects distinctly. It is not necessary to observe a combination or even all of the aforementioned points to assume a positive transfer in the experiments.

Following several possibilities to observe positive transfer learning effects, the approaches can also be separated into different categories. *Pan* and *Yang* introduce different categorization of transfer learning approaches and settings for the knowledge transfer [34]. Especially important is the differentiation between *inductive*, *transductive* and *unsupervised* transfer learning. Inductive transfer learning, which will be of interest in regards of this paper's content, assumes that while D_s and D_T are the same, T_S and T_T are different, given the respective definitions. The simplest setting of knowledge transfer is characterized by the availability of abundant training data of the source domain M_S and a limited but rather small amount of labeled training data of the target domain M_T. Exemplary use cases for transfer learning can be found in the field of handwriting recognition [35], recommendation systems for online shops [36] or natural language processing [37]. A good example for inductive transfer learning is presented by *Yosinski* et al. [38]: Convolutional neural networks perform well for the use case of image recognition, however, tend to require a substantial amount of training data, hindering the implementation for use cases with costly or difficult data generation. To overcome this obstacle, an approach of transfer learning is formulated, transferring the initial n layers of a pretrained model for a source data domain D_s on the model for the target domain D_T. *Yosinski* et al. concluded that the more similar data domains are, the more probable is a successful approach of knowledge transfer. Then, only finetuning of parts of the model has to be conducted as the central relationships between input and output parameters of the model have been learned by the source domain D_s, i. e. the presented training data M_S during pretraining [39].

For the injection molding process, it is particularly interesting to consider transfer learning for scarcity of data, e. g. if a new process has not been sampled abundantly to fit a model without any pre-learned knowledge (see Fig. 1). *Tercan* et al. propose an application of transfer learning for the improvement of model quality, transferring knowledge from a simulation database to an experiment [40]. A feedforward fully-connected ANN is fitted initially to a simulation dataset. While the melt temperature, cooling time, mold temperature, holding pressure time, holding pressure and injection time serve as input parameters, the part weight is considered as a suitable quality parameter. The model quality is measured by the degree of determination R^2, comparing predicted and true values of an unseen test dataset. After the initial training, the model weights are refined by a small amount of experimental data. In comparison with an ANN with the same structure but randomly initialized neuron weights, the fine tunes, i. e. transfer learned, model shows an expedited learning with the provided samples of the target dataset as well as a higher overall generalization level. *Lockner* and *Hopmann*, on the other hand, evaluate transfer learning regarding transfer learning between processes manufacturing different parts [14]. Injection molding process data for 60 different parts are generated with the simulation software Cadmould 3D—F Solver by simcon kunststofftechnische Software GmbH, Würselen, Germany. All simulation trials are conducted with the same material, PP 579S by Sabic, Riyadh, Saudi Arabia. While the process for one selected part serves as target domain D_T, the remaining process are considered source domains $D_{S,\,i}$. Three different transfer learning intensities are probed during the trials: Between one and three layers, counting from the input layer, are transferred onto the ANN for the target process. The results indicate that for a relatively shallow network, which performed well in the bespoken use case, a finetuning of the complete pretrained model generates the greatest benefit for the practitioner. With an ideal choice of the source model, transfer learning is observed to be capable of reducing the necessary amount of training data by about 80%. To consistently achieve this high amount of data saving potential, it is necessary to examine a feasible way to determine proximity of data domains, i. e. injection molding processes, before production start.

3. Data and methodology

In the injection molding process, product, material and settings of the machine parameters are crucial for high process and product quality. The following chapter introduces utilized software, describes the dataset and depicts the transfer learning methods used in the modelling approaches.

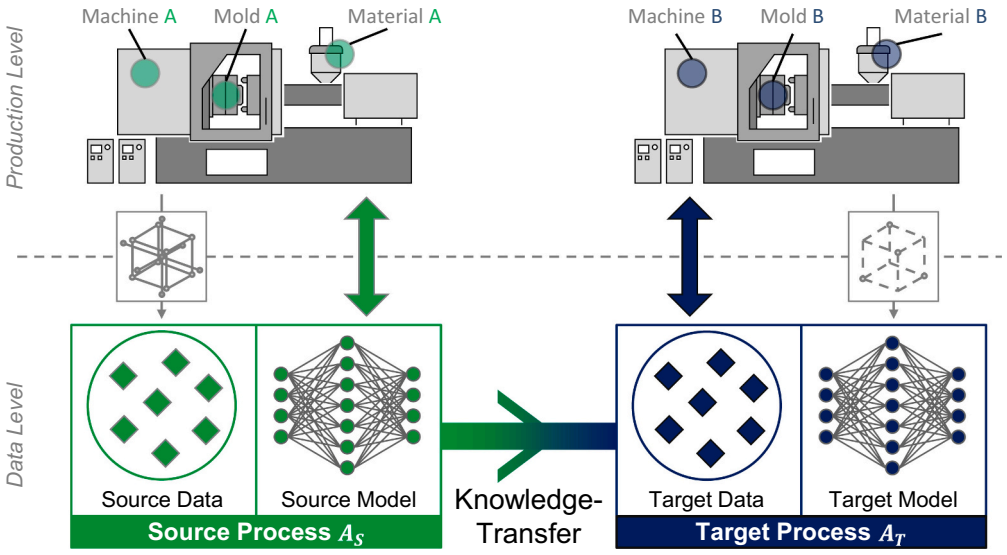

Fig. 1. Transfer learning concept between two processes with scarce data in injection molding.

3.1. Specimen and injection molding simulations

To obtain original source data, molding simulations of a toy building block are conducted. The simulations are performed with the software Cadmould 3D—F Solver in version 10.0.1.0 by simcon kunststofftechnische Software GmbH, Würselen, Germany. Cadmould models the injection and holding pressure phase by the implementation of the Navier-Stokes equations and Hele-Shaw approach for their approximation [41].

A toy building block with 2 rows of 4 studs each is chosen as an exemplary, industrially relevant part for the data generation (see Fig. 2). Relevant quality criterion for determination of part quality is part weight.

The simulations are conducted with 60 different, unreinforced thermoplastic materials chosen from the Cadmould material database. The materials can be divided into six material series of a different polymer class, namely polycarbonate (PC), polymethylmethacrylate (PMMA), polypropylene (PP), polyethylenimine (PEI), polyamide (PA) and polybutylene terephthalate (PBT). Table 1 lists 60 materials for the aforementioned polymer classes which belong to material series by different manufacturers.

During the simulations, the provided material data information from the Cadmould database has been used. The Carreau-WLF model served as viscosity model [42]. The Renner approach was utilized to represent the materials' pvT behavior [43]. As the complete description of all values for the 60 materials will exceed the scope of this paper, the ranges of the provided material data from Cadmould are displayed in Table 2. Important to note is the differentiation between solid and melt density. The values are either taken from the published material datasheets, from reports or read off pvT diagrams at the optimal processing temperatures.

Each point combination of material and machine setting parameters influences the part weight which is considered as the quality parameter of the process. Therefore, for each material, a 77-sample-points inscribed central composite design (CCD) experimental plan for process behavior identification is conducted. It is comprised of a full-factorial test plan, a central point as well as star points. For each CCD, the following six machine setting parameters are varied: melt temperature, cavity wall temperature, holding pressure time, cooling time, injection flow rate and holding pressure. Due to material-specific recommendations for the processing of the respective polymer materials, the extreme and step values for the variation of the machine setting parameters are adapted for each utilized polymer. In preceding simulation trials, the presented extreme values per material in each polymer class were determined. First, the center point was fixed by the default suggestion of the simulation program. Then, the extreme values for each machine setting were broadly set and fixed in case of a successful completion of the simulations. A diminution of the range was set if the simulation aborted. Table 3 illustrated the varying extreme values for the respective machine setting parameters, divided by the material classes. In total, 4620 times of simulation runs are conducted.

