# Showing Your Work Doesn't Always Work

**Raphael Tang,**[1,2] **Jaejun Lee,**[1] **Ji Xin,**[1,2] **Xinyu Liu,**[1]
**Yaoliang Yu,**[1,2] and **Jimmy Lin**[1,2]
[1]David R. Cheriton School of Computer Science, University of Waterloo
[2]Vector Institute for Artificial Intelligence
firstname.lastname@uwaterloo.ca

## Abstract

In natural language processing, a recently popular line of work explores how to best report the experimental results of neural networks. One exemplar publication, titled "Show Your Work: Improved Reporting of Experimental Results" (Dodge et al., 2019), advocates for reporting the expected validation effectiveness of the best-tuned model, with respect to the computational budget. In the present work, we critically examine this paper. As far as statistical generalizability is concerned, we find unspoken pitfalls and caveats with this approach. We analytically show that their estimator is biased and uses error-prone assumptions. We find that the estimator favors negative errors and yields poor bootstrapped confidence intervals. We derive an unbiased alternative and bolster our claims with empirical evidence from statistical simulation. Our codebase is at https://github.com/castorini/meanmax.

## 1 Introduction

Questionable answers and irreproducible results represent a formidable beast in natural language processing research. Worryingly, countless experimental papers lack empirical rigor, disregarding necessities such as the reporting of statistical significance tests (Dror et al., 2018) and computational environments (Crane, 2018). As Forde and Paganini (2019) concisely lament, *explorimentation*, the act of tinkering with metaparameters and praying for success, while helpful in brainstorming, does not constitute a rigorous scientific effort.

Against the crashing wave of explorimentation, though, a few brave souls have resisted the urge to feed the beast. Reimers and Gurevych (2017) argue for the reporting of neural network score distributions. Gorman and Bedrick (2019) demonstrate that deterministic dataset splits yield less robust results than random ones for neural networks. Dodge et al. (2019) advocate for reporting the expected validation quality as a function of the computation budget used for hyperparameter tuning, which is paramount to robust conclusions.

But carefully tread we must. Papers that advocate for scientific rigor must be held to the very same standards that they espouse, lest they birth a new beast altogether. In this work, we critically examine one such paper from Dodge et al. (2019). We acknowledge the validity of their technical contribution, but we find several notable caveats, as far as statistical generalizability is concerned. Analytically, we show that their estimator is negatively biased and uses assumptions that are subject to large errors. Based on our theoretical results, we hypothesize that this estimator strongly prefers underestimates to overestimates and yields poor confidence intervals with the common bootstrap method (Efron, 1982).

Our main contributions are as follows: First, we prove that their estimator is biased under weak conditions and provide an unbiased solution. Second, we show that one of their core approximations often contains large errors, leading to poorly controlled bootstrapped confidence intervals. Finally, we empirically confirm the practical hypothesis using the results of neural networks for document classification and sentiment analysis.

## 2 Background and Related Work

**Notation.** We describe our notation of fundamental concepts in probability theory. First, the cumulative distribution function (CDF) of a random variable (RV) $X$ is defined as $F(x) := \Pr[X \leq x]$. Given a sample $(x_1, \ldots, x_B)$ drawn from $F$, the empirical CDF (ECDF) is then $\hat{F}_B(x) := \frac{1}{B}\sum_{i=1}^{B} \mathbb{I}[x_i \leq x]$, where $\mathbb{I}$ denotes the indicator function. Note that we pick "$B$" instead of "$n$" to be consistent with Dodge et al. (2019). The error of the ECDF is pop-

*Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2766–2772
July 5 - 10, 2020. ©2020 Association for Computational Linguistics

ularly characterized by the Kolmogorov–Smirnov (KS) distance between the ECDF and CDF:

$$\mathrm{KS}(\hat{F}_B, F) := \sup_{x \in \mathbb{R}} |\hat{F}_B(x) - F(x)|. \tag{2.1}$$

Naturally, by definition of the CDF and ECDF, $\mathrm{KS}(\hat{F}_B, F) \leq 1$. Using the CDF, the expectation for both discrete and continuous (cts.) RVs is

$$\mathbb{E}[X] = \int_{-\infty}^{\infty} x \mathrm{d}F(x), \tag{2.2}$$

defined using the Riemann–Stieltjes integral.

