RESEARCH NOTE

The role of hyperparameters in machine learning models and how to tune them

Christian Arnold[1] (iD), Luka Biedebach[2], Andreas Küpfer[3] (iD) and Marcel Neunhoeffer[4,5] (iD)

[1]Department of Politics and International Relations, Cardiff University, Cardiff, UK, [2]Department of Computer Science, Reykjavik University, Reykjavik, Iceland, [3]Institute for Political Science, Technical University of Darmstadt, Darmstadt, Germany, [4]Rafik B. Hariri Institute for Computing and Computational Science & Engineering, Boston University, Boston, MA, USA and [5]Department of Statistics, LMU Munich, Munich, Germany
Corresponding author: Marcel Neunhoeffer; Email: marcel@marcel-neunhoeffer.com

Abstract

Hyperparameters critically influence how well machine learning models perform on unseen, out-of-sample data. Systematically comparing the performance of different hyperparameter settings will often go a long way in building confidence about a model's performance. However, analyzing 64 machine learning related manuscripts published in three leading political science journals (APSR, PA, and PSRM) between 2016 and 2021, we find that only 13 publications (20.31 percent) report the hyperparameters and also how they tuned them in either the paper or the appendix. We illustrate the dangers of cursory attention to model and tuning transparency in comparing machine learning models' capability to predict electoral violence from tweets. The tuning of hyperparameters and their documentation should become a standard component of robustness checks for machine learning models.

Keywords: Best Practice; Hyperparameter Optimization; Machine Learning

1 Why care about hyperparameters?

When political scientists work with machine learning models, they want to find a model that generalizes well from training data to new, unseen data.[1] Hyperparameters play a key role in this endeavor because they determine the models' capacity to generalize. Finding a good set of hyperparameters critically affects conclusions about a model's performance. The failure to correctly tune and report hyperparameters has recently been identified as a key impediment to the accumulation of knowledge in computer science (e.g. Henderson *et al.*, 2018; Melis *et al.*, 2018; Bouthillier *et al.*, 2019, 2021; Cooper *et al.*, 2021; Gundersen *et al.*, 2023). Is political science making the same mistake?

We examined 64 machine learning-related papers published between 1 January 2016 and 20 October 2021 in some of the top journals of our discipline—the American Political Science Review (APSR), Political Analysis (PA), and Political Science Research and Methods (PSRM). Of the 64 publications we analyzed, 36 (56.25 percent) do not report the values of their hyperparameters, neither in the paper nor the appendix. Forty-nine publications (76.56 percent) do not share information about how they used tuning to find the values of their hyperparameters.

---

[1]A machine learning algorithm is "a computer program [that is] said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E." (Mitchell, 1997)
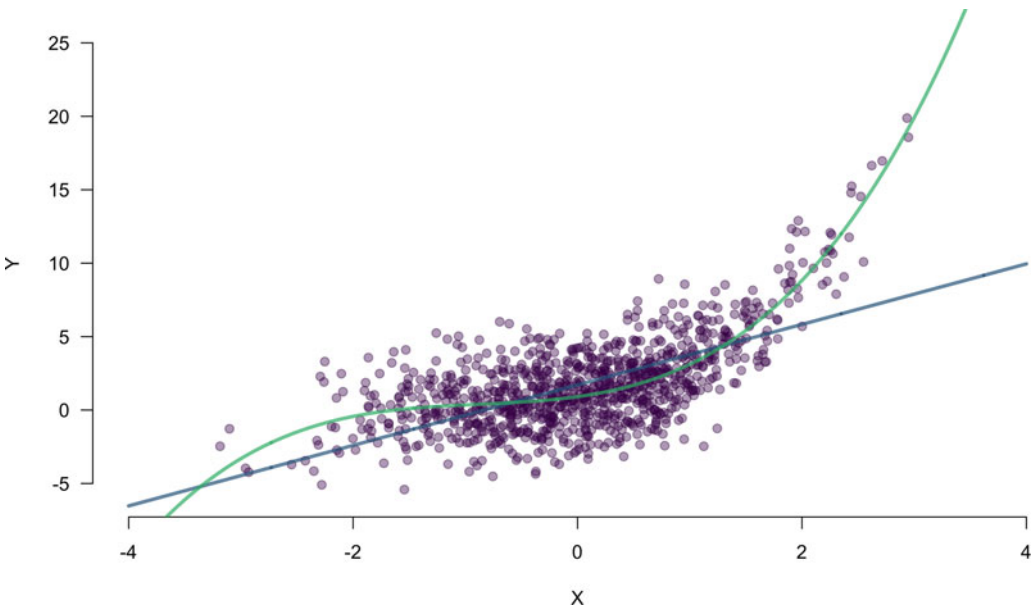
Figure 1. Example with polynomial regression. Data $X \sim N(0, 1)$. Data generating process: $Y = 1 + X + 0.8X^2 + 0.3X^3 + \epsilon$, with $\epsilon \sim N(0, 2)$. Regression Line for Bivariate OLS Model in Blue. Regression Curve for Polynomial Regression with $\lambda = 3$ in Teal.

