Expected Validation Performance and Estimation of a Random Variable's Maximum

Jesse Dodge[♣] **Suchin Gururangan**[♡] **Dallas Card**[♠]
Roy Schwartz[◇] **Noah A. Smith**[♡♣]

[♠]Stanford University
[◇]Hebrew University of Jerusalem
[♡]Paul G. Allen School of Computer Science & Engineering, University of Washington
[♣]Allen Institute for Artificial Intelligence
jessed@allenai.org

Abstract

Research in NLP is often supported by experimental results, and improved reporting of such results can lead to better understanding and more reproducible science. In this paper we analyze three statistical estimators for expected validation performance, a tool used for reporting performance (e.g., accuracy) as a function of computational budget (e.g., number of hyperparameter tuning experiments). Where previous work analyzing such estimators focused on the bias, we also examine the variance and mean squared error (MSE). In both synthetic and realistic scenarios, we evaluate three estimators and find the unbiased estimator has the highest variance, and the estimator with the smallest variance has the largest bias; the estimator with the smallest MSE strikes a balance between bias and variance, displaying a classic bias-variance tradeoff. We use expected validation performance to compare between different models, and analyze how frequently each estimator leads to drawing incorrect conclusions about which of two models performs best. We find that the two biased estimators lead to the fewest incorrect conclusions, which hints at the importance of minimizing variance and MSE.

1 Introduction

Drawing robust conclusions when comparing different methods in natural language processing is central to scientific progress. If two research groups set up the same set of experiments, they should expect to get similar results. One area that has high impact, but is often underreported, is hyperparameter tuning (Reimers and Gurevych, 2017; D'Amour et al., 2020; Dodge et al., 2019; Melis et al., 2018). Hyperparameter search is key to getting strong results; for example, RoBERTa (Liu et al., 2019) found stronger results than BERT (Devlin et al., 2019) partly due to an increased budget for hyperparameter tuning. Often researchers only report the performance of the single best-found model during a hyperparameter search (Ethayarajh and Jurafsky, 2020; Forde and Paganini, 2019; Sculley et al., 2018). What if a future researcher has a smaller computational budget for training models? What performance should they expect to find? One way of reporting such results is expected validation performance (EVP).

What is EVP? Assume a budget to train B models (e.g., B rounds of hyperparameter search), with resulting evaluation scores (e.g., accuracy) on the validation set $X_1 \ldots X_B$. Standard practice would report the maximum result, X_{max}, but this effectively hides the experiments which were required to achieve that maximum performance. Using all B results, EVP estimates what the maximum would have been if we had had a *smaller* budget n (where $1 \leq n < B$). This is estimating what the maximum of n trials *would be*, in expectation; this is thus a statistical estimation problem. The formulation was introduced by Dodge et al. (2019), who proposed a first estimator (defined as V_n^B in Equation 2). This estimator was later shown to be biased by Tang et al. (2020), who introduced an unbiased estimator (defined as U_n^B in Equation 3) for the same expected maximum.

In Section 2 we use tools from combinatorics to derive both previously-introduced estimators, relate them to each other, and show that they make two opposing assumptions; we show that changing only one of these assumptions instead of both leads to a third estimator, W_n^B, and prove that this estimator is even more biased than V_n^B.

Unbiased estimators are generally preferred, all else equal, but only analyzing the bias provides an incomplete picture of the quality of an estimator. In Section 3 we also measure the variance and mean squared error of these three estimators in synthetic experiments. We find that while U_n^B is unbiased (as expected) it has the highest variance, and that W_n^B is the most biased but has the lowest variance; V_n^B strikes a balance between the bias and variance,

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 4066–4073
November 7–11, 2021. ©2021 Association for Computational Linguistics

leading to the lowest mean squared error (so, the average squared distance to the true value being estimated is smallest).[1]

Finally, in Section 4 we explore how these estimators impact a common use case of EVP: comparing the results of hyperparameter searches for two models. Specifically, we examine how frequently the estimators lead to incorrectly concluding that the worse model outperforms the better one (for a given budget), and find that the high-variance (but unbiased) U_n^B more frequently leads to such incorrect conclusions than the other lower variance (but biased) estimators.