We write the $i^{\text{th}}$ order statistic of independent and identically distributed (i.i.d.) $X_1, \dots, X_B$ as $X_{(i:B)}$. Recall that the $i^{\text{th}}$ order statistic $X_{(i:B)}$ is an RV representing the $i^{\text{th}}$ smallest value if the RVs were sorted.

**Hyperparameter tuning.** In random search, a probability distribution $p(\mathcal{H})$ is first defined over a $k$-tuple hyperparameter configuration $\mathcal{H} := (H_1, \dots, H_k)$, which can include both cts. and discrete variables, such as the learning rate and random seed of the experimental environment. Commonly, researchers choose the uniform distribution over a bounded support for each hyperparameter (Bergstra and Bengio, 2012). Combined with the appropriate model family $\mathcal{M}$ and dataset $\mathcal{D} := (\mathcal{D}_T, \mathcal{D}_V)$—split into training and validation sets, respectively—a configuration then yields a numeric score $V$ on $\mathcal{D}_V$. Finally, after sampling $B$ i.i.d. configurations, we obtain the scores $V_1, \dots, V_B$ and pick the hyperparameter configuration associated with the best one.

## 3 Analysis of Showing Your Work

In "Show Your Work: Improved Reporting of Experimental Results," Dodge et al. (2019) realize the ramifications of underreporting the hyperparameter tuning policy and its associated budget. One of their key findings is that, given different computation quotas for hyperparameter tuning, researchers may arrive at drastically different conclusions for the same model. Given a small tuning budget, a researcher may conclude that a smaller model outperforms a bigger one, while they may reach the opposite conclusion for a larger budget.

To ameliorate this issue, Dodge et al. (2019) argue for fully reporting the expected maximum of the score as a function of the budget. Concretely, the parameters of interest are $\theta_1, \dots, \theta_B$, where $\theta_n := \mathbb{E}\left[\max\{V_1, \dots, V_n\}\right] = \mathbb{E}[V_{(n:n)}]$ for $1 \leq n \leq B$. In other words, $\theta_n$ is precisely the expected value of the $n^{\text{th}}$ order statistic for a sample of size $n$ drawn i.i.d. at tuning time. For this quantity, they propose an estimator, derived as follows: first, observe that the CDF of $V_n^* = V_{(n:n)}$ is

$$\begin{aligned} \Pr[V_n^* \leq v] &= \Pr[V_1 \leq v \wedge \cdots \wedge V_n \leq v] \quad (3.1)\\ &= \Pr[V \leq v]^n, \quad (3.2)\end{aligned}$$

which we denote as $F^n(v)$. Then

$$\theta_n = \mathbb{E}[V_{(n:n)}] = \int_{-\infty}^{\infty} v \mathrm{d}F^n(v). \tag{3.3}$$

For approximating the CDF, Dodge et al. (2019) use the ECDF $\hat{F}_B^n(v)$, constructed from some sample $S := (v_1, \dots, v_B)$, i.e.,

$$\hat{F}_B^n(v) = \left(\hat{F}_B(v)\right)^n = \left(\frac{1}{B}\sum_{i=1}^{B} \mathbb{I}[v_i \leq v]\right)^n. \tag{3.4}$$