Only 13 publications (20.31 percent) offer a complete account of the hyperparameters and their tuning. Not being transparent is a dangerous habit because readers and reviewers cannot assess the quality of a manuscript without access to the replication code.

With this paper, therefore, we raise the awareness that hyperparameters and their tuning matter. In statistical inference, the goal is to estimate the value of an unknowable population parameter. Including robustness checks in a paper and its appendix is good practice, allowing others to understand critical choices in research design and statistical modeling. The actual out-of-sample performance of a machine learning model is such an unknown quantity, too. We suggest handling estimates of population parameters and hyperparameters in machine learning models with the same loving care.

First, we explain what hyperparameters are and why they are essential. Second, we show why it is dangerous not to be transparent about hyperparameters. Third, we offer best practice advice about properly selecting hyperparameters. Finally, we illustrate our points by comparing the performance of several machine learning models to predict electoral violence from tweets (Muchlinski *et al.*, 2021).

2 What are hyperparameters and why do they need to be tuned?

Many machine learning models have parameters and also hyperparameters. Model parameters are learned during training, and hyperparameters are typically set before training. Hyperparameters determine how and what a model can learn and how well the model will perform on out-of-sample data. Hyperparameters are thus situated at a meta-level above the models themselves.

Consider the following stylized example displayed in Figure 1.[2] A linear regression approach could model the relationship between X and Y as $\hat{Y} = \beta_0 + \beta_1 X$. A more flexible model would include additional polynomials in X. For example, choosing $\lambda = 2$ encodes the theoretical belief that Y is best predicted by a quadratic function of X, i.e., $\hat{Y} = \beta_0 + \beta_1 X + \beta_2 X^2$. But it is also

---

[2]See also Shalev-Shwartz and Ben-David (2014) and Goodfellow *et al.* (2016).

possible to rely on data only to find the optimal value of λ. Measuring the generalization error with a metric like the mean squared error helps empirically select the most promising value of λ.

This polynomial regression comes with both parameters and hyperparameters. *Parameters* are variables that belong to the model itself, in our example, the regression equation coefficients. *Hyperparameters* are those variables that help specify the exact model. In the context of the polynomial regression, λ is the hyperparameter that determines how many parameters will be learned (Goodfellow *et al.*, 2016). Machine learning models can, of course, come with many more hyperparameters that relate not only to the exact parameterization of the machine learning model. Anything part of the function that maps the data to a performance measure and that can be set to different values can be considered a hyperparameter, e.g., the choice and settings of a kernel in a support vector machine (SVM), the number of trees in a random forest (RF), or the choice of a particular optimization algorithm.

3 Misselecting hyperparameters

Research on machine learning has recently identified several problems that may arise from handling hyperparameters without care. The failure to report the chosen hyperparameters impedes scientific progress (Henderson *et al.*, 2018; Bouthillier *et al.*, 2019, 2021; Gundersen *et al.*, 2023). In the face of a hyperparameter space marked by the curse of dimensionality, other researchers can only replicate published work if they know the hyperparameters used in the original study (Sculley *et al.*, 2018). In addition, it is essential to tune the hyperparameters of all models, including baseline models. Without such tuning, it is impossible to compare the performance of two different models M_a and M_b: While some may find that the performance of M_a is better than M_b, others replicating the study with different hyperparameter settings could conclude the opposite: that indeed M_a is *not* better than that of M_b. Such "hyperparameter deception" (Cooper *et al.*, 2021) has confused scientific progress in various subfields in computer science where machine learning plays a key role, including natural language processing (Melis *et al.*, 2018), computer vision (Musgrave *et al.*, 2020), and generative models (Lucic *et al.*, 2018). Reviewers and readers need to comprehend the hyperparameter tuning to assess whether a new model reliably performs better or whether a study tests new hyperparameters (Cooper *et al.*, 2021).

