

Optimizing Hybrid Recommender Systems via a Constrained Regularized Loss

Stefan Scholbe, Mateo Diaz-Bone, and Philip Toma
Department of Computer Science, ETH Zurich, Switzerland
Group: polymensa_staff

Abstract—The present work pertains to item recommender systems. By means of experimenting with a movie rating dataset, we show that it is possible to predict user preferences for movies from prior knowledge of disjoint user preferences by means of combining different autoencoder neural networks, factorization machines and further collaborative filtering (CF) techniques in an ensemble. We introduce a task-specific optimization objective to discover the per-user utility of ensemble base-predictors for user-preference prediction and evaluate how this objective impacts model performance.

I. INTRODUCTION

Recommender systems have been around for a long time. They came with the emergence of e-commerce, found widespread application in personalised online-streaming platforms and have even found their way into drug repurposing models [1]. The existence of challenges such as the Booking.com challenge in the year 2021 [2] shows: recommender systems research is not a thing of the past, it is alive well.

Due to the widespread use of recommender systems, a diverse set of models for different tasks have been developed: while collaborative recommendation approaches aim to predict item ratings from a sparse user-item-matrix, content based recommenders take into account item-specific features and knowledge based recommenders take into account prior knowledge about user demographics.

Differences in the level of model sophistication (e.g. amount of information used, computational power required) are manifold. Collaborative filtering methods are popular in sparse data settings with little to no prior knowledge about users and items (i.e. limited access to user-item interaction data on a platform). Here, we focus on collaborative filtering (CF) methods, given that our data is comprised **only** of user ratings for items in the database.

Progress made in recent years regarding neural networks (NNs) has lead to a substantial growth of applications with an underlying NN model. Recommender systems have not been spared. Models such as CRANet [3] and ReDa [4] represent examples of autoencoders (NN architecture) in a CF setting, with Kuchaiev and Ginsburg of NVIDIA laying the groundwork for an efficient and practical application of autoencoders in recommendation-dependent industries [5].

A. Related work

a) *Dimension Reduction*: The fundamental idea of dimension reduction is that a small number of continuous latent variables is the underlying cause for the observed data (in our setting: the user-item matrix). Simply put: dimension reduction refers to a process which defines a mapping from the high dimensional input space to a lower dimensional

latent space. Prominent examples are principle component analysis (PCA) and singular value decomposition (SVD).

b) *Item-Based CF Recommenders*: In their seminal paper, Sarwar et al. introduce the item-based CF recommender system approach [6]. They evaluate various similarity metrics, taking into account that users tend to be biased when rating items on a platform. Such bias also exists in the dataset at our disposal.

In their model, the authors calculate the similarity of items, and using a nearest-neighbors approach, calculate a predicted item rating, given a user, using a weighted sum approach.

c) *Hybrid Recommenders*: A majority of published recommender system methods utilize a single, underlying recommendation mechanism. Research into ensemble models in recommender systems, i.e. shifting from a unimodal model approach to so-called *hybrid designs* where multiple base-predictors are combined for the final prediction, has shown promising results. Here, two subgroups exist: parallelized hybrid designs and pipelined hybrid designs.

Parallelized hybrid designs of recommender systems are equivalent to the idea of ensemble methods (from the machine learning community): the predictive power of multiple recommender system architectures is combined to improve generalizability over that of a single (weak) recommender. Pipelined designs, on the other hand, are similar to the idea of refinement models, where different models are used sequentially to refine predictions of previous models.

In this work, we focus on parallelized architectures, which are promising especially in settings where base-predictors disagree regarding predictions of user subgroups [7].

d) *Quadratic Programming*: Quadratic programming (QP) refers to the formal procedure of solving a constrained mathematical problem. In general, the goal is to minimize a function f given constraints on the variables present in the function.

For our method, we employ the sequential least squares programming solution (SLSQP) approach. While this approach has seen its occasional use in ensemble methods (see [8]), it is far from a standard procedure.