2 Estimation of the Expected Maximum

Notation We begin by defining some notation. Consider n i.i.d. random variables, $X_1, \ldots, X_n \sim F$, for some unknown F.[2]

- $Y_n = \max\{X_1, \ldots, X_n\}$, a random variable representing the maximum of n i.i.d. random variables.
- $\theta_n = \mathbb{E}\left[Y_n\right]$, the true expected value of Y_n .
- $\hat{\theta}_n$, an estimator of θ_n (the expected value).
- Bias $(\hat{\theta}) = \mathbb{E}[\hat{\theta}] - \theta$, the bias of $\hat{\theta}$.
- Var $(\hat{\theta})$, the estimator's variance due to sampling.
- MSE $(\hat{\theta})$ = Bias $(\hat{\theta})^2$ + Var $(\hat{\theta})$, the mean squared error of the estimator. MSE is the average squared difference between the estimator and true value, or the expected value of the squared error loss between the estimator and the true statistic.

Estimation of the Expected Maximum We consider the estimation of θ, the expected maximum. With a finite sample of B draws from F, we can estimate this quantity for $1 \le n \le B$. We begin with the definition of an expectation over a discrete set: $\mathbb{E}\left[Y_n\right] = \sum_{i=1}^B X_i P(Y_n = X_i)$. This can be rewritten using order statistics. Let $X_{(i)}$ denote the ith largest sample (distinct from X_i). Then,

$$\begin{aligned} \mathbb{E}\left[Y_n\right] &= \textstyle\sum_{i=1}^B X_{(i)} P(Y_n = X_{(i)}) \qquad (1) \\ &= \textstyle\sum_{i=1}^B X_{(i)} \left(P(Y_n \le X_{(i)}) - P(Y_n < X_{(i)})\right) \\ &= \textstyle\sum_{i=1}^B X_{(i)} \left(P(Y_n \le X_{(i)}) - P(Y_n \le X_{(i-1)})\right) \end{aligned}$$

This estimation depends on $P(Y_n \le X_{(k)})$, the probability that a sample of size n has a maximum that is less than or equal to the kth order statistic. We can estimate this probability by counting: from our B points how many sets of size n are there which only include order statistics up to k, out of the total number of sets of size n? We turn to combinatorics, which provides tools for counting such sets. Two key assumptions must be made: whether the sets will contain repetition or not and whether the items in the sets will be ordered or unordered. These assumptions will lead to different estimators.

Ordered subsets that allow repetition are known as **strings**, and there are B^n strings of size n from B points. With these assumptions, we now have a closed form for $P(Y_n \le X_{(k)})$, and plugging this into Equation 1 we define our first estimator:

$$V_n^B = \sum_{i=1}^B X_{(i)} \left(\frac{i^n}{B^n} - \frac{(i-1)^n}{B^n} \right). \qquad (2)$$

This is exactly the estimator introduced in Dodge et al. (2019), derived using the plug-in estimator for the CDF (the empirical CDF).

Making the opposite two assumptions, unordered subsets without repetition are **combinations**, for which there are $\binom{B}{n}$ subsets of size n from B points. The corresponding estimator is

$$U_n^B = \sum_{i=1}^B X_{(i)} \left(\frac{\binom{i}{n}}{\binom{B}{n}} - \frac{\binom{i-1}{n}}{\binom{B}{n}} \right). \qquad (3)$$

This is the estimator of Tang et al. (2020), which they derived as an unbiased estimator.