It is good to see political scientists also discuss and stress the relevance of hyperparameter tuning in their work (e.g., Cranmer and Desmarais, 2017; Fariss and Jones, 2018; Chang and Masterson, 2020; Miller *et al.*, 2020; Rheault and Cochrane, 2020; Torres and Francisco, 2021). But does the broader political science community fulfill the requirements suggested in the computer science literature? To understand how hyperparameters are used in the discipline, we searched for the term "machine learning" in all papers published in APSR, PA, and PSRM after 1 January 2016 and before 20 October 2021. Suppose a paper applies a machine learning model with tunable hyperparameters. In that case, we first annotate whether the authors report the final values of hyperparameters for all models in their paper or its appendix.[3] We also record whether authors transparently describe how they tuned hyperparameters.[4] Table 1 summarizes the findings from our annotations. We find that 34 (53.12 percent) publications neither report the values of the final hyperparameters nor the tuning regime in the publication or its appendix. Another 15 publications (23.44 percent) offer information about the final hyperparameter values but not how they tuned the machine learning models. In two cases (3.12 percent), we find no information about the final values of the hyperparameters but about the tuning regime. Finally, only 13 publications (20.31 percent) offer a full account of both the final choice of the hyperparameters and the way the tuning occurred in either the paper itself or its appendix.

---

[3]We call this "model transparency," i.e., could a reader understand the final models without access to the replication code?

[4]We call this "tuning transparency," i.e., could a reader understand the hyperparameter tuning without access to the replication code? Please see Appendix 1 for more details about our annotations.

Table 1. Can readers of a publication learn how hyperparameters were tuned and what hyperparameters were ultimately chosen? Hyperparameter explanations in papers published in APSR, PA, and PSRM between 1 January 2016 and 20 October 2021

		Tuning transparency	
		No	Yes
Model transparency	No	34	2
	Yes	15	13

   Note that we annotated the literature in a way that helps understand whether reviewers and readers can assess the robustness of the analyses based on the manuscript and its appendix. Our analysis does not consider the replication code since it typically does not find consideration in the review process. In addition, we do not make any judgments about correctness. A paper without information about hyperparameter values or their tuning can still be correct. Similarly, a paper that reports hyperparameter values and a complete account of the tuning can still be wrong. It is the realm of reviewers to evaluate the quality of a manuscript. But without a complete account of hyperparameter values and tuning, readers and, in particular, reviewers cannot judge whether hyperparameter tuning is technically sound.

4 Best practice

Hyperparameters are a fundamental element of machine learning models. Documenting their careful selection helps build trust in the insights gained from machine learning models.

4.1 Selecting hyperparameters for performance tuning

Without automated procedures for finding hyperparameters, researchers need to rely on heuristics (Probst *et al.*, 2019). The classic approach to hyperparameter optimization is to systematically try different hyperparameter settings and compare the models using a performance measure. Machine learning splits the data into training, validation, and test data (Friedman *et al.*, 2001; Goodfellow *et al.*, 2016). The model parameters are optimized using the training data. The validation data is used to optimize the hyperparameters by estimating and then comparing an estimate of the performance of all the different models. Finally, the test data helps approximate the performance of the best model for out-of-sample data. Researchers should train a final machine learning model for a realistic estimate of the model's performance. This model relies upon the identified best set of hyperparameters, uses a combined set of the training and validation data, and is evaluated on the so far withheld test set. Note that this last evaluation can be done only once to avoid information leakage. Tuning hyperparameters is therefore not a form of "p-hacking" (Wasserstein and Lazar, 2016; Gigerenzer, 2018) where researchers try different models until they find the one that generates the desired statistics. On the contrary, transparently testing different hyperparameter values is necessary to find a model that generalizes well.

   In hyperparameter grid search, researchers manually define a grid of hyperparameter values, then try each possible permutation and record the validation performance for each set of hyperparameters. More recently, some instead suggest randomly sampling a large number of hyperparameter candidate values from a pre-defined search space (Bergstra and Bengio, 2012) and recording the validation performance of each set of sampled hyperparameter values.[5] This random search can help explore the space of hyperparameters more efficiently if some

---

[5]How many permutations from the search space should be tried depends on the search space size and the available computational resources.

hyperparameters are more important than others. Both approaches typically yield reliable and good results for practitioners and build trust regarding the out-of-sample performance.