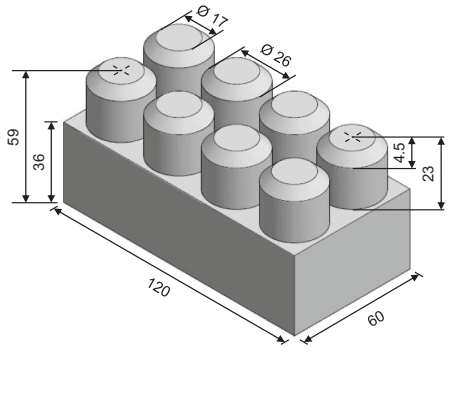

Fig. 2. Exemplary depiction of the toy building block '4 × 2 original' (comp. [14]).

Table 1
Polymer classes and aggregated material classes.

Material class	Amount of materials	Polymer class	Samples in the dataset	Manufacturer
APEC	13	PC	1001	Covestro
PLEXIGLAS	10	PMMA	770	Evonik Industries AG
SABIC PP	17	PP	1309	Sabic
ULTEM	6	PEI	462	Sabic
ULTRAMID	10	PA	770	BASF
VALOX	4	PBT	308	Sabic

3.2. Transfer learning approach

An often recurring situation in companies is the injection molding manufacturing with different, but related polymer materials. This, e. g., applies for the production of parts in different areas of operation or simply with a different color. Therefore, the processed material variant might be the same base polymer or from the same polymer class, but e. g., with deviating viscosity or other material properties. Manufacturing issues widely appear when being confronted with the required processing of new material or polymer classes. It is therefore to be probed if a transfer between different material classes can be successful in order to alleviate this frequently observed manufacturing obstacles for the setup process in injection molding. For the transfer learning approach in this paper, a "single domain assumption" for the respective material classes was made. The data of five material classes, PMMA, PP, PEI, PA and PBT were utilized as source domains. The PC class was selected to function as target domain.

Transfer learning can be defined as the degree of generality of a set of features that can be used on a target task but have been learned on a source or base task [38]. A two-step finetuning approach aims to enable an efficient transfer of knowledge obtained from the source dataset to improve learning in the target task using a comparably small dataset. Thereby, the necessity to collect large and costly amounts of data for model training to achieve satisfactory model qualities will be relaxed. If not specified differently in this work, the target task is represented by the PC dataset while dataset for the base task, or the base learner in this case (see Fig. 3), is composed of the remaining datasets for the other material classes.

Firstly, a base learner is trained on the data in the composed dataset. This pre-training is conducted in order to pre-configure the model's neuron weights to achieve feasible, yet not accurate predictions in the target domain with pending similarity assumption.

Secondly, in a transfer learning approach, the weights learned by the base learner are transferred to the model for the target task (ANN-TL). The hybrid network is completed by newly initialized layers to form a five-layered network. The ANN-TL model is retrained using a comparably small target domain dataset.

The notation "$B2_FI3$" as shown in Fig. 3 on the right identifies the amount of transferred layers, counted forward from the input layer, from the base learner ("B2") and the amount of untrained layers ("I3"), counted backwards from the output layer. Additionally, the subscript "F" or "T" declares the neuron weights of all transferred layers as trainable or fixed. While trainable layers ensure an adaption of to the dataset in the target task, fixed layers could speed up model training. However, a fixation only allows high model qualities if during base learner training somewhat general relationships to the injection molding process have been identified here that can be found again for the target task [38]. Subsequently, the more similar source and target task are, the higher is the probability to be able to accelerate the training velocity.

Several constellations of ANN-TL are probed together with the

Table 2
Excerpt of material features from the Cadmould material database and their value ranges.

Material features	Unit	PC	PMMA	PP	PEI	PA	PBT
No flow temperature	[°C]	[182, 240]	[125, 132]	[120, 134.5]	[240, 245]	[180, 247]	[200, 205]
Solid density (at room temperature)	[kg/m^3]	[1180, 1120]	[1190, 1170]	[905, 907]	[1270, 1310]	[1100, 1130]	1310
Thermal conductivity	[$W/m \cdot K$]	[0.163, 0.171]	[0.181, 0.190]	[0.1726, 0.1980]	[0.24, 0.278]	[0.271, 0.33]	[0.22, 0.24]
Maximum mold temperature	[°C]	[120, 160]	[70, 90]	[50, 60]	[165, 180]	[50, 60]	100
Ejection temperature	[°C]	[142, 200]	[104, 112]	[100, 107]	[195, 228]	[60, 215]	183
Melt density	[kg/m^3]	[960,1010]	[1040, 1062]	[735, 753]	[1150, 1210]	[946.3, 984.6]	[1086, 1110]

Table 3
Range [min, max] of machine settings for performed DoE.

Machine settings	PC	PMMA	PP	PEI	PA	PBT
Melt temperature [°C]	[320, 340]	[220, 260]	[200, 260]	[370, 410]	[234, 300]	[250, 270]
Cavity wall temperature [°C]	[100, 155]	[53, 90]	[19, 60]	[140, 180]	[24, 60]	[40, 100]
Holding pressure time [s]	[10, 20]	[2, 6]	[3, 16]	[1, 5]	[1, 2]	[0.2, 1.2]
Cooling time [s]	[20, 37]	[7, 16]	[16, 30]	[5, 14]	[3, 10]	[1.5, 2]
Injection flow rate [cm^3/s]	[31, 58]	[35, 52]	[80, 200]	[30, 56]	[40, 66]	[30, 56]
Holding pressure [bar]	[50, 306]	[350, 691]	[100, 527]	[200, 541]	[400, 656]	[400, 912]

variation of provided training samples. The output layer will always be reinitialized as it is assumed to be highly domain specific. To evaluate the transfer learning success, the models' degree of determination (R^2) for each constellation will be compared, as seen in other pertinent work [14,17,44,45]. R^2 is commonly defined as the quotient of the sum of squares explained (SQE) and the sum of squares total (SQT). Here, it is calculated

$$R^2 = \frac{SQE}{SQT} = \frac{\sum \left(\hat{y}_i - \bar{y}\right)^2}{\sum (y_i - \bar{y})^2} = 1 - \frac{SQR}{SQT} = 1 - \frac{\sum (y_i - \hat{y})^2}{\sum (y_i - \bar{y})^2}$$

It shall be noted that R^2 become smaller than 0 if the model's predictions (yˆi) are further off the true value (yi) than the mean average (y-). If not stated otherwise, all results are averaged by modelling results from a 5-fold cross-validation.

4. Results and discussion

In this chapter, a holistic approach is presented for the data analysis steps that are necessary once process data have been generated, aiming to minimize the application effort for the operator. The section of experimental results regarding the weight-prediction model is divided into four subsections. In the first subsection, the data pre-processing by scaling and separation is described, as well as the development and application of suitable algorithms for feature selection. In the second subsection, the appropriate ANN hyperparameters for the modelling task are tuned using grid search optimization. The third subsection elucidates the effects on prediction performance with transfer learning caused by different target dataset sizes, different number of transferred layers and single or multisource data and its contribution on industrial application of the methodology.

4.1. Data pre-processing and feature selection

Data pre-processing consists of two main parts: unification of the distribution of the input data, and separation of the data into training, validation, and test sets. If the scales of the input values are significantly different, convergence during training is slow or not achieved and the model's prediction quality remains subpar. As the process conditions have very different scales (see Table 3), feature scaling is used in order to scale and translate each feature individually in the range between zero and one.

Fig. 4 shows an excerpt of the available input and output parameters in the generated dataset for process modelling. Machine settings and material properties are defined as model inputs while the output is the part weight as quality parameter. Firstly, in order to transform raw samples into applicable data for model training it is cleaned. Constant, empty, zero and meaningless columns are removed from the dataset. Columns with insignificant variance measure, namely, lower than 0.01, are removed as well.

After data re-scaling, the dataset is divided randomly into subsections of 80% for training, 10% for validation, and 10% for testing.