What about changing only one of these assumptions? Ordered subsets without repetition are **permutations**, for which there are ${}_BP_n$ subsets of size n from B points. Though these assumptions are different, the corresponding estimator is equivalent to U_n^B, since:

$$\frac{{}_kP_n}{{}_BP_n} = \frac{\frac{k!}{(k-n)!}}{\frac{B!}{(B-n)!}} = \frac{\frac{k!}{n!\,(k-n)!}}{\frac{B!}{n!\,(B-n)!}} = \frac{\binom{k}{n}}{\binom{B}{n}} \qquad (4)$$

Finally, unordered subsets with repetition are **multisets**, the number of which is denoted $\left(\!\binom{B}{n}\!\right) = \binom{B+n-1}{n}$. We introduce the corresponding estimator:

$$W_n^B = \sum_{i=1}^B X_{(i)} \left(\frac{\left(\!\binom{i}{n}\!\right)}{\left(\!\binom{B}{n}\!\right)} - \frac{\left(\!\binom{i-1}{n}\!\right)}{\left(\!\binom{B}{n}\!\right)} \right). \qquad (5)$$

[1] There is a long tradition of preferring biased estimators over unbiased ones (Wasserman, 2004), such as when estimating the population variance using the sample variance, or the James–Stein estimator (James and Stein, 1961).

[2] For clarity, we dispense with notation mapping into the use case of interest, as well as the computational details; see Dodge et al. (2019) for a full discussion.

Comparing estimators To compare these estimators we turn to the standard statistical tools of bias, variance, and mean squared error. U_n^B was shown to be unbiased, and Bias $(V_n^B) \leq 0$ (Tang et al., 2020). We show that Bias $(W_n^B) \leq$ Bias (V_n^B), that is W_n^B has a larger negative bias than V_n^B.

Theorem 1 *Assume* $X_1, \ldots, X_B \sim F$ *are i.i.d. from unknown distribution* F. *Let* $1 \leq k < B$, *and* $1 \leq n \leq B$. *Then, Bias* $(W_n^B) \leq$ *Bias* (V_n^B).

Consider V_n^B as defined in Equation 2. The sum of the coefficients of the $X_{(i)}$ up to k is $\frac{k^n}{B^n}$. It is sufficient to show that, for a given k, this term is less than the sum of the coefficients for W_n^B, which is $\left(\!\binom{k}{n}\!\right) \Big/ \left(\!\binom{B}{n}\!\right)$; this implies that V_n^B places less probability mass on the smaller order statistics than W_n^B.

$$\frac{k^n}{B^n} < \frac{\left(\!\binom{k}{n}\!\right)}{\left(\!\binom{B}{n}\!\right)} \iff \frac{k^n}{B^n} < \frac{\binom{k+n-1}{n}}{\binom{B+n-1}{n}} \tag{6}$$

$$\iff \frac{\binom{B+n-1}{n}}{B^n} < \frac{\binom{k+n-1}{n}}{k^n}. \tag{7}$$

The left side of Eq. 7 can be rewritten as:

$$\frac{\binom{B+n-1}{n}}{B^n} = \frac{\frac{(B+n-1)!}{n!\,B!}}{B^n} = \frac{1}{n!}\frac{\prod_{j=0}^{n-1}(B+n-1-j)}{B^n}$$
$$= \left(\tfrac{1}{n!}\right)\prod_{j=0}^{n-1}\left(1+\tfrac{(n-1)-j}{B}\right). \tag{8}$$

Rewriting the right side of Eq. 7 in a similar manner, we have

$$\prod_{j=0}^{n-1}\left(1+\tfrac{(n-1)-j}{B}\right) < \prod_{j=0}^{n-1}\left(1+\tfrac{(n-1)-j}{k}\right)$$

since $B > k$. This completes our proof.