The first identity in Eq. (3.4) is clear from Eq. (3.2). Without loss of generality, assume $v_1 \leq \cdots \leq v_B$. To construct an estimator $\hat{\theta}_n$ for $\theta_n$, Dodge et al. (2019) then replace the CDF with the ECDF:

$$\hat{\theta}_n := \int_{-\infty}^{\infty} v \mathrm{d}\hat{F}_B^n(v), \tag{3.5}$$

which, by definition, evaluates to

$$\hat{\theta}_n = \sum_{i=1}^{B} v_i \left(\hat{F}_B^n(v_i) - \hat{F}_B^n(v_{i-1})\right), \tag{3.6}$$

where, with some abuse of notation, $v_0 < v_1$ is a dummy variable and $\hat{F}_B^n(v_0) := 0$. We henceforth refer to $\hat{\theta}_n$ as the **MeanMax** estimator. Dodge et al. (2019) recommend plotting the number of trials on the $x$-axis and $\hat{\theta}_n$ on the $y$-axis.

### 3.1 Pitfalls and Caveats

We find two unspoken caveats in Dodge et al. (2019): first, the MeanMax estimator is statistically biased, under weak conditions. Second, the ECDF, as formulated, is a poor drop-in replacement for the true CDF, in the sense that the finite sample error can be unacceptable if certain, realistic conditions are unmet.

**Estimator bias.** The bias of an estimator $\hat{\theta}$ is defined as the difference between its expectation and its estimand $\theta$: $\mathrm{Bias}(\hat{\theta}) := \mathbb{E}[\hat{\theta}] - \theta$. An estimator is said to be *unbiased* if its bias is zero; otherwise, it is *biased*. We make the following claim:

**Theorem 1.** *Let $V_1, \dots, V_B$ be an i.i.d. sample (of size $B$) from an unknown distribution $F$ on the real line. Then, for all $1 \le n \le B$, $\mathrm{Bias}(\hat{\theta}_n) \le 0$, with strict inequality iff $V_{(1)} < V_{(n)}$ with nonzero probability. In particular, if $n = 1$, then $\mathrm{Bias}(\hat{\theta}_1) = 0$ while if $n > 1$ with $F$ continuous or discrete but non-degenerate, then $\mathrm{Bias}(\hat{\theta}_n) < 0$.*

*Proof.* Let $1 < n \le B$. We are interested in estimating the expectation of the maximum of the $n$ i.i.d. samples:

$$\theta_n := \mathbb{E}[V_{n:n}] = \mathbb{E}[\max\{V_1, \dots, V_n\}].$$

An obvious unbiased estimator, based on the given sample of size $B$, is the following:

$$\hat{U}_n^B := \frac{1}{\binom{B}{n}} \sum_{1 \le i_1 < i_2 < \cdots < i_n \le B} \max\{V_{i_1}, \dots, V_{i_n}\}.$$

This estimator is obviously unbiased since

$$\mathbb{E}[\hat{U}_n^B] = \mathbb{E}[\max\{V_{i_1}, \dots, V_{i_n}\}] = \theta_n,$$

due to the i.i.d. assumption on the sample.

A second, biased estimator is the following:

$$\hat{V}_n^B := \frac{1}{B^n} \sum_{1 \le i_1 \le i_2 \le \cdots \le i_n \le B} \max\{V_{i_1}, \dots, V_{i_n}\}. \tag{3.7}$$

This estimator is only asymptotically unbiased when $n$ is fixed while $B$ tends to $\infty$. In fact, we will prove below that for all $1 \le n \le B$:

$$\hat{V}_n^B \le \hat{U}_n^B, \tag{3.8}$$

with strict inequality iff $V_{(1)} < V_{(n)}$, where $V_{(i)} = V_{(i:B)}$ is defined as the $i^{\text{th}}$ smallest order statistic of the sample. We start with simplifying the calculation of the two estimators. It is easy to see that the following holds:

$$\hat{U}_n^B = \sum_{j=1}^{B} \frac{\binom{j-1}{n-1}}{\binom{B}{n}} V_{(j)},$$

where we basically enumerate all possibilities for $\max\{V_{i_1}, \dots, V_{i_n}\} = V_{(j)}$. By convention, $\binom{m}{n} = 0$ if $m < n$ so the above summation effectively goes from $k$ to $B$, but our convention will make it more convenient for comparison. Similarly,