B. Contributions

While the ground work for hybrid recommender systems stands, optimal training procedures which best allow generalisation, coupled with model interpretability remains an open question. In this work we present an ensemble model, which through our constrained regularized optimization objective, aims to solve both problems.

II. MODELS AND METHODS

A. Baseline Models

To evaluate our model performance, we employed various baseline methods. Our baseline evaluation consists of following models:

- Mean-Impute, where a mean rating is calculated either per user or item, and the corresponding user or item is imputed by this value.
- Low-rank SVD matrix factorization (denoted as SVD in the following) on the item-based mean-imputed rating matrix, as described in [9].
- Funk SVD, popularized during the Netflix prize and described in [10].

B. Ensemble Sub-Models

Our proposed ensemble model takes into account various recommender system methods, including the mean-impute method and the low-rank SVD approximation method (see II-A). In addition to the baseline methods we include two groups of models: variations of autoencoders and variations of the SVD++flipped matrix factorization model.

1) *Autoencoders*: We experimented with various variations of a generic autoencoder. The base autoencoder consists of 5 hidden layers with 12 nodes each in combination with an ELU activation function σ after each layer. During the training process, we employ a dropout of 50% at the first layer. All following variations of the autoencoder extend the base autoencoder.

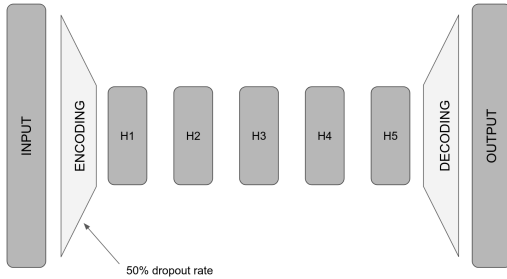


Fig. 1: Visualization of the general AE architecture. Here, H_i refers to hidden layer i .

a) *Bias AE*: To account for user bias in ratings, the network learns scalar factors z_u for each user u . This value is added to the output of the final layer of the autoencoder, i.e. the network predicts

$$\hat{P}(u) = \sigma(W_k \cdot F_{k-1}(x_u) + b_k) + z_u.$$

b) *Residual AE*: To better propagate information through the network, we use skip connections between layers. Intuitively this means, that a layer k is defined as

$$F_k(x) = \sigma(W_k \cdot F_{k-1}(x) + b_k) + F_{k-1}(x)$$

with $F_0(x) = x$ and where b_k is the bias of layer k .

c) β -VAE: The assumption of probabilistic latent variable models is, that the underlying latent features of the observed data are defined by a distribution, which can be inferred. Variational autoencoders (VAEs) aim to learn this distribution by simultaneously minimizing the reconstruction loss and the KL divergence between prior and

learned distribution [11]. β -VAEs introduce a hyperparameter β to regulate the impact of the KL term.

During model evaluation it became apparent, that the model performed best when setting $\beta = 0$, implying that the KL divergence term does not affect the overall loss. The performance is therefore best, when the learned distribution overfits towards the training data, ignoring the regulatory effect of the KL divergence.

d) *Cluster AE*: Here, we aim to take into account prior information about items. Due to missing knowledge about items, we aim to find a latent distribution underlying the observed data. To do this, we perform dimension reduction (PCA with 25 principal components) followed by another dimension reduction (UMAP [12]) on the zero-imputed observed data. Using this procedure, we can visually detect

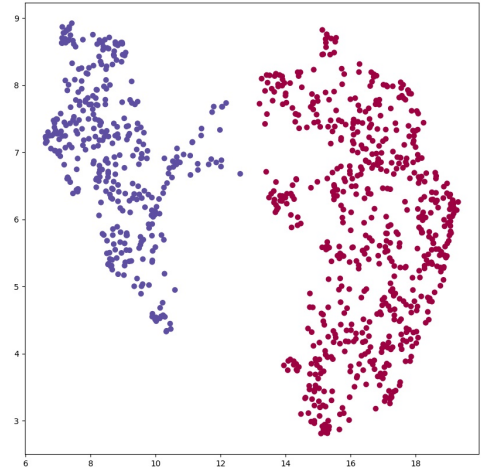


Fig. 2: Item-clusters as discovered by PCA+UMAP dimension reduction.

that two clusters of items are discovered (see Figure 2). To incorporate this prior information into the AE model, cluster labels are encoded into the output layer by adding the cluster label to the output layer. Similar to Bias AE, we learn an item-specific bias, however this time one bias per cluster, which is then selectively added to the specific output ratings.