3 Simulation Experiment

In the previous section we proved that W_n^B is at least as biased as V_n^B, but such a bound tells us little about how these estimators behave in practice. In this section we provide a simulation experiment which allows us to measure the bias and variance of each estimator directly. We assume a distribution for X_i, which allows us to draw many samples of size B so we can evaluate how these estimators behave. Recall that the motivating application of our estimators is when $\{X_i\}_{i=1}^n$ represent the evaluations from different trials of hyperparameter optimization, so designing a reasonable distribution for X_i allows us to evaluate the estimators with tens of thousands of simulated trials without having to train that many models.

3.1 Synthetic Experiments Setup

To begin, we sample 100,000 random values from a Normal$(0.6, 0.07)$ distribution (truncated to $[0, 1]$). We then sample 10,000 values from this set, resulting in 9536 unique values, with a true maximum of 0.854. Call this bag of values $\mathcal{V}$. We then set $B = 30$, and estimate the true EVP as a function of n for $n = 1, ..., 30$, by drawing 50,000 samples of size n from $\mathcal{V}$, for each value of n, and reporting the average maximum for each n ("True EVP" in Figure 1, top). To estimate the mean and variance of a given estimator we sample 10,000 B values from $\mathcal{V}$ and compute the value of the estimator for each, then calculate the mean and variance across those 10,000 samples.

3.2 Bias, Variance, MSE

Figure 1 shows the estimated mean (top), variance (middle), and MSE (bottom) of each estimator. As can be seen in the top figure, Bias $(W_n^B) \leq$ Bias $(V_n^B) \leq$ Bias $(U_n^B) = 0$ with a a difference that grows with n, confirming the proved bounds for these estimators. In the middle figure we measure the variance of these estimators, and we see that Var $(W_n^B) \leq$ Var $(V_n^B) \leq$ Var (U_n^B), with the difference in variance again growing with n.

In the bottom of Figure 1 we plot the mean squared error (MSE); as a reminder, MSE $(\hat{\theta}) =$ Bias $(\hat{\theta})^2 +$ Var $(\hat{\theta})$, so lower is better. Although U_n^B is unbiased, and W_n^B has the lowest variance, V_n^B strikes the balance between bias and variance that leads to the lowest MSE.

Thus we see that a higher variance estimator may, on average, be farther from the true value than a biased but lower variance estimator. Again tying this back to our motivating application of hyperparameter tuning, in this scenario V_n^B is more likely to underestimate than overestimate performance for a given budget, but overall will have lower variance between researchers running sets of experiments, and will on average have closer predictions to the true value than the other two estimators.

4 Incorrect Conclusions

While analyzing how close each estimator is to the true expected maximum for one model is important, in practice these curves are often used to compare two or more different models. For example, NLP practitioners may run hyperparameter searches for two different models, compute the expected validation curves for each, and select the model which

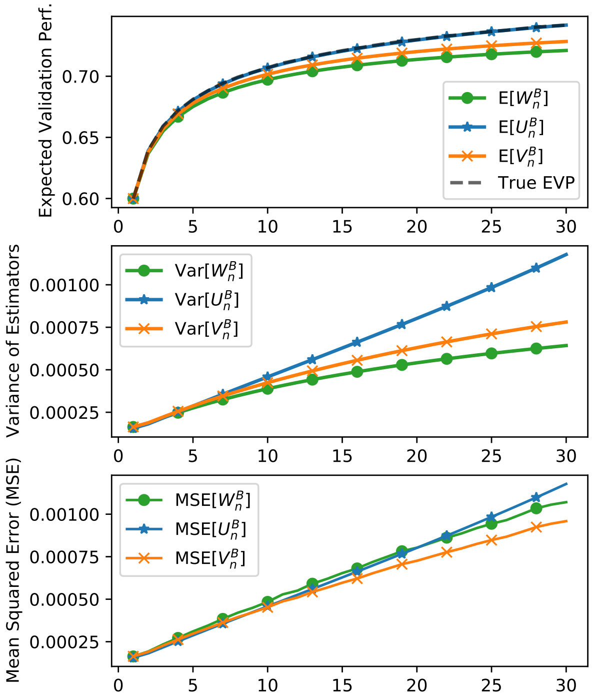

Figure 1: Expected value (top) variance (middle) and mean squared error (MSE; bottom) of the three estimators, based on a 10,000 simulations for a large random bag of possible validation scores. As expected, U_n^B is unbiased while V_n^B and W_n^B have negative bias. However, W_n^B has the lowest variance, and V_n^B balances bias and variance, leading to lowest MSE.