$$\hat{V}_n^B = \sum_{j=1}^{B} \frac{j^n - (j-1)^n}{B^n} V_{(j)}.$$

We make an important observation that connects our estimators to that of Dodge et al. Let $\hat{F}_B(x) = \frac{1}{B}\sum_{i=1}^{B} \mathbb{I}[V_i \le x]$ be the empirical distribution of the sample. Then, the plug-in estimator, where we replace $F$ with $\hat{F}_B$, is

$$\begin{aligned}\hat{\theta}_n^B &= \hat{\mathbb{E}}[\max\{\hat{V}_1, \dots, \hat{V}_n\}], \quad \text{where} \quad \hat{V}_i \overset{iid}{\sim} \hat{F}_B \\ &= \sum_{j=1}^{B} [\hat{F}_B^n(V_{(j)}) - \hat{F}_B^n(V_{(j-1)})] V_{(j)} = \hat{V}_n^B,\end{aligned}$$

since $\hat{F}_B^n(V_{(j)}) = (j/B)^n$ if there are no ties in the sample. The formula continues to hold even if there are ties, in which case we simply collapse the ties, using the fact that $\sum_{j=i}^{k} \hat{F}_B^n(V_{(j)}) - \hat{F}_B^n(V_{(j-1)}) = \hat{F}_B^n(V_{(k)}) - \hat{F}_B^n(V_{(i-1)})$ when $V_{(i-1)} < V_{(i)} = V_{(i+1)} = \cdots = V_{(k)} < V_{(k+1)}$.

Now, we are ready to prove Eq. (3.8). All we need to do is to compare the cumulative sums of the coefficients in the two estimators:

$$\sum_{j=1}^{k} \frac{\binom{j-1}{n-1}}{\binom{B}{n}} = \frac{\binom{k}{n}}{\binom{B}{n}}, \quad \sum_{j=1}^{k} \frac{j^n - (j-1)^n}{B^n} = \frac{k^n}{B^n}.$$

We need only consider $k \ge n$ (the case $k < n$ is trivial). One can easily verify the following expression backwards:

$$\begin{aligned}\frac{\binom{k}{n}}{\binom{B}{n}} < \frac{k^n}{B^n} &\iff \frac{\binom{k}{n}}{k^n} < \frac{\binom{B}{n}}{B^n} \\ &\iff \prod_{i=0}^{n-1}(1 - \frac{i}{k}) < \prod_{i=0}^{n-1}(1 - \frac{i}{B}),\end{aligned}$$

where the last inequality follows from $k < B$ and $n > 1$. Thus, we have verified the following for all $1 \le k < B$:

$$\sum_{j=1}^{k} \frac{\binom{j-1}{n-1}}{\binom{B}{n}} < \sum_{j=1}^{k} \frac{j^n - (j-1)^n}{B^n}.$$

Eq. (3.8) now follows since $V_{(1)} < \cdots < V_{(B)}$ lies in the isotonic cone while we have proved the difference of the two coefficients lies in the dual cone of the isotonic cone. An elementary way to see this is to first compare the coefficients in front of $V_{(B)}$: clearly, $\hat{U}_n^B$'s is larger since it has smaller sum of all coefficients (but the one in front of $V_{(B)}$; take $k = B - 1$) whereas the total sum is always one. Repeat this comparison for $V_{(1)}, \dots, V_{(B-1)}$.