2) *SVD++flipped*: We experimented with various matrix factorization models, including regular low-rank singular value decomposition (SVD), Funk SVD and SVD++ [13] which takes into account implicit feedback of what movies the user rated as well. However all these methods performed rather poorly in comparison to our variations of SVD++flipped [14] which takes into account implicit feedback about other movies (like SVD++) as well as implicit feedback about other users. Using feature engineering, we were able to mimic the SVD++flipped problem formulation as a Factorization Machine [15]. To solve the FM problems, we use the *myFM*¹ library by Tomoki Ohtsuki for Python. All models use an embedding size of 18.

a) *(Gibbs) Regression FM*: For a typical linear regression with (two-way²) interactions, we model our target $y \in \mathbb{R}$ given the explanatory variables $\mathbf{x} \in \mathbb{R}^N$ as

¹<https://github.com/tohtsky/myFM>

²For brevity, we use two-way interactions. Both FMs and Bayesian FMs can be generalized to N-way interactions.

$$y \sim w_0 + \sum_i w_i x_i + \sum_{ij} w_{ij} x_i x_j$$

Since the number of coefficients w_{ij} increases quadratically with the number of features in \mathbf{x} , solving such a regression becomes challenging, especially in the case of SVD++flipped.

Factorization Machines approximate w_{ij} as a dot product between two latent vectors $\mathbf{v}_i, \mathbf{v}_j \in \mathbb{R}^K$ with $K \ll N$, i. e., $w_{ij} \approx \mathbf{v}_i \cdot \mathbf{v}_j$. Instead of determining $N \times N$ parameters, we only need to estimate $N \times K$ parameters. We usually include some form of regularization (e. g., L2) on the coefficients.

In practice, FMs are typically trained in batches. However, training is sensitive to hyperparameters such as the learning rate and the regularization. Bayesian Factorization Machines (BFM) [16], which we use in our Regression FM model, enhance FMs with structured Bayesian inference. Here, hierarchical hyperpriors regularize the underlying model parameters. As posterior inference due to the complexity of the model is usually intractable, BFMs rely on Gibbs sampling to draw from the posterior.

b) Variational FM: Variational FMs [17] extend the concept of FMs by introducing priors on the regression coefficients themselves, meaning $w_i \sim \mathcal{N}(\mu_w, \sigma_w)$ and $\mathbf{v}_i \sim \mathcal{N}(\mu_v, \sigma_v)$. The latent vectors \mathbf{v}_i are therefore drawn from a multivariate normal distribution.

c) Ordered Probit FM: In an ordered probit model, the idea is that there is a latent continuous metric underlying the ordinal responses (in our case ratings 1 to 5). We partition the real line into regions corresponding to the categories. We model the intermediate y^* as a regression seen in the Regression FM, however we decide on a rating r based on

$$y = r \iff \mu_{r-1} < y^* < \mu_r$$

This adds new parameters μ_0 to μ_5 into the equation. For a more detailed explanation, we refer to [18].

C. Hybrid Recommender (Ensemble)

Our ensemble model works by generating a per-user weighted sum of predictions. Such a weighting is often learned by a linear or logistic regression. We found that these models worked in parts, however non-sensible predictions (such as negative ratings) occurred in a regular frequency, thereby generating the need for a better combination mechanism. In comparison to classic regression models, our proposed ensemble model provides

- **Greater resistance** to overfitting, due to a regularization term that combats model imbalance.
- **Better interpretability** of the results, as the resulting weights satisfy the properties of a probability mass function (e. g., no negative weights).
- **User-specific predictions** to exploit the strengths of individual predictors.
- **Range preservation**, meaning the ratings remain in the interval $[1, 5]$.