presents a higher estimated maximum performance (Zhang et al., 2021; Gehman et al., 2020). In this section we examine the three estimators in such a scenario, asking how frequently each estimator leads to drawing *incorrect* conclusions about which model performs best for a specific budget.

4.1 Experimental Setup

We proceed by performing a sensitivity analysis: we run 100 trials of random hyperparameter search (far more than is typically necessary to establish that one model outperforms another in current practice) for a CNN (Kim, 2014) and a linear bag-of-embedding (LBoE) (Yogatama and Smith, 2015). These models are trained on the Stanford sentiment treebank 5-way text classification task (Socher et al., 2013). We include details about the dataset (and a link to download it) in Appendix B.

For all three estimators, the CNN has higher expected performance than the LBoE, for all $n \leq B$.[3] We then simulate a more practical scenario where a practitioner runs $B \in \{15, \ldots, 30\}$ rounds of hyperparameter search for the two models and compares their estimated maximum at $n = B$ (so, the estimated maximum of B points) to conclude which is best (that is, which estimator has lower error).

We are interested in the rate at which each estimator would draw an incorrect conclusion about which model performs best. To evaluate this question we do the following: for each value of B we sample 50,000 times from the 100 real experiments and compute the fraction for which the value of each estimator for the CNN is less than for LBoE. For example, to estimate the proportion with which U_B^B draws an incorrect conclusion with $B = 15$ we draw 50,000 samples of size 15 from the 100 real experimental results for each of the CNN and LBoE, then compute the fraction of those samples for which U_B^B for the CNN is less than U_B^B for LBoE. A stable estimator will make the same prediction with small and large B.

4.2 Results

In Figure 2 we see the results of this experiment: U_B^B more frequently would lead a practitioner to incorrectly conclude that the LBoE outperforms the CNN for budgets $B \in \{15, \ldots, 30\}$ than V_B^B or W_B^B. This scenario models what we expect a practitioner would care about: the frequency with which one draws conclusions that would be consistent with conclusions drawn with a larger budget. Here the high variance of U_B^B likely plays a role the stability of its predictions; while it may be unbiased, the lower variance estimators are more reliable.

5 Conclusion

Drawing reproducible conclusions from our experimental results is of paramount importance to NLP researchers, practitioners, and users of language technologies. Expected validation performance curves are tools for comparing the results of hyperparameter searches; we showed how two previously-introduced estimators are connected through combinatorial assumptions, and introduced a third estimator by varying such assumptions. In synthetic experiments, we analyzed the bias, variance, and mean squared error, and found a classic example of a bias-variance tradeoff; the unbiased estimator U_n^B had the largest variance, and the most biased estimator W_n^B had the lowest vari-

[3] See Appendix C for details. Figure 3 shows expected validation curves for $B = 100$ for all three estimators; with B this large, the three estimators are very similar.

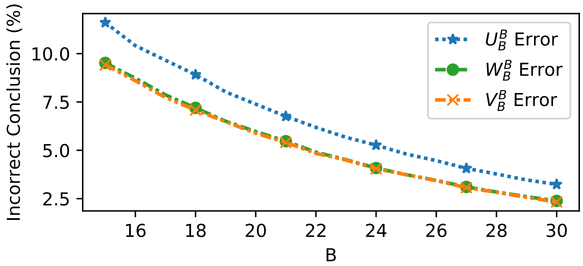

Figure 2: For a budget of B trials, what fraction of the time does each estimator *incorrectly* predict that the expected maximum of those B trials (so, $n = B$) is higher for the LBoE than for the CNN? Lower is better. The proportion of errors made by the unbiased estimator U_n^B when $n = B$ is higher than for either of the biased estimators, V_n^B and W_n^B. Confidence intervals around this proportion are not shown, as they are small.