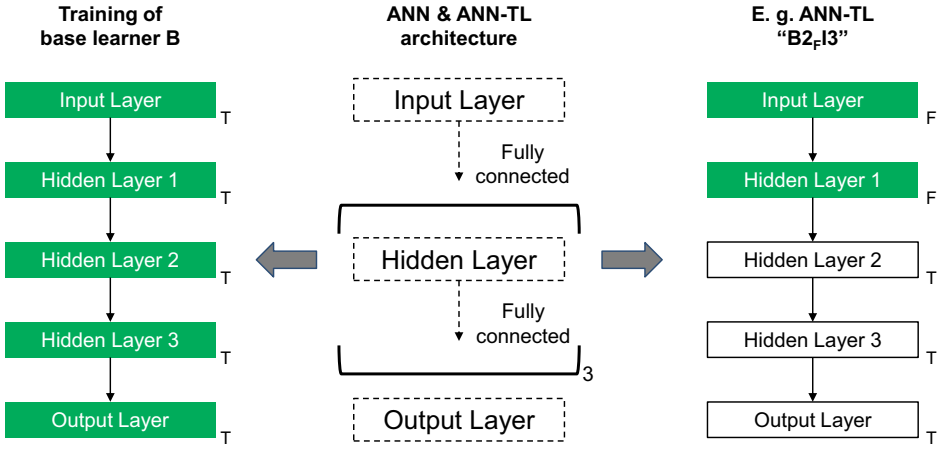

Fig. 3. Overview of the examined structural transfer learning approaches.

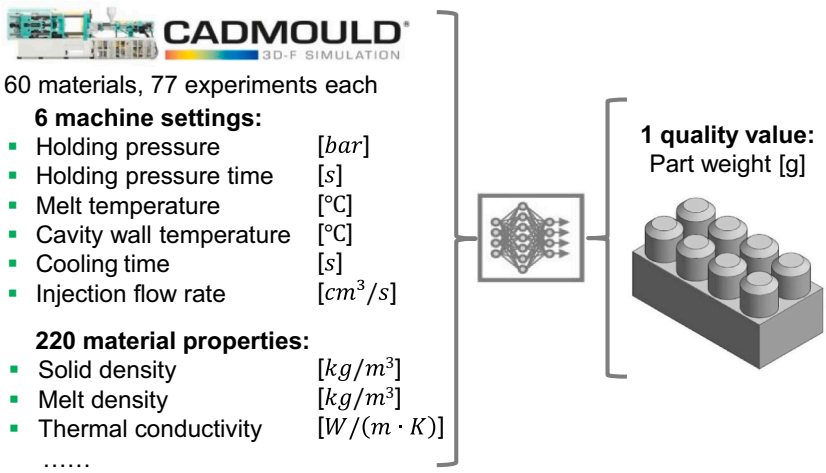

Fig. 4. Possible input and output variables of the ANN before feature selection.

The training set is used to train the ANN model and for hyperparameter tuning. The test set is used to evaluate the model quality of the tuned model. Early stopping is implemented together with the validation set to prevent overfitting of the model to the training set.

As shown in Fig. 4, approximately 220 material features have been found in the simulation dataset. However, the processing of this many input variables will have a significant impact on the data demand during training of the estimator. Furthermore, it can be assumed that some of the features are irrelevant for the bespoken modelling task. Therefore, feature selection is performed firstly. Feature selection algorithms rank the features in accordance with their predictive scores.

Algorithms for the determination of feature importance like XGBoost [46], Extra Tree [47], Decision Tree [48], Gradient Boosting [49], Ada Boost [50] and Random Forest [51] each employ a different calculation method. It is not possible to determine in advance which algorithm's results can be considered particularly favorable for the sequential modelling. Therefore, all previously named algorithms are used together in an ensemble feature selection procedure.

For each algorithm, the features are ranked with the calculated feature importance score, the sum of which is equal to one. The algorithms are employed by scikit-learn python package with default hyperparameters [52]. The final importance per feature is set to the mean value of the respective feature for all probed algorithms. Fig. 5 shows the structure of feature selection by ensemble learning method.

As transfer learning between different material classes and polymers will be employed with the necessity of the same input parameter space, the combined dataset of all 60 materials is utilized for feature selection. The machine settings and material properties are the inputs while part weight is the target. The feature importance scores are generated during the training of the feature selection models. Table 4 ranks the five features with the highest mean feature importance score of six feature selection algorithms. A complete list of the features can be found in the appendix (Table 9).

For an overview of the selected features, Fig. 6 illustrates the mean feature importance of each feature. Features with score lower than the threshold set to 10^{-5} are ignored for the consequential modelling. The threshold is chosen that low as to accommodate possible variations of the feature importance for other material characteristics in the injection molding process that have not been sampled here during data generation but could be probed for a later validation with more materials. After this process, 6 machine setting features and 44 material property features are selected as input of ANN model.

4.2. Hyperparameter optimization

In order to optimize the model quality, several important hyperparameters have been considered for a tuning before the experiments. Grid Search is conducted to structurally vary the chosen values [16]. To implement grid search more efficiently, firstly, the tuning process starts with broad ranges of hyperparameter values and performs a coarse grid search for smaller number of epochs or smaller training set. Secondly, a narrower search is performed with more epochs or a larger part of the training set. A sequential tuning approach is chosen due to the large amount of variables in the tuning process. The hyperparameter tuning

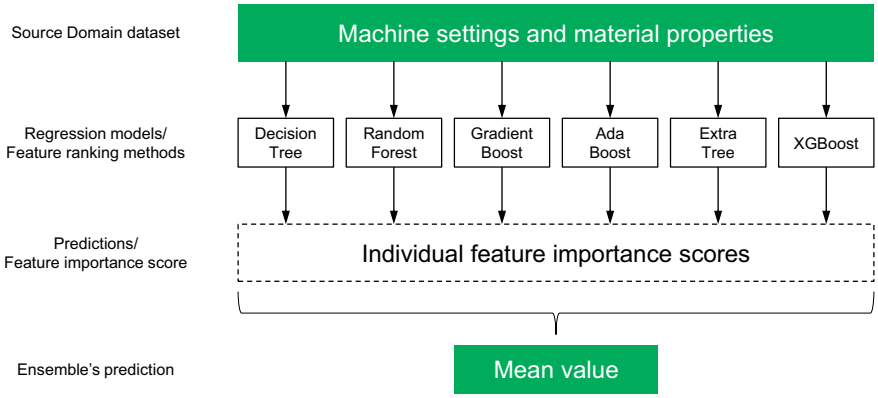

Fig. 5. Custom ensemble feature selection by six different estimators.

Table 4
Custom ensemble feature selection results for feature importance.

No	Feature	Decision tree	Random forest	Gradient boost	Ada boost	Extra tree	XGBoost	Mean
1	Solid density	0.5599	0.1751	0.1976	0.2312	0.0704	0.1355	0.2283
2	Holding pressure	0.1691	0.1671	0.1698	0.0372	0.1705	0.0364	0.1250
3	Thermal conductivity	0.0079	0.0125	0.0106	0.0300	0.0125	0.5781	0.1086
4	No flow temperature	0.0000	0.1856	0.0417	0.1198	0.0981	0.0946	0.0900
5	Ejection temperature	0.0169	0.1288	0.1822	0.0862	0.0643	0.0003	0.0798

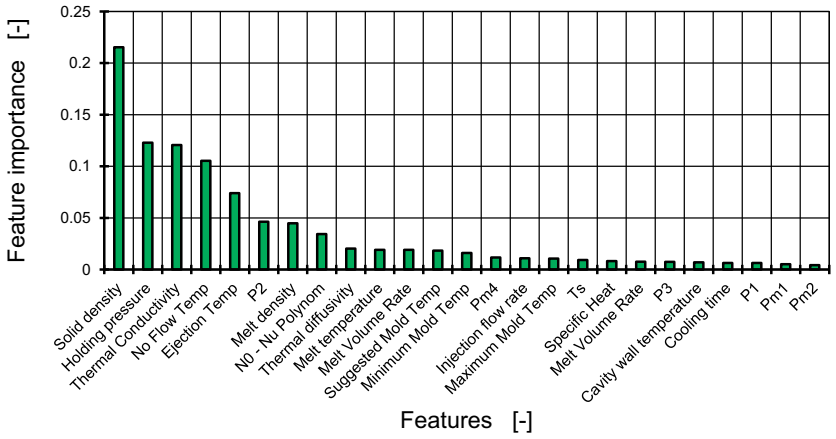

Fig. 6. Feature importance score ranking for prediction of the part weight for the 20 most important features.

sequence, probed values as well as the results can be seen in Table 5.