But the tuning of hyperparameters might be too involved for grid or random search in light of resource constraints. It is then useful to not try all combinations of hyperparameters but rather focus on the most promising ones.[6] Sequential model-based Bayesian optimization formalizes such a search for a new candidate set of hyperparameters (Snoek *et al.*, 2012; Shahriari *et al.*, 2016). The core idea is to formulate a surrogate model—think non-linear regression model—that predicts the machine learning model's performance for a set of hyperparameters. At iteration t, the underlying machine learning model is trained with the surrogate model's suggestion for the next best candidate set of hyperparameters. The results from this training at t are fed back into the surrogate model and used to refine the predictions for the candidate set of hyperparameters in the next iteration $t + 1$.[7]

Without a formal solution, the selection of hyperparameters requires human judgment. We suggest relying on the following short heuristics when tuning and communicating hyperparameters.[8]

1. **Understanding the model.** What are the available hyperparameters? How do they affect the model?
2. **Choosing a performance measure.** What is a good performance for the machine learning model? Depending on the respective task, appropriate measures help assess the model's success. For example, a regression model is trained to minimize the mean squared error. Classification models can be trained to maximize the F1 score. With an appropriate performance measure, it is also possible to systematically tune the hyperparameters of unsupervised models (Fan *et al.*, 2020).
3. **Defining a sensible search space.** Useful starting points for the hyperparameters can be the default values in software libraries, recommendations from the literature, or own previous experience (Probst *et al.*, 2019). Any choice may also be informed by considerations about the data-generating process. If the hyperparameters are numerical, there may be a difference between mathematically possible and reasonable values.
4. **Finding the best combination in the search space.** In grid search, researchers should try every possible combination of the hyperparameters of the search space to find the optimal combination. In random search, each run picks a different random set of hyperparameters from the search space.
5. **Tuning under strong resource constraints.** If the model training is too involved, adaptive approaches such as sequential model-based Bayesian optimization allow for efficiently identifying and testing promising hyperparameter candidates.

Researchers should describe in either the main body or the appendix of their publication how they tuned their hyperparameters and also what final values they chose. Only then can reviewers and readers assess the robustness of machine learning models.

4.2 Illustration: Comparing machine learning models to predict electoral violence from tweets

To illustrate our point, we compare machine learning models trained to predict electoral violence from tweets. Muchlinski *et al.* (2021) collected Tweets around elections in three countries

---

[6]For other promising strategies, see the thorough overviews in, e.g., Hutter *et al.* (2015), Luo (2016), Probst *et al.* (2019) and Bischl *et al.* (2023).

[7]Adaptive hyperparameter optimization is conveniently implemented in many software frameworks: for R see, e.g., `mlr3` package on CRAN (Lang *et al.*, 2019), for Python, e.g., `scikit-optimize` (Pedregosa *et al.*, 2011) or `keras` (Chollet *et al.*, 2015).

[8]See also Sculley *et al.* (2018), Bouthillier *et al.* (2021) and Cooper *et al.* (2021).

Table 2. Performance benchmarking of Muchlinski *et al.* (2021) on different classifiers using our scraped data.

Classifier	Default F1	Tuned F1	Default F1	Tuned F1	Default F1	Tuned F1
	Ghana		**The Philippines**		**Venezuela**	
NB	0.000	0.538	0.000	0.390	0.000	0.308
RF	0.341	0.603	0.400	0.160	0.237	0.479
SVM	0.381	0.727	0.357	0.561	0.080	0.465
CNN	0.632	0.604	0.356	0.500	0.319	0.304

On the left: results with default values for the hyperparameters. On the right: results from tuned hyperparameters

(Ghana, the Philippines, and Venezuela) and annotated whether these messages described occurrences of electoral violence. We re-scraped the data based on the shared Tweet IDs. To predict these occurrences from the content of these Tweets, we use four different machine learning models—a naive Bayes classifier (NB), random forest (RF), a support vector machine (SVM), and a convolutional neural network (CNN).