ance, while V_n^B struck a balance leading to the lowest mean squared error. Finally, in realistic experiments we found that the unbiased estimator led to incorrectly identifying the better of two models at a higher rate than the lower variance estimator. Overall, V_n^B had the lowest MSE and the lowest rate of drawing incorrect conclusions, so V_n^B is our recommendation for estimating the expected maximum.

Acknowledgements

Dallas Card was supported in part by the Stanford Data Science Institute.

References

Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D. Hoffman, Farhad Hormozdiari, Neil Houlsby, Shaobo Hou, Ghassen Jerfel, Alan Karthikesalingam, Mario Lucic, Yian Ma, Cory McLean, Diana Mincu, Akinori Mitani, Andrea Montanari, Zachary Nado, Vivek Natarajan, Christopher Nielson, Thomas F. Osborne, Rajiv Raman, Kim Ramasamy, Rory Sayres, Jessica Schrouff, Martin Seneviratne, Shannon Sequeira, Harini Suresh, Victor Veitch, Max Vladymyrov, Xuezhi Wang, Kellie Webster, Steve Yadlowsky, Taedong Yun, Xiaohua Zhai, and D. Sculley. 2020. Underspecification presents challenges for credibility in modern machine learning.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL*.

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 (EMNLP-IJCNLP)*, pages 2185–2194, Hong Kong, China.

Kawin Ethayarajh and Dan Jurafsky. 2020. Utility is in the eye of the user: A critique of nlp leaderboards. In *Proc. of EMNLP*.

Jessica Zosa Forde and Michela Paganini. 2019. The scientific method in the science of machine learning.

Sam Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. 2020. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. In *Findings of EMNLP*.

W. James and C. Stein. 1961. Estimation with quadratic loss. *Proc. Fourth Berkeley Symp. Math. Statist. Prob.*, page 361–379.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proc. of EMNLP*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Gábor Melis, Chris Dyer, and Phil Blunsom. 2018. On the state of the art of evaluation in neural language models. In *Proc. of EMNLP*.

Nils Reimers and Iryna Gurevych. 2017. Reporting score distributions makes a difference: Performance study of LSTM-networks for sequence tagging. In *Proc. of EMNLP*.

D. Sculley, Jasper Snoek, Ali Rahimi, and Alex Wiltschko. 2018. Winner's curse? On pace, progress, and empirical rigor. In *Proc. of ICLR (Workshop Track)*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proc. of EMNLP*.

Raphael Tang, Jaejun Lee, Ji Xin, Xinyu Liu, Yaoliang Yu, and Jimmy Lin. 2020. Showing your work doesn't always work. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2766–2772, Online.

Larry A. Wasserman. 2004. *All of Statistics: A Concise Course in Statistical Inference*. Springer Texts in Statistics.

Dani Yogatama and Noah A. Smith. 2015. Bayesian optimization of text representations. In *Proc. of EMNLP*.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q. Weinberger, and Yoav Artzi. 2021. Revisiting few-sample bert fine-tuning. In *Proc. of ICLR*.

Label	Train	Valid	Test
0	1092	139	279
1	2218	289	633
2	1624	229	389
3	2322	279	510
4	1288	165	399

Table 1: Label distributions for SST-5. 0 is "very negative", 2 is "neutral", and 4 is "very positive".

A Expected Validation Curves for two models, all three estimators

We include expected validation curves of the same data using all three estimators in Figure 3. They look roughly the same.