The best results are obtained with 150 samples randomly chosen for each batch. Convergence is consequentially observed after 300 epochs. Stochastic gradient descent (SGD) and RMSProp optimizer algorithms are evaluated but best results are achieved with Adaptive Momentum stochastic gradient descent optimizer (Adam) with default parameters. Learning rate is set to 0.001 per iteration which is fit to guarantee convergence of the adapted gradient decent algorithm. The neuron weights of artificial neural networks need an initial weight for the training. The weights can be set based on dependencies on the model architecture such as the number of neurons in a consecutive or preceding layer. In this case, the best results could be achieved for the initialization strategy glorot_uniform. Activation functions influence the neuron's output intensity.

Many functions such as sigmoid, tanh, the rectified linear unit (ReLU) are compared. The softplus activation function is tested to be the most appropriate activation function (see Table 6). It is clearly visible that function with an upper boundary, like Tanh or Sigmoid, cannot handle the dataset, whereas activation functions with an unrestricted range generate good results. ReLu dominates the linear activation function, but Softplus performs even better with an R^2 score of 0.946. An explanation could be the problem of vanishing gradients with ReLu, which does not occur with the Softplus function. [53]

Both dropout and regularization are options during model fitting to reduce or eliminate the risk of overfitting the model to the training dataset. No dropout probability is set for the fully connected layers but a L2 penalty with default value of 0.01 is set on backpropagation.

Table 5
Hyperparameter tuning search space and results.

Sequence	Hyperparameter	Initial	Search space	Result
1	Batch size	none	[10, 50, 100, 150, 200]	150
2	Number of epochs	300	[100, 200, 300, 500]	300
3	Optimizer	'Adam'	['SGD', 'RMSProp', 'Adagrad', 'Adadelta', 'Adam', 'Adamax', 'Nadam']	'Adam'
4	Learning rate	0.001	[0.001, 0.01, 0.1, 0.2, 0.3]	0.001
5	Weight initialization	none	['uniform', 'lecun_uniform', 'normal', 'zero', 'glorot_normal', 'glorot_uniform', 'he_normal', 'he_uniform']	'glorot_uniform'
6	Activation function	'relu'	['softmax', 'softplus', 'softsign', 'relu', 'tanh', 'sigmoid', 'hard_sigmoid', 'linear']	'softplus'
7	Dropout	0.0	[0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]	0.0
8	Regularization	none	['l1', 'l2', 'l1_l2']	'l2'

4.3. Comparison of the transfer learning approaches

As motivated before, transfer learning is especially attractive in situations where only incomplete datasets of an injection molding process are available. This subchapter elucidates the effects on prediction performance of transfer learned artificial neural networks (ANN-TL) caused

Table 6
Comparison of different activation functions.

Activation function	R^2 score
Softmax	−2403.637573
Softplus	0.946451
Softsign	−1510.902820
Relu	0.936178
Tanh	−1446.382356
Sigmoid	−1705.264945
Hard Sigmoid	−2096.795019
Linear	0.835198

by different number of transferred layers. With a defined number of layers chosen for the transfer approach, different target dataset sizes and single or multisource data for processes with different polymer classes for transfer learning will be examined to determine the fit of transfer learning for the specific use case in injection molding. Each experiment is repeated five times and the occurring standard deviations are illustrated.

4.3.1. Effect of the number of transferred layers in transfer learning

To explore whether ANN-TL can help to provide more accurate quality estimations than an ANN model trained from scratch, experiments are implemented to empirically investigate the effect of the number of layers transferred from a pre-trained ANN on the performance of the ANN-TL. Fig. 7 shows the resulting models' degree of determination in dependence of the number of layers copied of the base learner. The probed transfer learning approaches follows the denoted procedures in Fig. 3. Choosing $n = 0$, which denotes that no layers are copied from the pre-trained ANN model and all the layers are initialized from scratch, essentially results in an ANN model without transfer learning (*target task base learner*).

For finetuning with trainable layers, a gradual improvement of the model quality can be observed when raising the number of transferred layers: The transfer of $n = 4$ results in a degree of determination of 0.966 in comparison to 0.922 for the target task base learner. An explanation could be that in a typical deep learning architecture, the features in the last few layers are specific to a task (*specification features*). The objective of both the source and target tasks is to estimate the part weight of injection molding with different materials. From the results it can be concluded that the similar features in the last few layers of the considered models contribute to the upward performance after $n = 2$.

For the transfer of layers with fixed neuron weights, a significantly different result can be observed. Firstly, depending on the number of transferred layers *n*, the transfer approach shows an alternation between an increase in model quality and a decrease. E. g., for the $B1_FI4$, the model's degree of determination decreases from 0.922 to 0.889, whereas for $B2_FI3$ it surges to 0.959, which is in this case better than the average result for the approach with trainable neuron weights. The underperformance of $B1_FI4$ suggests that the pre-trained ANN (base learner, comp. Fig. 3) model contains fragile *co-adapted features* throughout successive layers. These features interact with each other in a complex way such that this co-adaptation together with the fixing of neuron weights of the layers, preventing a re-training, cannot guarantee good results on a small dataset for ANN-TL [38]. Although fast computation of the retraining can be observed for the finetuning with fixed neuron weights, the underperformance does not rectify this approach as no real time constraint can be found for the offline optimization of machine setting parameters in injection molding.

The finetuning approach with trainable neuron weights and four transferred layers, $B4_TI1$, results in the highest R^2 value compared with the models with any other numbers of copied layers. Thus, when transferring from a source model to a target model, the ANN-TL model is proposed to contain four layers directly copied from a pre-trained ANN model, as this depth of transferred layers seems to ensure a satisfactory performance of the proposed method.

4.3.2. Effect of training dataset size on transfer learning

It is industrially particularly relevant to determine the influence of the availability of training data on the resulting model quality. To examine the results for the transfer learning approach, data partition of 80% from the target task is chosen randomly for training while the remaining samples serve as testing dataset. 10 subsets of the training dataset of different sizes are constructed by increasing and decreasing the number of samples N from the original training dataset. The maximum number of samples in a training dataset is denoted as n_{max} and ranged from $0.1N$ to N, specifically, $n_{max} \in \{0.1N, 0.2N, 0, 3N, 0.4N, 0.5N, 0, 6N, 0.7N, 0.8N, 0.9N, N\}$. For $n_{max} = N$, the original training dataset is used. For each of the 10 training datasets, the same test dataset is used to evaluate the generalization ability of the trained ANN-TL model. As well as in Section 4.3.1, finetuning with trainable and fixed layers is probed against the target task base learner approach. Because the transfer of 4 pretrained layers from the base learner achieved the highest R^2 values, the architectures are set to $B4_TI1$ and $B4_FI1$.

The results of this comparison and the effect of the training dataset size on the R^2 values are illustrated in Fig. 8. The x axis denotes the artificially reduced availability of target domain data, based on the whole training dataset size.

It can generally be observed that transfer learning seems to be significantly beneficial to the degree of determination of the model when one material class as target task and all the other material classes as source task. The performance improvement, e. g. between $B4_TI1$ and the target task base learner is significantly large when only a small dataset in the size of 10 to 20% of the overall training dataset is available for training.

As the size of the training dataset increases, the performance improvements by employing transfer learning gradually decrease. $B4_TI1$ is oscillating on a high level of model quality between the minimum R^2 of 0.979 for 80% training data and the maximum of 0.990 for 60% training data. This alternating performance is due to the rather small dataset in regards to the number of degrees of freedom in the ANN-TL model.