Lastly, if $V_{(1)} < V_{(n)}$, then there exists a subset (with repetition) $1 \le i_1 \le \ldots \le i_n \le n$ such that $\max\{V_{(i_1)}, \ldots, V_{(i_n)}\} < V_{(n)}$. For instance, setting $i_1 = \ldots = i_n = 1$ would suffice. Since $\hat{V}_n^B$ puts positive mass on every subset of $n$ elements (with repetitions allowed), the strict inequality follows. We note that if $F$ is continuous, or if $F$ is discrete but non-degenerate, then $V_{(1)} < V_{(n)}$ with nonzero probability, hence

$$\text{Bias}(\hat{\theta}_n) = \mathbb{E}(\hat{V}_n^B - \hat{U}_n^B) < 0.$$

The proof is now complete. □

For further caveats, see Appendix A. The practical implication is that researchers may falsely conclude, on average, that a method is worse than it is, since the MeanMax estimator is negatively biased. In the context of environmental consciousness (Schwartz et al., 2019), more computation than necessary is used to make a conclusion.

**ECDF error.** The finite sample error (Eq. 2.1) of approximating the CDF with the ECDF (Eq. 3.4) can become unacceptable as $n$ increases:

**Theorem 2.** *If the sample does not contain the population maximum,* $\text{KS}(\hat{F}_B^n, F^n) \to 1$ *exponentially quickly as* $n$ *and* $B$ *increase.*

*Proof.* See Appendix B. □

Notably, this result always holds for cts. distributions, since the population maximum is never in the sample. Practically, this theorem suggests the failure of bootstrapping (Efron, 1982) for statistical hypothesis testing and constructing confidence intervals (CIs) of the expected maximum, since the bootstrap requires a good approximation of the CDF (Canty et al., 2006). Thus, relying on the bootstrap method for constructing confidence intervals of the expected maximum, as in Lucic et al. (2018), may lead to poor coverage of the true parameter.

## 4 Experiments

### 4.1 Experimental Setup

To support the validity of our conclusions, we opt for cleanroom Monte Carlo simulations, which enable us to determine the true parameter and draw millions of samples. To maintain the realism of our study, we apply kernel density estimation to actual results, using the resulting probability density (or discretized mass) function as the ground truth distribution. Specifically, we examine the experimental results of the following neural networks:

**Document classification.** We first conduct hyperparameter search over neural networks for document classification, namely a multilayer perceptron (MLP) and a long short-term memory (LSTM; Hochreiter and Schmidhuber, 1997) model representing state of the art (for LSTMs) from Adhikari et al. (2019). For our dataset and evaluation metric, we choose Reuters (Apté et al., 1994) and the $\text{F}_1$ score, respectively. Next, we fit discretized kernel density estimators to the results—see the appendix for experimental details. We name the distributions after their models, MLP and LSTM.

**Sentiment analysis.** Similar to Dodge et al. (2019), on the task of sentiment analysis, we tune the hyperparameters of two LSTMs—one ingesting embeddings from language models (ELMo; Peters et al., 2018), the other shallow word vectors (GloVe; Pennington et al., 2014). We choose the binary Stanford Sentiment Treebank (Socher et al., 2013) dataset and apply the same kernel density estimation method. We denote the distributions by their embedding types, GloVe and ELMo.

### 4.2 Experimental Test Battery

**False conclusion probing.** To assess the impact of the estimator bias, we measure the probability of researchers falsely concluding that one method underperforms its true value for a given $n$. The *unbiased* estimator has an expectation of $0.5$, preferring neither underestimates nor overestimates.

Concretely, denote the true $n$-run expected maxima of the method as $\theta_n$ and the estimator as $\hat{\theta}_n$. We iterate $n = 1, \ldots, 50$ and report the proportion of samples (of size $B = 50$) where $\hat{\theta}_n < \theta_n$. We compute the true parameter using 1,000,000 iterations of Monte Carlo simulation and estimate the proportion with 5,000 samples for each $n$.

**CI coverage.** To evaluate the validity of bootstrapping the expected maximum, we measure the coverage probability of CIs constructed using the percentile bootstrap method (Efron, 1982). Specifically, we set $B = 50$ and iterate $n = 1, \ldots, 50$. For each $n$, across $M = 1000$ samples, we compare the empirical coverage probability (ECP) to the nominal coverage rate of 95%, with CIs constructed using $5,000$ bootstrapped resamples. The ECP $\hat{\alpha}_n$ is computed as

$$\hat{\alpha}_n := \frac{1}{M} \sum_{i=1}^{M} \mathbb{I}\left(\theta_n \in \text{CI}_i\right), \qquad (4.1)$$

where $\text{CI}_i$ is the CI of the $i^{\text{th}}$ sample.