Table 2 summarizes our results. In the left column of each country, we report the results from training the models with default hyperparameters. On the right, we show the results after hyperparameter tuning.[9] Hyperparameter tuning improves the out-of-sample performance for most machine learning models in our experiment.[10] Table 2 also shows how easy it is to be deceived about the relative performance of different models—if hyperparameters are not properly tuned. The performance gains from tuning are so substantial that most tuned models outperform any other model with default hyperparameters. In the case of Venezuela, for example, comparing a tuned model with all other baseline models at their default hyperparameter settings could lead to different conclusions. Researchers could mistakenly conclude that (a tuned) NB classifier (F1 = 0.308) is at eye-level with a CNN model (F1 = 0.319) and better than any other method; or also that the RF is the better model (F1 = 0.479), or the SVM (F1 = 0.465), or the CNN (F1 = 0.304). In short, model comparisons and model choices are only meaningful if all hyperparameters of all models are systematically tuned and if this tuning is transparently documented.

5 Tuning hyperparameters matters

Hyperparameters critically influence how well machine learning models perform on unseen, out-of-sample data. Despite the relevance of tuned hyperparameters, we found that only 20.31 percent of the papers using machine learning models published in APSR, PA, and PSRM between 2016 and 2021 include information about the ultimate hyperparameter choice and how they were found in the manuscript or the appendix. Furthermore, 34 papers (53.12 percent) neither report the hyperparameters nor their tuning. This is a dangerous habit since handling hyperparameters without care can lead to wrong conclusions about model performance and model choice.

The search for an optimal set of hyperparameters is a vibrant research area in computer science and statistics. For most of the applications in our discipline, acknowledging and discussing how the choice of hyperparameters could influence results in combination with a proper and systematic search for appropriate hyperparameters would go a long way. It would allow others to understand original work, assess its validity, and thus ultimately help build trust in political science that uses machine learning.

---

[9]In line with (Muchlinski *et al.*, 2021), we chose the F1 score as the performance metric. We include details on the tuned hyperparameters, the default values we chose, the search method, the search space for each model, and any random seeds in the Online Appendix.

[10]In cases where hyperparameter tuning does not improve the performance over default hyperparameter values, the default values are closer to the optimal solution than the best-performing hyperparameters from a cross-validation procedure. However, the only way to find this out is through systematic hyperparameter tuning.

References

Bergstra J and Bengio Y (2012) Random search for hyper-parameter optimization. *Journal of Machine Learning Research* **13**, 281–305.

Bischl B, Binder M, Lang M, Pielok T, Richter J, Coors S, Thomas J, Ullmann T, Becker M, Boulesteix A-L, Deng D and Lindauer M (2023) Hyperparameter optimization: foundations, algorithms, best practices, and open challenges. *WIREs Data Mining and Knowledge Discovery* **13**, e1484.

Bouthillier X, Laurent C and Vincent P (2019) Unreproducible research is reproducible. In Chaudhuri K. and Salakhutdinov R. (eds), *Proceedings of the 36th International Conference on Machine Learning*. 09–15 Jun. Long Beach, California, USA, Vol. 97, pp. 725–734.

Bouthillier X, Delaunay P, BronziM, Trofimov A, Nichyporuk B, Szeto J, Sepahvand N, Raff E, Madan K, Voleti V, Kahou SE, Michalski V, Arbel T, Pal C, Varoquaux G and Vincent P (2021) Accounting for variance in machine learning benchmarks. In Smola A, Dimakis A and Stoica I (eds), *Proceedings of Machine Learning and Systems*. Conference Proceedings, Virtual Conference, Vol. 3, pp. 747–769.

Chang C and Masterson M (2020) Using word order in political text classification with long short-term memory models. *Political Analysis* **28**, 395–411.

Chollet F *et al.* (2015) Keras. https://keras.io.

Cooper AF, Lu Y, Forde J and De Sa CM (2021) Hyperparameter optimization is deceiving us, and how to stop it. In Ranzato M, Beygelzimer A, Dauphin Y, Liang P, and Vaughan JW (eds), *Advances in Neural Information Processing Systems*. Curran Associates, Inc, Virtual Conference, Vol. 34, pp. 3081–3095.

Cranmer SJ and Desmarais BA (2017) What can we learn from predictive modeling?. *Political Analysis* **25**, 145–166.

Fan X, Yue Y, Sarkar P and Wang YXR (2020) On hyperparameter tuning in general clustering problems. In Daumé H, III and Singh A (eds), *Proceedings of the 37th International Conference on Machine Learning*. Proceedings of Machine Learning Research, 13–18 Jul. PMLR, Virtual Conference, Vol. 119, pp. 2996–3007.

Fariss CJ and Jones ZM (2018) Enhancing validity in observational settings when replication is not possible. *Political Science Research and Methods* **6**, 365–380.