Fig. 9 depicts the comparison of the true values of the part weight and the predictions by the transfer learning approach $B4_TI1$. It can be seen that the besides some smaller outliers the predictions are close to the true values. The pretraining of the base learner, as described before,

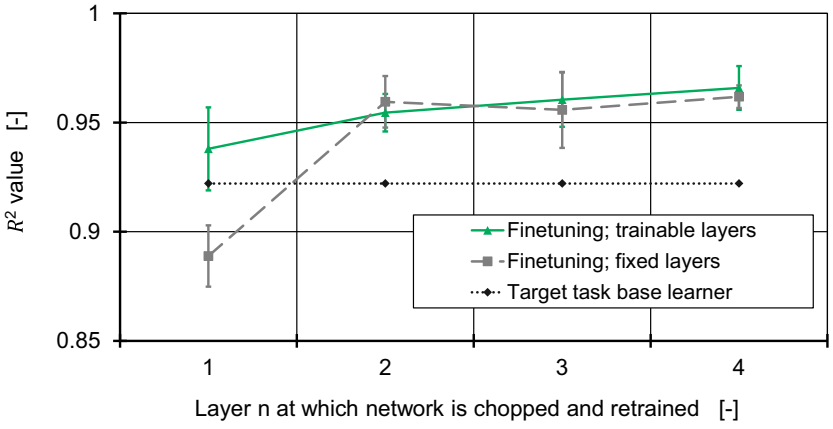

Fig. 7. Transfer learning results for varied number of transferred layers.

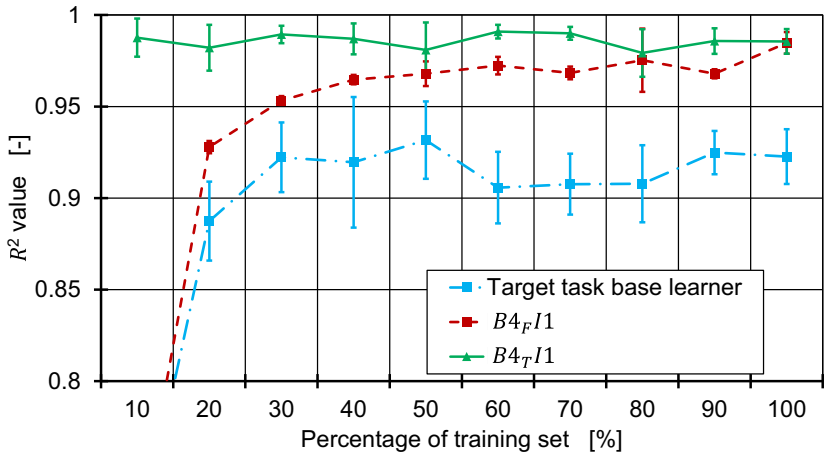

Fig. 8. Transfer learning results based on the amount of training samples. Note that 100% means the whole training set, excluding the testing set.

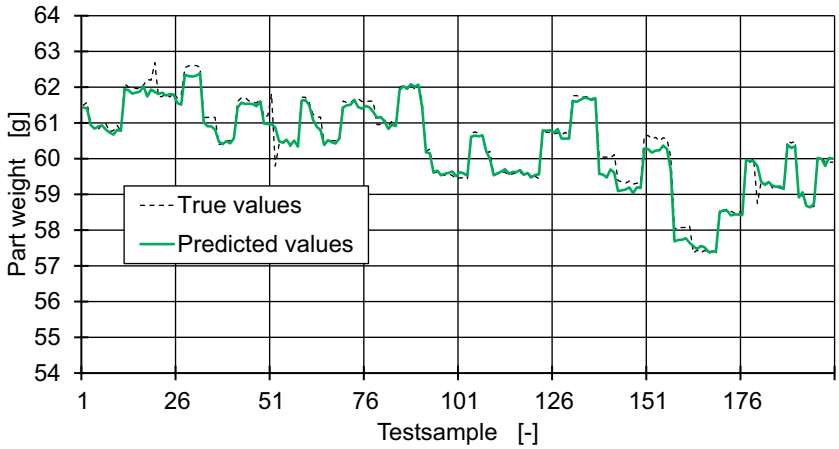

Fig. 9. Comparison of true and predicted values by $B4_TI1$ for a test data subset of the polycarbonate dataset.

has been conducted with the source domains (compare Section 3.2). For the adaption to the target domain, 80% of available data from the polycarbonate class dataset were used for training. 20% (roughly 200 samples, as depicted in Fig. 9) represent the test data subset.

$B4_FI1$ and the target task base learner show a significant initial increase in model quality between 10 and 30% of training data. For a higher number of provided samples, $B4_FI1$ seems to simply slow down the resulting increase in model performance which appears to converge gradually for $B4_TI1$. The target task base learner, however, shows a saturation and oscillation of model quality between 30 and 100% of training data. The minimum and maximum R^2 values in this range are 0.906 for 60% and 0.932 for 50%, respectively. The implications of diverging model quality results between transfer learning with frozen weights and no transfer learning are particularly interesting: While the target task base learner only accesses the implicit information about the injection molding process from the data for material of class APEC, the $B4_FI1$ approach for ANN-TL has been pre-trained on data from other polymer materials and classes. This leads to the observation, keeping all other parameters constant as it has been done in these experiments, that the pretraining with other materials contains enough relatable information for the APEC material class to result in a better model performance compared to a model training without pretrained neuron weights.

Assessing by the displayed results, leveraging transfer learning for the training of injection molding models for different materials rather than training a deep learning model from scratch is a better way to achieve high model quality. Transfer learning therefore qualifies as a viable technique to reuse knowledge from previously recorded process data. It is thinkable to implement the technique as component to a specific case-based reasoning (CBR) environment for the setup of injection molding processes. Previous work has shown the complimentary characteristics of CBR and transfer learning [54–56]. Furthermore, an intelligent retrieval strategy for stored injection molding process data should be developed for optimized knowledge reuse.

4.3.3. *Effect of the number of source datasets on transfer learning*

In Section 4.3.2, five different material classes have been used simultaneously for pre-training of a base ANN to examine the general feasibility of agglomerated heterogeneous injection molding process data from processes with different materials for transfer learning in the domain. However, it is a research question how every single of the five material classes contributed to the advantages of the transfer learning approach that could be observed. Therefore, an additional experiment is conducted, analyzing the influence of each material class data on the transfer learning results in a cumulative approach when discretely adding new datasets as training data for the model pretraining.

Fig. 10 shows a composition of histogram plots of the weight distributions of the different materials classes that have been sampled in the experiments. The x axes have been standardized for better comparison, the y axes show the respective count of samples in each of 25 bins of the material class histogram plots. For PC, PMMA, PEI, PA and PBT, the weight distributions cluster in the range between 55 and 75 g,

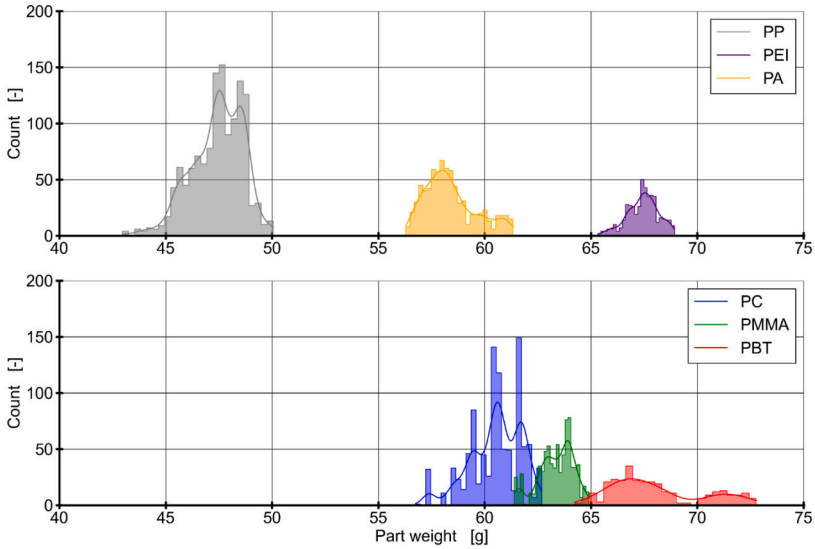

Fig. 10. Different part weight distributions within the material classes.

whereas the distribution of PP stands apart below the ones of other material classes. The different mean values in the weight distributions are mainly influenced by the varying densities of the polymer classes.