Given the base predictors $P_1(u, i)$ to $P_N(u, i)$ and ground truth $P^*(u, i)$ for a user u and an item i , we obtain our ensemble predictor $\hat{P}(i)$ for a specific user u by solving the constrained, non-linear optimization problem 1. We use the

Sequential Least Squares Programming (SLSQP) solver to obtain one predictor per user in the system.

Algorithm 1 User-Specific Ensemble Predictor

Minimize the user-specific loss w.r.t. $\mathbf{w} \in \mathbb{R}^N$:

$$L(\mathbf{w}) = \frac{1}{|I_u|} \sum_{i \in I_u} (P^*(u, i) - \hat{P}(i; \mathbf{w}))^2 + \frac{\lambda}{N} \sum_{k=1}^N (w_k - \bar{\mathbf{w}})^2$$

subject to $w_k \geq 0$ for $1 \leq k \leq N$ and $\sum_{k=1}^N w_k = 1$, where I_u is the set of items the user has rated, $\bar{\mathbf{w}}$ denotes the mean of the weights, and

$$\hat{P}(i; \mathbf{w}) = \sum_{k=1}^N w_k P_k(u, i)$$

is our weighted predictor for the user u and item i .

The first term in $L(\mathbf{w})$ is the mean squared reconstruction error on the known ratings to ensure high-quality predictions. The second term is a regularization on the variance of the weights controlled by λ . It is intended to prevent predictors that reconstruct too well from making the prediction alone, therefore reducing the risk of overfitting. For large λ , the solution $\hat{\mathbf{w}}$ becomes uniform.

From a statistical standpoint, the weights satisfy the properties of a probability mass function $p(w)$. If w_k specifies the probability that P_k provides the correct predictions for a certain user, then

$$\hat{P} = \mathbb{E}_{p(w)}[P]$$

meaning we impute using the mean of the base predictors according to their degree of correctness. Our optimization objective thus is equivalent to

$$\arg \min_{p(w)} \mathbb{E}[(P^* - \hat{P})^2] + \lambda \cdot \text{Var}[W]$$

assuming the probability of a user rating a specific movie is uniform. We can then interpret the first term in our objective as finding the best approximation of P^* regardless of the uncertainty about the error induced by overfitted models. The second term mitigates the uncertainty by slightly pushing towards a more realistic uniform distribution.

III. RESULTS

In this section, we present the comparison of our ensemble model and base-predictors to the baseline models defined in section II. The ensemble contains all models except Funk-SVD and User Mean.

a) Validation Score: To validate our model predictions, we perform 10-fold cross validation. To avoid data leakage, validation sets are not seen by the base-predictors or by the ensemble model during training. To achieve this, 10% of the initially known ratings are masked and treated as ground truth.

For masked ratings $P^*(u, i) \in D_{\text{mask}}$ and predicted ratings $\hat{P}(u, i)$, the validation score is defined as the RMSE

$$\text{RMSE} = \sqrt{\frac{1}{|D_{\text{mask}}|} \cdot \sum_{(u, i) \in D_{\text{mask}}} (P^*(u, i) - \hat{P}(u, i))^2}$$

TABLE I: Performance wrt. Baseline Models

	Cross-Validation Score	Public Score
User Mean	1.09458	1.09267
Item Mean	1.03037	1.02982
SVD	1.00744	1.00431
Funk SVD	1.00089	0.99694
β -VAE	0.98496	0.98335
Residual AE	0.98062	0.97969
Cluster AE	0.98021	0.97933
Bias AE	0.98136	0.97928
Variational FM	0.97480	0.97110
Regression FM	0.97084	0.96785
Ordered Probit FM	0.96837	0.96606
Ensemble Model	0.96816	0.96561

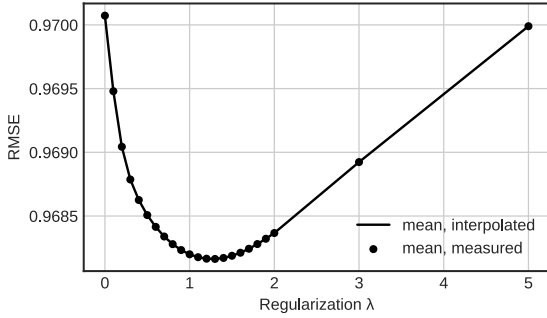


Fig. 3: Cross-validation score of the ensemble with increasing regularization parameter λ . The minimum is located around $\lambda = 1.3$ with score 0.96816 and is used in our submission.