B Training Data

The CNN and LBoE in Section 4 are trained on the Stanford sentiment treebank 5-way text classification task (Socher et al., 2013). There are 8544 train examples, 2210 test examples, and 1101 validation examples. It can be downloaded here: `http://nlp.stanford.edu/sentiment`. We present label distributions in Table 1.

C Hyperparameter Ranges

The hyperparameter bounds for the CNN and LBoE in Section 4, which were trained on SST-5 as described in Appendix B.

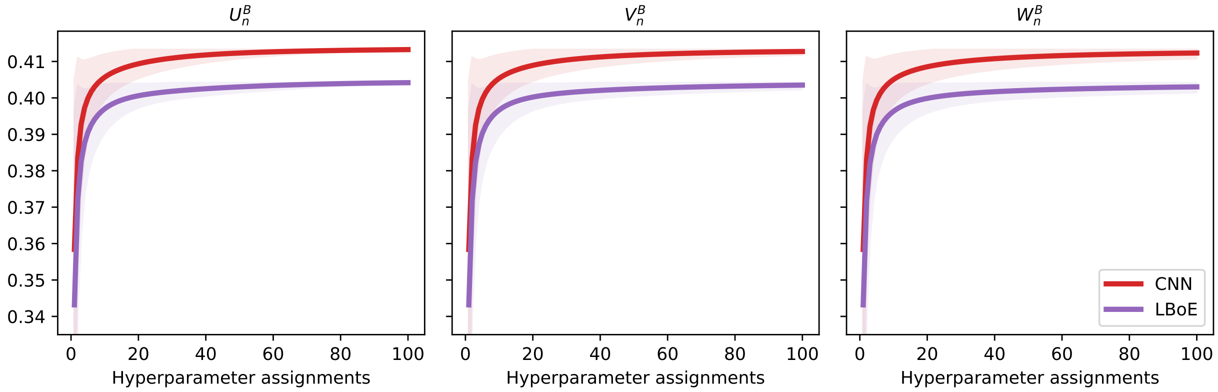

Figure 3: U_n^B (left), V_n^B (middle) and W_n^B (right) curves of the same data, a CNN and a Linear Bag of Embeddings (LBoE), evaluated on SST-5, with B=100. With such a large B the three estimators are very similar. For all three estimators, the CNN has higher expected performance than the LBoE for all n.

Computing infrastructure	GeForce GTX 1080 GPU
Number of search trials	100
Search strategy	uniform sampling
Best validation accuracy	41.3
Training duration	77 sec

HP	number of epochs	patience	batch size	embedding	encoder	max filter size
Search space	50	10	64	GloVe (50 dim)	Convnet	*uniform-integer*[1, 9]
Best assignment	50	10	64	GloVe (50 dim)	Convnet	9

HP	number of filters	dropout	LR scheduler	patience	reduction factor
Search space	*uniform-integer*[64, 512]	*uniform-float*[0, 0.5]	reduce on plateau	2 epochs	0.5
Best assignment	390	0.2	reduce on plateau	2 epochs	0.5

HP	optimizer	LR
Search space	Adam	*loguniform-float*[1e-6, 1e-1]
Best assignment	Adam	0.0004

Table 2: SST (fine-grained) CNN classifier search space and best assignments.

Computing infrastructure	GeForce GTX 1080 GPU
Number of search trials	100
Search strategy	uniform sampling
Best validation accuracy	42.7
Training duration	41 sec

Hyperparameter	number of epochs	patience	batch size	embedding	dropout
Search space	50	10	64	GloVe (50 dim)	*uniform-float*[0, 0.5]
Best assignment	50	10	64	GloVe (50 dim)	0.4

Hyperparameter	LR scheduler	patience	reduction factor	optimizer	LR
Search space	reduce on plateau	2 epochs	0.5	Adam	*loguniform-float*[1e-6, 1e-1]
Best assignment	reduce on plateau	2 epochs	0.5	Adam	0.044

Table 3: SST (fine-grained) BOE classifier search space and best assignments.