In order to validate the effect of transfer learning for other target domains, instead of PC, as seen before, PBT is chosen as target domain (Base B). In Base B, 80% of PBT dataset are chosen randomly for training and the rest for testing. PBT is the material class which contains the fewest data samples (see Table 1). When adding the dataset of a new material class (Base A) to the pre-training dataset, the contained samples are shuffled to avoid any bias by the data generation process. Fine tuning with trainable neuron weights ($B4_TI1$) is used as the transfer learning model.

The order in which additional datasets are included for the model pre-training is denoted in Table 7. The histograms per polymer class in Fig. 10 mostly display a highly irregular distributions of the respective part weights. This is because each material class contains several different polymers which have been omitted for simplicity reasons. The order is a result of a dataset examination of each material class: For the experiments, the objective is to determine observable differences in the transfer learning results for the chosen target domain in relation to the provided material class injection molding simulation data. It is assumed that close proximity of source and target data domains contribute positively to the effectiveness of transfer learning (compare Section 2.2). Therefore, firstly the material class PEI is used as training dataset which shows a similar weight distribution compared to PBT. Then, the additional impact on the transfer learning result for a less related material class in terms of the quality parameters to the target domain is examined by adding the dataset for PP. Afterwards, the additional datasets are added in no specific order.

Fig. 11 illustrates the observed results for the cumulative approach. The varying composition of the source domain is depicted by the legend.

Table 7
Mean R^2 for transfer learning with different amounts of source datasets and target training samples.

Amount of source domains	Amount samples	Amount of samples in the training dataset	Mean R^2 value
1	462	462 (PEI)	0.7713
2	1771	+ 1309 (PP)	0.7723
3	2541	+ 770 (PA)	0.8512
4	3311	+ 770 (PMMA)	0.8570
5	4312	+ 1001 (PC)	0.8657

It can be observed that the multi-source distribution transfer yields better results compared to the single source distribution, namely transfer on 1 source. The scarcity of the source domain samples causes difficulty to find an appropriate mapping between inputs and outputs. As the number of sources rises, the performance of ANN-TL models increases roughly. Table 8 summarizes the mean R^2 value of different training data percentages, which shows that the R^2 value rises with the increase of target dataset size.

For small amounts of target data, i. e. 10%, in the case of PBT as target domain it seems that the results lead to the conclusion that the more datasets are included in the source domain, the better the model quality will be. For higher target domain data availability, no clear trend towards higher model quality for higher data availability can be observed. However, in comparison to the no transfer learning results, a clearly higher asymptote for rising data amounts can be observed. This corresponds to the third advantage of transfer learning by *Torrey* and *Shavlik* as mentioned before [33].

Another interesting observation can be made when comparing Fig. 8 ("Transfer learning fine-tuning") and Fig. 11 ("Transfer on 5 sources"). These results resemble the exact same approach with the exchange of the PC with the PBT dataset from training to test dataset. While for the PC dataset an initially already very high R^2 can be achieved by transfer learning, the model quality in that range is delayed for the PBT dataset. This can partly be explained by the different dataset sizes of PC and PBT (see Table 1). 10% of target domain training data is represented roughly by 80 data samples, whereas this amount of training data samples is reached between 30% and 40% in Fig. 11. However, adapting to the PBT dataset seems to be a bigger challenge during model training, compared to PC. A possible explanation could be complex crystallization processes in a semi-crystalline polymer material such as PBT. The observed specific material class differences regarding behavior during simulation need to be further examined and finally validated in experimental trials.

5. Conclusion and outlook

Machine learning methods, especially ANNs, are suitable to learn the relation between machine setting and quality parameters in injection molding. Transfer learning can help to reduce the required amount of required training data of the target process and therefore raise the attractiveness of data-based process modelling for industry. In this work, simulation datasets have been generated for 60 different materials of 6 polymer classes, each dataset containing 77 samples, while constantly

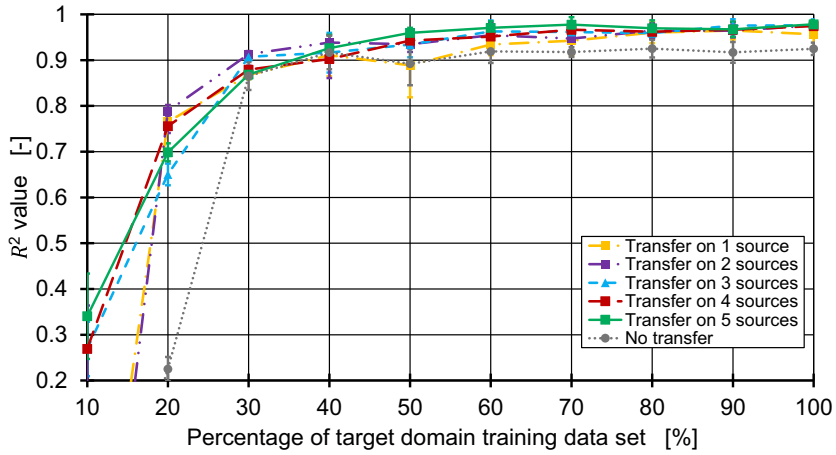

Fig. 11. Transfer learning results based on the number of source domains.

Table 8
Mean R^2 for varied amount of training samples for various training data compositions.

Training data	Size of target dataset	Worst R^2 value	Mean R^2 value	Best R^2 value	R^2 No transfer
10%	24	−0.6578	−0.1007	0.3404	−0.348137
20%	49	0.2250	0.6472	0.7886	0.225099
30%	73	0.8654	0.8835	0.9129	0.865422
40%	98	0.9024	0.9185	0.9385	0.916698
50%	123	0.8887	0.9253	0.9595	0.892198
60%	147	0.9190	0.9487	0.9706	0.919008
70%	172	0.9176	0.9521	0.9777	0.917695
80%	196	0.9252	0.9568	0.9697	0.925273
90%	221	0.9170	0.9592	0.9755	0.91701
100%	246	0.9248	0.9642	0.9783	0.924858

simulating the same part to focus on the impact of material variances on the part weight.

Finetuning with trainable neuron weights in constellation $B4_TI1$ has proven to generate the best transfer learning results in this case. For both probed target polymer classes, PC and PBT, transfer learning resulted in a higher model quality for small amounts of training data and a higher asymptote. This indicates that the adaption of process models in injection molding to new polymer classes can be significantly facilitated by transfer learning. Based on the results, negative transfer by the assumption of multiple polymer classes as one source domain cannot be identified, consequently indicating that available datasets of injection molding process data with varying material can be used together for pretraining.

Different materials are sampled experimentally at the moment to validate the findings by applying the approach to experimental datasets. With the experimental datasets, an analysis will be conducted if the transfer learning success can be explained by differences in the material characteristics. Furthermore, the combination of process, material and part influences on the injection molding process when modelling the processes needs to be the near objective as this could lead to the development of a machine-specific modelling approach for machine parameter optimization.

Furthermore, the presented results indicate good transferability between polymer classes. It is also interesting if within a certain material class, the seen transfer learning success can be observed as well, maybe even at a higher level as polymers within a class generally show higher similarity.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This research was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC-2023 Internet of Production – 390621612. The support and sponsorship is gratefully acknowledged and appreciated. Additionally, we want to thank simcon kunststofftechnische Software GmbH for the support.

Appendix A. Appendix

A.1. Appendix: features after data cleaning

The following table lists the machine parameters and material properties used in the modelling approach.

Table 9
Features after data cleaning.