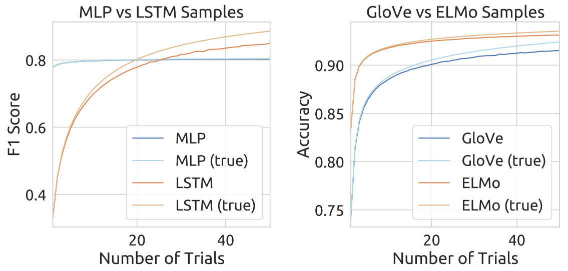

Figure 1: The estimated budget–quality curves, along with the true curves.

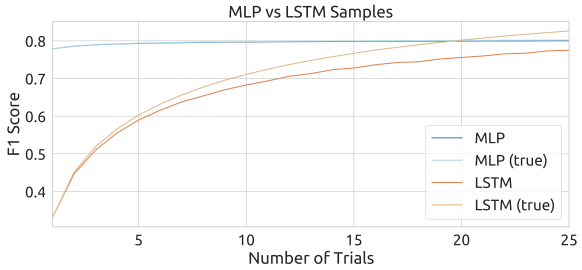

Figure 2: Illustration of a failure case with $B = 25$.

### 4.3 Results

Following Dodge et al. (2019), we present the budget–quality curves for each model pair in Figure 1. For each $n$ number of trials, we vertically average each curve across the 5,000 samples. We construct CIs but do not display them, since the estimate is precise (standard error $< 0.001$). For document classification, we observe that the LSTM is more difficult to tune but achieves higher quality after some effort. For sentiment analysis, using ELMo consistently attains better accuracy with the same number of trials—we do not consider the wall clock time.

In Figure 2, we show a failure case of biased estimation in the document classification task. At $B = 25$, from $n = 20$ to 25, the averaged estimate yields the wrong conclusion that the MLP outperforms the LSTM—see the true LSTM line, which is above the true MLP line, compared to its estimate, which is below.

**False conclusions probing.** Figure 3 shows the results of our false conclusion probing experiment. We find that the estimator quickly prefers negative errors as $n$ increases. The curves are mostly similar for both tasks, except the MLP fares worse. This requires further analysis, though we conjecture that the reason is lower estimator variance, which would result in more consistent errors.

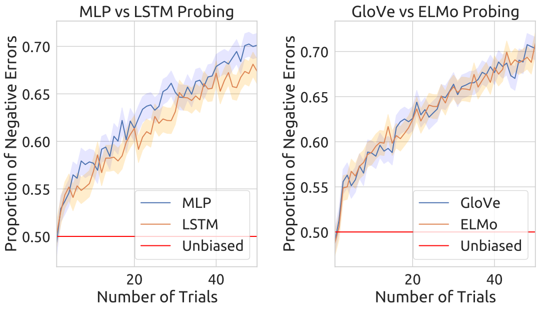

Figure 3: The false conclusion probing experiment results, along with Clopper–Pearson 95% CIs.

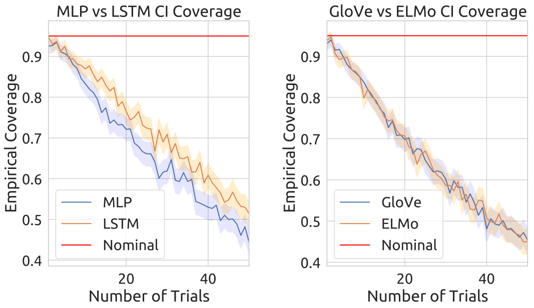

Figure 4: The CI coverage experiment results, along with Clopper–Pearson 95% CIs.