Friedman J, Hastie T and Tibshirani R (2001) *The Elements of Statistical Learning*. Vol. **1**, New York: Springer Series in Statistics.

Gigerenzer G (2018) Statistical rituals: the replication delusion and how we got there. *Advances in Methods and Practices in Psychological Science* **1**, 198–218.

Goodfellow I, Bengio Y and Courville A (2016) *Deep Learning*. Cambridge: MIT Press.

Gundersen OE, Coakley K, Kirkpatrick C and Gil Y (2023) Sources of irreproducibility in machine learning: a review. arXiv:2204.07610 [cs.LG].

Henderson P, Islam R, Bachman P, Pineau J, Precup D and Meger D (2018) Deep reinforcement learning that matters. In *Proceedings of AAAI'18/IAAI'18/EAAI'18*, AAAI'18/IAAI'18/EAAI'18. New Orleans, Louisiana, USA: AAAI Press.

Hutter F, Lücke J and Schmidt-Thieme L (2015) Beyond manual tuning of hyperparameters. *KI-Künstliche Intelligenz* **29**, 329–337.

Lang M, Binder M, Richter J, Schratz P, Pfisterer F, Coors S, Au Q, Casalicchio G, Kotthoff L and Bischl B (2019) mlr3: a modern object-oriented machine learning framework in R. *Journal of Open Source Software*.

Lucic M, Kurach K, Michalski M, Gelly S and Bousquet O (2018) Are GANs created equal? a large-scale study. In Bengio S, Wallach H, Larochelle H, Grauman K, Cesa-Bianchi N, and Garnett R (eds), *Advances in Neural Information Processing Systems*. Vol. 31, Red Hook, NY, USA: Curran Associates Inc.

Luo G (2016) A review of automatic selection methods for machine learning algorithms and hyper-parameter values. *Network Modeling Analysis in Health Informatics and Bioinformatics* **5**, 1–16.

Melis G, Dyer C and Blunsom P (2018) On the state of the art of evaluation in neural language models. In *6th International Conference on Learning Representations*, Vancouver, Canada.

Miller B, Linder F and Mebane WR (2020) Active learning approaches for labeling text: review and assessment of the performance of active learning approaches. *Political Analysis* **28**, 532–551.

Mitchell TM (1997) *Machine Learning*. McGraw-Hill International Edn. New York City, USA: McGraw-Hill.

Muchlinski D, Yang X, Birch S, Macdonald C and Ounis I (2021) We need to go deeper: measuring electoral violence using convolutional neural networks and social media. *Political Science Research and Methods* **9**, 122–139.

Musgrave K, Belongie S and Lim S-N (2020) A metric learning reality check. In Vedaldi A, Bischof H, Brox T and Frahm J-M (eds), *Computer Vision – ECCV 2020*. Cham: Springer International Publishing, pp. 681–699.

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M and Duchesnay E (2011) Scikit-learn: machine learning in Python. *Journal of Machine Learning Research* **12**, 2825–2830.

Probst P, Boulesteix A-L and Bischl B (2019) Tunability: importance of hyperparameters of machine learning algorithms. *Journal of Machine Learning Research* **20**, 1–32.

Rheault L and Cochrane C (2020) Word embeddings for the analysis of ideological placement in parliamentary corpora. *Political Analysis* **28**, 112–133.

Sculley D, Snoek J, Wiltschko A and Rahimi A (2018) Winner's curse?. On pace, progress, and empirical rigor.

Shahriari B, Swersky K, Wang Z, Adams RP and de Freitas N (2016) Taking the human out of the loop: a review of Bayesian optimization. *Proceedings of the IEEE* **104**, 148–175.

Shalev-Shwartz S and Ben-David S (2014) *Understanding Machine Learning: From Theory to Algorithms*. Cambridge: Cambridge University Press.

Snoek J, Larochelle H and Adams RP (2012) Practical Bayesian optimization of machine learning algorithms. In Pereira F, Burges C, Bottou L and Weinberger K (eds), *Advances in Neural Information Processing Systems*. Vol. 25, Red Hook, NY, USA: Curran Associates Inc.

Torres M and Francisco C (2021) Learning to see: convolutional neural networks for the analysis of social science data. *Political Analysis* **30**, 1–19.

Wasserstein RL and Lazar NA (2016) The ASA statement on p-values: context, process, and purpose. *The American Statistician* **70**, 129–133.