No	Name	Unit	Parameter type
1	Solid density	$\frac{kg}{m^3}$	Material property
2	Holding pressure	bar	Machine parameter
3	No flow temperature	$°C$	Material property

(*continued on next page*)

Table 9 (*continued*)

No	Name	Unit	Parameter type
4	Thermal conductivity (λ) - constant	$\frac{W}{m*K}$	Material property
5	Ejection temperature	$°C$	Material property
6	P2 (Carreau-WLF Law)	s	Material property
7	Melt density	$\frac{kg}{m^3}$	Material property
8	N0 (ν_0) - Nu polynom	–	Material property
9	Thermal diffusivity	$\frac{m^2}{s}$	Material property
10	Melt temperature	$°C$	Machine parameter
11	Melt Volume Rate (MVR)	$\frac{cm3}{10min}$	Material property
12	Suggested mold temperature	$°C$	Material property
13	Minimum mold temperature	$°C$	Material property
14	Pm4 (pvT-behavior)	bar	Material property
15	Injection flow rate	$\frac{cm3}{s}$	Machine parameter
16	Maximum mold temperature	$°C$	Material property
17	Melt temperature (T_s)	$°C$	Machine parameter
18	Specific heat (C_p) - constant	$\frac{J}{kg*K}$	Material property
19	Melt Volume Rate (MVR)	$\frac{g}{10min}$	Material property
20	P3 (Carreau-WLF Law)	–	Material property
21	Cavity wall temperature	$°C$	Machine parameter
22	Cooling time	s	Machine parameter
23	P1 (Carreau-WLF Law)	$Pa * s$	Material property
24	Pm1 (pvT-behavior)	$\frac{bar*cm3}{g}$	Material property
25	Pm2 (pvT-behavior)	$\frac{bar*cm3}{\frac{g}{K}}$	Material property
26	Ps3 (pvT-behavior)	bar	Material property
27	E0 - E polynom	MPa	Material property
28	T0 (Carreau-WLF Law)	$°C$	Material property
29	Pt1 (pvT-behavior)	$°C$	Material property
30	E3 - E polynom	$\frac{MPa}{K^3}$	Material property
31	Ps1 (pvT-behavior)	$\frac{bar*cm3}{g}$	Material property
32	E2 - E polynom	$\frac{MPa}{K^2}$	Material property
33	Pt2 (pvT-behavior)	$\frac{°C}{bar}$	Material property
34	Pm3 (pvT-behavior)	bar	Material property
35	Ps4 (pvT-behavior)	bar	Material property
36	Ps2 (pvT-behavior)	$\frac{bar*cm3}{\frac{g}{K}}$	Material property
37	E1 - E polynom	$\frac{MPa}{K}$	Material property
38	E constant	MPa	Material property
39	Suggested Melt Temperature	$°C$	Material property
40	Coefficient of Thermal Elongation (CTE) constant	$\frac{1}{°C}$	Material property
41	Minimum melt temperature	$°C$	Material property
42	Holding pressure time	s	Machine parameter
43	Load (MFR/MFI)	kg	Material property
44	Temperature (MFR/MFI)	$°C$	Material property
45	Temperature (MVR)	$°C$	Material property
46	Load (MVR)	kg	Material property
47	Maximum melt temperature	$°C$	Material property
48	Ps5 (pvT-behavior)	$\frac{cm3}{g}$	Material property
49	Ps6 (pvT-behavior)	$\frac{1}{K}$	Material property
50	Ps7 (pvT-behavior)	$\frac{1}{bar}$	Material property

References

[1] Brecher C, Jeschke S, Schuh G, Aghassi S, Arnoscht J, Bauhoff F, et al. Integrative Produktionstechnik für Hochlohnländer. In: Brecher C, editor. Integrative Produktionstechnik für Hochlohnländer. Berlin H: Springer Verlag; 2011.

[2] Meiabadi MS, Vafaeesefat A, Sharifi F. Optimization of plastic injection molding process by combination of artificial neural network and genetic algorithm. J Opt Ind Eng 2013;6:49–54. Unknown DoI.

[3] Ademujimi TT, Brundage MP, Prabhu VV. A review of current machine learning techniques used in manufacturing diagnosis. In: Lödding H, Riedel R, Thoben K-D, von Cieminski G, Kiritsis D, editors. Advances in production management systems. The path to intelligent, collaborative and sustainable manufacturing. APMS 2017.