**CI coverage.** We present the results of the CI coverage experiment results in Figure 4. We find that the bootstrapped confidence intervals quickly fail to contain the true parameter at the nominal coverage rate of 0.95, decreasing to an ECP of 0.7 by $n = 20$. Since the underlying ECDF is the same, this result extends to Lucic et al. (2018), who construct CIs for the expected maximum.

## 5 Conclusions

In this work, we provide a dual-pronged theoretical and empirical analysis of Dodge et al. (2019). We find unspoken caveats in their work—namely, that the estimator is statistically biased under weak conditions and uses an ECDF assumption that is subject to large errors. We empirically study its practical effects on tasks in document classification and sentiment analysis. We demonstrate that it prefers negative errors and that bootstrapping leads to poorly controlled confidence intervals.

## Acknowledgments

This research was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

## References

Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin. 2019. Rethinking complex neural network architectures for document classification. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4046–4051.

Chidanand Apté, Fred Damerau, and Sholom M. Weiss. 1994. Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems*, 12(3):233–251.

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

Angelo J. Canty, Anthony C. Davison, David V. Hinkley, and Valérie Ventura. 2006. Bootstrap diagnostics and remedies. *Canadian Journal of Statistics*, 34(1):5–27.

Matt Crane. 2018. Questionable answers in question answering research: Reproducibility and variability of published results. *Transactions of the Association for Computational Linguistics*, 6:241–252.

Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. 2019. Show your work: Improved reporting of experimental results. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 2185–2194.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 1383–1392.

Bradley Efron. 1982. The Jackknife, the Bootstrap and other resampling plans. In *CBMS-NSF Regional Conference Series in Applied Mathematics, Philadelphia: Society for Industrial and Applied Mathematics*.

Jessica Forde and Michela Paganini. 2019. The scientific method in the science of machine learning. *arXiv:1904.10922*.

Kyle Gorman and Steven Bedrick. 2019. We need to talk about standard splits. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2786–2791.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. 2018. Are GANs created equal? A large-scale study. In *Advances in Neural Information Processing Systems*, pages 700–709.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2227–2237.

Nils Reimers and Iryna Gurevych. 2017. Reporting score distributions makes a difference: Performance study of LSTM-networks for sequence tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 338–348.

Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. 2019. Green AI. *arXiv:1907.10597*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642.

| Model | Mode | Batch Size | Learning Rate | Seed | Dropout | # Layers | Hidden Dim. | WDrop | EDrop | $\beta_{\text{EMA}}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| MLP | – | $(16, 32, 64)$ | $0.001$ | $[0, 10^7]_D$ | $[0.05, 0.7]$ | $1$ | $[256, 768]_D$ | – | – | – |
| LSTM | (nonstatic$[0.5]$,<br>static$[0.4]$, rand$[0.1]$) | $(16, 32, 64)$ | TExp$[0.001, 0.099]$ | $[0, 10^7]_D$ | $[0.05, 0.7]$ | $(1[0.75], 2[0.25])$ | $[384, 768]_D$ | $[0, 0.3]$ | $[0, 0.3]$ | $[0.985, 0.995]$ |

Table 1: Hyperparameter random search bounds. $[\cdot, \cdot]_D$ indicates a discrete uniform range, while $[\cdot, \cdot]$ continuous uniform. TEXP$[\cdot, \cdot]$ denotes the truncated exponential distribution. Tuples represent categorical distributions, uniform by default. WDrop and EDrop denote weight and embed dropout. For the GloVe- and ELMo-based search bounds, see https://github.com/allenai/show-your-work.

## A Cautionary Notes

We caution that the estimator described in *the text* of Dodge et al. is $\hat{V}_n^n$. This is clear from their equation (7) where the empirical distribution is defined over the first $n$ samples, instead of the $B$ samples that we use here. In other words, they claim, at least in the text, to use $\hat{F}_n$ instead of $\hat{F}_B$ for their estimator $\hat{V}_n^n$. Clearly, the estimator $\hat{V}_n^n$ is (much) worse than $\hat{V}_n^B$ since the latter exploits all $B$ samples while the former only looks at the first $n$ samples. However, close examination of their codebase[1] reveals that they use $\hat{V}_n^B$, so the paper discrepancy is a simple notation error.