Public scores, as calculated during the competition, are the RMSE for predictions where the ground-truth was not known to us.

From our evaluation (see Table I) three things become apparent:

- The modified autoencoder architectures, based on both the public and validation score, outperform our **baseline models**.
- Using models in an ensemble yields the best results, both based on the validation and the public score.
- Regularization in our ensemble ($\lambda = 1.3$) improves model performance (compare $\lambda = 0$).

IV. DISCUSSION

A. Ensemble

We were able to optimize model combinations through quadratic programming with SLSQP. This is not a standard approach in ensemble methods.

The use of quadratic programming allows for the optimization of an objective under a-priori-known constraints on model parameters. This is a direct benefit over other stacking methods, where model parameters are obtained through a regression (linear regression, [adaptive] lasso) or classification model. This does not allow a similar level of control [19], [20].

Due to the constraints on our ensemble weights, our method allows simple a-posteriori model-analysis, showing which method best attends to a specific user. Importantly, the

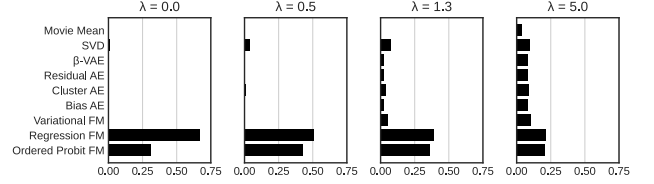


Fig. 4: Mean of the ensemble weights over all users in the system with increasing regularization parameter λ .

non-negativity constraint on model weights has been shown to improve prediction accuracy when combining learned models [21].

While we were able to achieve competitive results using our optimization method, there are open possibilities for future investigation. As is standard-procedure in stacking methods, we optimize model combination weights using previously optimized base predictors. However, Shahhosseini et al. propose that, when computational resources are ubiquitously available, an iterative and coupled optimization process of base-predictors and ensemble weights can improve model performance [22].

The resulting model combination weights can be used to decide, per user, how a model performs in different situations (e. g., non-regular platform user vs. active user).

B. Effects Of Varying λ

Without regularization, the predictions are chosen virtually exclusively by our two best models (Regression FM and Ordered Probit FM). As we increase λ in our ensemble, we see in Fig. 4 that the weights slowly converge to a uniform distribution. What is surprising is that although the influence of our best model is drastically reduced and worse scoring models like SVD gain weight, the change in the validation is comparatively small (Fig. 3). This means that although we do not achieve a significant improvement with our ensemble over the best model, we can identify many more evenly distributed ensembles with comparable scores.

We believe that implementing an additional regularization step that is capable of "toggling" models (similar to ReLU activations) for a specific user could be beneficial. If λ becomes too large, weaker models will have too much influence due to high pressure on the variance of the weights. This likely limits our possible gains. We therefore suggest a model selection step to run before our ensemble regularization.

V. SUMMARY

Our work, building on previous works from ensemble methods and collaborative filtering, showed that optimization via quadratic programming (SLSQP) outperforms other stacking ensemble methods and can be effectively utilized for hybrid recommender systems. Our model decides on a user-to-user basis, how to best weight model predictions.

We showed that a combination of statistical learning and "simpler" models yields state-of-the-art results. Prominently it is visible, that ensemble models improve prediction accuracy. Owing to the fact that data access was limited, it remains to be shown, how our model performs in a classical setting, where data is available in a greater quantity.