IFIP advances in information and communication technology. Cham: Springer International Publishing; 2017. p. 407–15.
[4] Weichert D, Link P, Stoll A, Rüping S, Ihlenfeldt S, Wrobel S. A review of machine learning for the optimization of production processes. Int J Adv Manuf Technol 2019;104:1889–902. https://doi.org/10.1007/s00170-019-03988-5.
[5] Kim D-H, Kim TJY, Wang X, Kim M, Quan Y-J, Oh JW, Min S-H, Kim H, Bhandari B, Yang I, Ahn S-H. Smart machining process using machine learning: a review and perspective on machining industry. Int J Precis Eng Manuf Green Technol 2018;5:555–68. https://doi.org/10.1007/s40684-018-0057-y.
[6] Fazel Zarandi MH, Sadat Asl AA, Sotudian S, Castillo O. A state of the art review of intelligent scheduling. Artif Intell Rev 2018;53:501–93. https://doi.org/10.1007/s10462-018-9667-6.
[7] Popov VL, Heß M, Willert E. Viscoelastic materials. In: Handbook of contact mechanics. Berlin H: Springer Verlag; 2019. p. 213–49.
[8] Autoren (Nachname VV). Injection molding handbook. 2th ed. New York, NY, USA: Springer US; 2012.
[9] Cao Y, Fan X, Guo Y, Li S, Huang H. Multi-objective optimization of injection-molded plastic parts using entropy weight, random forest, and genetic algorithm methods. J Polym Eng 2020;40:360–71. https://doi.org/10.1515/polyeng-2019-0326.
[10] Bensingh RJ, Machavaram R, Boopathy SR, Jebaraj C. Injection molding process optimization of a bi-aspheric lens using hybrid artificial neural networks (ANNs) and particle swarm optimization (PSO). Measurement 2019;134:359–74. https://doi.org/10.1016/j.measurement.2018.10.066.
[11] Bourdon R, Hellmann A, Schreckenberg J-B, Schwegmann R. Standardisierte prozess- und Qualitätsoptimierung mit DOE-methoden - eine kurzanleitung für die praxis beim Spritzgießen. Z Kunstgesch 2012;8:525–49.
[12] Giordano G. Buying power. Plast Eng 2019;75:28–35. https://doi.org/10.1002/peng.20056.
[13] Schiffers R. Verbesserung der Prozessfähigkeit beim Spritzgießen durch nutzung von prozessdaten und eine neuartige Schneckenhubführung. In. Universität Duisburg-Essen; 2009.
[14] Lockner Y, Hopmann C. Induced network-based transfer learning in injection molding for process modelling and optimization with artificial neural networks. Int J Adv Manuf Technol 2021;112:3501–13. https://doi.org/10.1007/s00170-020-06511-3.
[15] Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. Nature 1986;323:533–6.
[16] Autoren (Nachname VV). Hands-on machine learning with Scikit-Learn & Tensorflow. Sebastopol, CA, United States of America: O'Reilly; 2017.
[17] Tsai K-M, Luo H-J. An inverse model for injection molding of optical lens using artificial neural network coupled with genetic algorithm. J Intell Manuf 2014;28:473–87. https://doi.org/10.1007/s10845-014-0999-z.
[18] Iniesta AA, Alcaraz JLG, Borbón MIR. Optimization of injection molding process parameters by a hybrid of artificial neural network and artificial bee colony algorithm. Rev Fac Ingenería Univ Antioquia 2013;67:43–51. Unknown DOI.
[19] Zhang J, Wang J, Lin J, Guo Q, Chen K, Ma L. Multiobjective optimization of injection molding process parameters based on opt LHD, EBFNN, and MOPSO. Int J Adv Manuf Technol 2016;85:2857–72. https://doi.org/10.1007/s00170-015-8100-4.
[20] Nagorny P, Pillet M, Pairel E, Le Goff R, Loureaux Jérome, Wali M, Kiener P. Quality prediction in injection molding. In: 2017 IEEE international conference on computational intelligence and virtual environments for measurement systems and applications, Annecy, Frankreich; 2017.
[21] Jain ESM, Bhuyan RK. Simulation and optimization of warpage of fiber reinforced using human behavior based optimization. Int J Innov Technol Explor Eng 2019;8:296–302. https://doi.org/10.35940/ijitee.I8187.0881019.
[22] Kenig S, Ben-David A, Omer M, Sadeh A. Control of properties in injection molding by neural networks. Eng Appl Artif Intell 2001;14:819–23. https://doi.org/10.1016/S0952-1976(02)00006-4.
[23] Chen W-C, Wang M-W, Chen C-T, Fu G-L. An integrated parameter optimization system for MISO plastic injection molding. Int J Adv Manuf Technol 2009;44:501–11. https://doi.org/10.1007/s00170-008-1843-4.
[24] Trovalusci F, Ucciardello N, Baiocco G, Tagliaferri F. Neural network approach to quality monitoring of injection molding of photoluminescent polymers. Appl Phys A Mater Sci Process 2019;125:781–7. https://doi.org/10.1007/s00339-019-3067-x.
[25] Patel GCM, Krishna P. Prediction and optimization of dimensional shrinkage variations in injection molded parts using forward and reverse mapping of artificial neural networks. Adv Mat Res 2012;463–464:674–8. https://10.4028/www.scientific.net/AMR.463-464.674.
[26] Shi F, Lou ZL, Lu JG, Zhang YQ. Optimisation of plastic injection moulding process with soft computing. Int J Adv Manuf Technol 2003;21:656–61.
[27] Sedighi R, Meiabadi MS, Sedighi M. Optimisation of gate location based on weld line in plastic injection moulding using computer-aided engineering, artificial neural network, and genetic algorithm. Int J Automot Mech Eng 2017;14:4419–31. https://doi.org/10.15282/ijame.14.3.2017.3.0350.
[28] Yarlagadda PKDV. Prediction of processing parameters for injection moulding by using a hybrid neural network. Proc Inst Mech Eng 2001;215:1465–70. https://doi.org/10.1243/0954405011519097.
[29] Spina R. Optimisation of injection moulded parts by using ANN-PSO approach. J Achiev Mater Manuf Eng 2006;15:146–52. unknown DOI.
[30] Lee H, Liau Y, Ryu K. Real-time parameter optimization based on neural network for smart injection molding. IOP Conf. Ser Mater Sci Eng 2017;324:012076. https://doi.org/10.1088/1757-899X/324/1/012076.
[31] Weiss K, Khoshgoftaar TM, Wang D. A survey of transfer learning. J Big Data 2016;3:1–40. https://doi.org/10.1186/s40537-016-0043-6.
[32] Rosenstein MT, Marx Z, Kaelbling LP, Dietterich TG. To transfer or not to transfer. In: Inductive transfer: 10 years later - NIPS 2005 workshop, Whistler, Kanada; 2005.
[33] Torrey L, Shavlik J. Transfer Learning. In: Olivas ES, Guerrero JDM, Sober MM, Benedito J, Magdalena Rafael, Lopez AJS, editors. Handbook of research on machine learning applications and trends: algorithms, methods and techniques. Hershey U S O A: IGI Global; 2009. p. 242–64.
[34] Pan SJ, Yang Q. A survey on transfer learning. IEEE Trans Knowl Data Eng 2010;22:1345–59. https://doi.org/10.1109/TKDE.2009.191.
[35] Ciresan DC, Meier U, Schmidhuber J. Transfer Learning for Latin and Chinese Characters with Deep Neural Networks. In: IEEE World Congress on Computational Intelligence, Brisbane, Australien; 2012.
[36] Li B, Yang Q, Xue X. Transfer learning for collaborative filtering via a rating-matrix generative model. In: 26th international conference on machine learning, Montreal, Kanada; 2009.
[37] Collobert R, Weston J. A unified architecture for natural language processing: deep neural networks with multitask learning. In: 25th international conference on machine learning, Helsinki, Finnland; 2008.
[38] Yosinski J, Clune J, Bengio Y, Lipson H. How transferable are features in deep neural networks?. In: NIPS'14 proceedings of the 27th international conference on neural information processing systems - volume 2, Montreal, Kanada; 2014.
[39] Bengio Y. Deep learning of representations for unsupervised and transfer learning. In: JMLR workshop and conference proceedings. 7; 2011. p. 1–20.
[40] Tercan H, Guajardo A, Heinisch J, Thiele T, Hopmann C, Meisen T. Transfer-learning: bridging the gap between real and simulation data for machine learning in injection molding. Procedia CIRP 2018;72:185–90. https://doi.org/10.1016/j.procir.2018.03.087.
[41] Simulation of fluid flow and structural analysis within thin walled three dimensional geometries. In: Simcon kunststofftechnische Software GmbH, 52134 Herzogenrath (DE). European Patent Office; 2004.
[42] Osswald TA, Rudolph N. Generalized Newtonian fluid (GNF) models. In: Osswald TA, editor. Polymer rheology. Munich C: Hanser Verlag; 2015. p. 59–99.
[43] Cadmould 3D-F user manual. In, simcon kunststofftechnische Software GmbH.
[44] Heinisch J, Lockner Y, Hopmann C. Comparison of design of experiment methods for modeling injection molding experiments using artificial neural networks. J Manuf Process 2021;61:357–68. https://doi.org/10.1016/j.jmapro.2020.11.011.
[45] Tercan H, Guajardo A, Heinisch J, Thiele T, Hopmann C, Meisen T. Transfer-learning: bridging the gap between real and simulation data for machine learning in injection molding. In: 51st CIRP conference on manufacturing systems, Stockholm, Schweden; 2018.
[46] Chen T, Guestrin C. XGBoost: a scalable tree boosting system. In: 22nd ACM SIGKDD international conference on knowledge discovery and data mining, San Francisco, CA, USA; 2016.
[47] Geurts P, Ernst D, Wehenkel L. Extremely randomized trees. Mach Learn 2006;63:3–42. https://doi.org/10.1007/s10994-006-6226-1.
[48] Loh W-Y. Classification and regression trees. WIREs Data Min Knowl Discovery 2011;1:14–23. https://doi.org/10.1002/widm.8.
[49] Hastie T, Tibshirani R, Friedman J. Boosting and additive trees. In: The elements of statistical learning. New York N, USA: Springer; 2008. p. 337–87.
[50] Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci 1997;55:119–39.
[51] Breiman L. Random forests. Mach Learn 2001;45:5–32. https://doi.org/10.1023/A:1010933404324.
[52] Scitkit-Learn. 2021.
[53] Zheng H, Yang Z, Liu W-J, Liang J, Li Y. Improving deep neural networks using softplus units. In: 2015 international joint conference on neural networks (IJCNN), Killarney, Irland; 2015.
[54] Aha DW, Molineaux M, Sukthankar G. Case-based reasoning in transfer learning. In: Wilson DC, McGinty L, editors. Case-based reasoning research and development. Berlin H: Springer; 2009. p. 29–44.
[55] Goel A, Diaz-Agudo B. What's hot in case-based reasoning. In: Proceedings of the AAAI conference on artificial intelligence. 31; 2017. p. 5067–9.
[56] Klenk M, Aha DW, Molineaux M. The case for case-based transfer learning. AI Mag 2011;32:54–69. https://doi.org/10.1609/aimag.v32i1.2331.

Yannik Lockner studied business engineering in bachelor and master at the RWTH Aachen University in Germany with a specialization in plastics processing from 2012 until 2018. After his graduation, he began doing a doctorate at the Institute for Plastics Processing (IKV) in Industry and the Skilled Crafts in 2019. In his research he focuses on the process control and process set-up for injection molding machines utilizing mainly data-based transfer learning approaches.

Christian Hopmann holds the Chair for Plastics Processing and is director of the industry based IKV – Institute for Plastics Processing in Industry and Crafts at RWTH Aachen University. He is co-founder of the AZL – Aachen Center for Lightweight Production and vice-dean of the faculty for mechanical engineering. After studying mechanical engineering, he received his doctoral degree from RWTH Aachen University/Germany and participated the program for executive development at the International Institute for Management Development (IMD), Lausanne/Switzerland. Following a senior vice-director position at IKV, he started his industrial career in 2005 at the plastics processing company RKW SE, mainly as head of production and latterly as managing director of RKW Sweden A.B. in Helsingborg/Sweden.

Weibo Zhao studied General Mechanical Engineering at the RWTH Aachen University with a specialization in automatization and control. His interests include machine learning, plastics processing, automatic control and cyber-physical production systems with an interdisciplinary focus.