Lastly, we mention that our notation for $\hat{U}_n^B$ and $\hat{V}_n^B$ is motivated by the fact that the former is a $U$-statistic while the latter is a $V$-statistic. The relation between the two has been heavily studied in statistics since Hoeffding's seminar work. For us, it suffices to point out that $\hat{V}_n^B \leq \hat{U}_n^B$, with the latter being unbiased while the former is only asymptotically unbiased. The difference between the two is more pronounced when $n$ is close to $B$. We note that $\hat{U}_n^B$ can be computed by a reasonable approximation of the binomial coefficients, using say Stirling's formula.

## B Proof of Theorem 2

**Theorem 3.** *If the sample does not contain the population maximum,* $\text{KS}(\hat{F}_B^n, F^n) \to 1$ *exponentially quickly as* $n$ *and* $B$ *increase.*

*Proof.* Suppose $v^*$ is not in the sample $v_1, \ldots, v_B$, where $v_1 \leq \cdots \leq v_B < v^*$. Then

$$\sup_{x \in \mathbb{R}} |\hat{F}_B^n(x) - F^n(x)| \geq |\hat{F}_B^n(v_B) - F^n(v_B)|.$$

From Equation 2.1, $\hat{F}_B^n(v_B) = (\hat{F}_B(v_B))^n = 1 > (F(v_B))^n = F^n(v_B)$, hence

$$|\hat{F}_B^n(v_B) - F^n(v_B)| = 1 - (F(v_B))^n.$$

Thus concluding the proof. □

| Model | # Runs | Bandwidth | Support | Bins |
|---|---|---|---|---|
| MLP | 145 | 0.0049 | $[0.72, 0.82]$ | 511 |
| LSTM | 152 | 0.059 | $[-0.18, 1.08]$ | 511 |
| GloVe | 114 | 0.018 | $[0.46, 0.97]$ | 511 |
| ELMo | 84 | 0.041 | $[0.39, 0.99]$ | 511 |

Table 2: Model kernel parameters. Bandwidth chosen using Scott's normal reference rule. Bins denote the number of discretized slots.

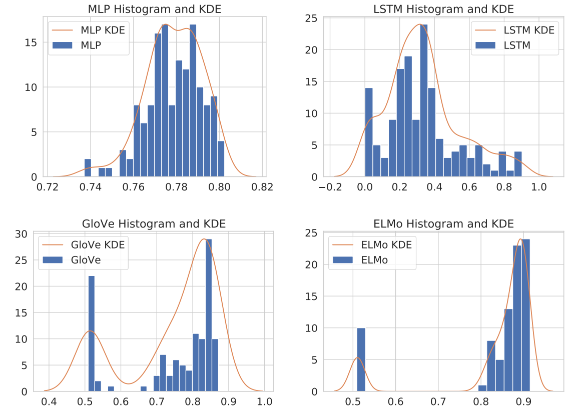

Figure 5: Gaussian kernel density estimators fitted to each model's results, along with the histograms of the original runs.

## C Experimental Settings

We present hyperparameters in Tables 1 and 2 and Figure 5. We conduct all GloVe and ELMo experiments using PyTorch 1.3.0 with CUDA 10.0 and cuDNN 7.6.3, running on NVIDIA Titan RTX, Titan V, and RTX 2080 Ti graphics accelerators. Our MLP and LSTM experiments use PyTorch 0.4.1 with CUDA 9.2 and cuDNN 7.1.4, running on RTX 2080 Ti's. We use Hedwig[2] for the document classification experiments and the Show Your Work codebase (see link in Table 1) for the sentiment classification ones.

[1] https://github.com/allenai/allentune

[2] https://github.com/castorini/hedwig