REFERENCES

- [1] C. Suphavitai, D. Bertrand, and N. Nagarajan, "Predicting Cancer Drug Response using a Recommender System," *Bioinformatics*, vol. 34, no. 22, pp. 3907–3914, 06 2018. [Online]. Available: <https://doi.org/10.1093/bioinformatics/bty452>
- [2] D. Goldenberg, K. Kofman, P. Levin, S. Mizrachi, M. Kafry, and G. Nadav, "Booking.com wsdm webtour 2021 challenge," in *WebTour@WSDM*, 2021.
- [3] L. Xia, C. Huang, Y. Xu, H. Xu, X. Li, and W. Zhang, "Collaborative reflection-augmented autoencoder network for recommender systems," *ACM Transactions on Information Systems*, vol. 40, no. 1, pp. 1–22, jan 2022. [Online]. Available: <https://doi.org/10.1145%2F3467023>
- [4] F. Zhuang, Z. Zhang, M. Qian, C. Shi, X. Xie, and Q. He, "Representation learning via dual-autoencoder for recommendation," *Neural Networks*, vol. 90, pp. 83–89, 2017. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0893608017300655>
- [5] O. Kuchaiev and B. Ginsburg, "Training deep autoencoders for collaborative filtering," 2017. [Online]. Available: <https://arxiv.org/abs/1708.01715>
- [6] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th International Conference on World Wide Web*, ser. WWW '01. New York, NY, USA: Association for Computing Machinery, 2001, p. 285–295. [Online]. Available: <https://doi.org/10.1145/371920.372071>
- [7] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.
- [8] M. Gupta and B. Gupta, "An ensemble model for breast cancer prediction using sequential least squares programming method (slsqp)," in *2018 Eleventh International Conference on Contemporary Computing (IC3)*, 2018, pp. 1–3.
- [9] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl, "Application of dimensionality reduction in recommender system - a case study," 2000.
- [10] B. Webb, "Netflix update: Try this at home." 2006. [Online]. Available: <https://sifter.org/simon/journal/20061211.html>
- [11] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2013. [Online]. Available: <https://arxiv.org/abs/1312.6114>
- [12] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," 2018. [Online]. Available: <https://arxiv.org/abs/1802.03426>
- [13] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '08. New York, NY, USA: Association for Computing Machinery, 2008, p. 426–434. [Online]. Available: <https://doi.org/10.1145/1401890.1401944>
- [14] —, "Collaborative filtering with temporal dynamics," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '09. New York, NY, USA: Association for Computing Machinery, 2009, p. 447–456. [Online]. Available: <https://doi.org/10.1145/1557019.1557072>
- [15] S. Rendle, "Factorization machines," 2010.
- [16] C. Freudenthaler, "Bayesian factorization machines," 2011.
- [17] J.-J. Vie, "Fast variational learning of factorization machines for large-scale recommender systems," 2019.
- [18] S. Jackman, "Models for ordered outcomes," 2000. [Online]. Available: <https://web.stanford.edu/class/polisci203/ordered.pdf>
- [19] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996. [Online]. Available: <http://www.jstor.org/stable/2346178>
- [20] H. Zou, "The adaptive lasso and its oracle properties," *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1418–1429, 2006. [Online]. Available: <https://doi.org/10.1198/016214506000000735>
- [21] L. Breiman, "Better subset regression using the nonnegative garrote," *Technometrics*, vol. 37, no. 4, pp. 373–384, 1995. [Online]. Available: <http://www.jstor.org/stable/1269730>
- [22] M. Shahhosseini, G. Hu, and H. Pham, "Optimizing ensemble weights and hyperparameters of machine learning models for regression problems," *Machine Learning with Applications*, vol. 7, p. 100251, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2666827022000020>



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Optimizing Hybrid Recommender Systems via a Constrained Regularized Loss

Authored by (in block letters):

For papers written by groups the names of all authors are required.

Name(s):

Toma

Scholbe

Diaz-Bone

First name(s):

Philip

Stefan

Mateo

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the '[Citation etiquette](#)' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

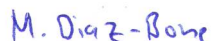
Place, date

Zürich, 31.07.2022

Signature(s)







For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.