Interpreting the Robustness of Neural NLP Models to Textual Perturbations

Anonymous ACL submission

Abstract

Modern Natural Language Processing (NLP) models are known to be sensitive to input perturbations and their performance can decrease when applied to real-world, noisy data. However, it is still unclear why models are less robust to some perturbations than others. In this work, we test the hypothesis that the extent to which a model is affected by an unseen textual perturbation (robustness) can be explained by the learnability of the perturbation (defined as how well the model learns to identify the perturbation with a small amount of evidence). We further give a causal justification for the learnability metric. We conduct extensive experiments with four prominent NLP models — TextRNN, BERT, RoBERTa and XLNet — over eight types of textual perturbations on three datasets. We show that a model which is better at identifying a perturbation (higher learnability) becomes worse at ignoring such a perturbation at test time (lower robustness), providing empirical support for our hypothesis.

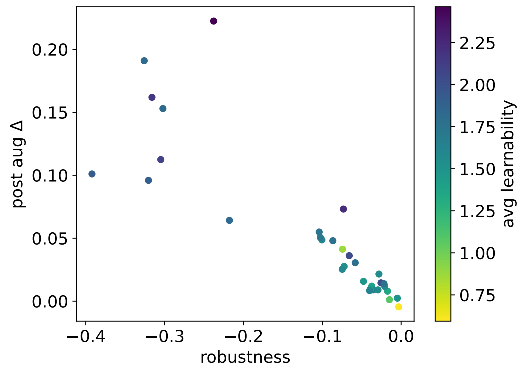


Figure 1: Robustness vs. post data augmentation Δ vs. average learnability on IMDB dataset. Each point in the plots represents a model-perturbation pair. We define "robustness" as the performance drop on perturbed test set, "post aug Δ" as the performance boost on perturbed test set after data augmentation along such a perturbation, and "average learnability" as how well the model learns to identify the perturbation with a small amount of evidence.

1 Introduction

Despite the success of deep neural models on many Natural Language Processing (NLP) tasks (Liu et al., 2016; Devlin et al., 2019; Liu et al., 2019b), recent work has discovered that these models are not robust to noisy input from the real world and thus their performance will decrease (Prabhakaran et al., 2019; Niu et al., 2020; Ribeiro et al., 2020; Moradi and Samwald, 2021). A reliable NLP system should not be easily fooled by slight noise in the text. Although a wide range of evaluation approaches for robust NLP models have been proposed (Ribeiro et al., 2020; Morris et al., 2020; Goel et al., 2021; Wang et al., 2021), few attempts have been made to *understand* these benchmark results. Given the difference of robustness between models and perturbations, it is a natural question why models are more sensitive to some perturbations than others. It is crucial to avoid over-sensitivity to input perturbations, and understanding why it happens is useful for revealing the weaknesses of current models and designing more robust training methods. To the best of our knowledge, a quantitative measure to *interpret* the robustness of NLP models to textual perturbations has yet to be proposed. To improve the robustness under perturbation, it is common practice to leverage data augmentation (Li and Specia, 2019; Min et al., 2020; Tan and Joty, 2021). Similarly, how much data augmentation through the perturbation improves model robustness varies between models and perturbations. In this work, we aim to investigate two Research Questions (RQ):

- **RQ1:** *Why NLP models are less robust to some perturbations than others?*
- **RQ2:** *Why data augmentation works better at improving the model robustness to some perturbations than others?*

We test a hypothesis for RQ1 that the extent to which a model is affected by an unseen textual perturbation (robustness) can be explained by the learnability of the perturbation (defined as how well the model learns to identify the perturbation with a small amount of evidence). We also validate another hypothesis for RQ2 that the learnability metric is predictive of the improvement on robust performance brought by data augmentation along a perturbation. Our proposed learnability is inspired by the concepts of Randomized Controlled Trial (RCT) and Average Treatment Effect (ATE) from Causal Inference (Rubin, 1974; Holland, 1986). Estimation of perturbation learnability for a model consists of three steps: ① randomly labelling a dataset, ② perturbing examples of a particular pseudo class with probabilities, and ③ using ATE to measure the ease with which the model learns the perturbation. The core intuition for our method is to frame an RCT as a perturbation identification task and formalize the notion of learnability as a causal estimand based on ATE. We conduct extensive experiments on four neural NLP models with eight different perturbations across three datasets and find strong evidence for our two hypotheses. Combining these two findings, we further show that data augmentation is *only* more effective at improving robustness against perturbations that a model is more sensitive to, contributing to the interpretation of robustness and data augmentation. Learnability provides a clean setup for analysis of the model behaviour under perturbation, which contributes better model interpretation as well.

Contribution. This work provides an empirical explanation for why NLP models are less robust to some perturbations than others. The key to this question is perturbation learnability, which is grounded in the causality framework. We show a statistically significant inverse correlation between learnability and robustness.

2 Setup and Terminology

As a pilot study, we consider the task of binary text classification. The training set is denoted as $D_{train} = \{(x_1, l_1), ..., (x_n, l_n)\}$, where x_i is the i-th example and $l_i \in \{0, 1\}$ is the corresponding label. We fit a model $f : (x; \theta) \mapsto \{0, 1\}$ with parameters θ on the training data. A textual perturbation is a transformation $g : (x; \beta) \rightarrow x^*$ that injects a specific type of noise into an example x with parameters β and the resulting perturbed example is x^*. We design several experiment settings (Table 1) to answer our research questions. Experiment 0 in Table 1 is the standard learning setup, where we train and evaluate a model on the original dataset. Below we detail other experiment settings.

2.1 Definitions

Robustness. We apply the perturbations to test examples and measure the robustness of model to said perturbations as the decrease in accuracy. In Table 1, Experiment 1 is related to robustness measurement, where we train a model on unperturbed dataset and test it on perturbed examples. We denote the test accuracy of a model $f(\cdot)$ on examples perturbed by $g(\cdot)$ in Experiment 1 as $\mathcal{A}_1(f, g, D^*_{test})$. Similarly, the test accuracy in Experiment 0 is $\mathcal{A}_0(f, D_{test})$. Consequently, the robustness is calculated as the difference of test accuracies:

$$\text{robustness}(f, g, D) = \mathcal{A}_1(f, g, D^*_{test}) - \mathcal{A}_0(f, D_{test}). \tag{1}$$

Models usually suffer a performance drop when encountering perturbations, therefore the robustness is usually negative, where lower values indicate decreased robustness.

Improvement by Data Augmentation (Post Augmentation Δ). To improve robust accuracy (Tu et al., 2020) (i.e., accuracy on the perturbed test set), it is a common practice to leverage data augmentation (Li and Specia, 2019; Min et al., 2020; Tan and Joty, 2021). We simulate the data augmentation process by appending perturbed data to the training set (Experiment 2 of Table 1). We calculate the improvement on performance after data augmentation as the difference of test accuracies:

$$\Delta_{\text{post\_aug}}(f, g, D) = \mathcal{A}_2(f, g, D^*_{test}) - \mathcal{A}_1(f, g, D^*_{test}). \tag{2}$$

where $\mathcal{A}_2(f, g, D^*_{test})$ denotes the test accuracy of Experiment 2. $\Delta_{\text{post\_aug}}(f, g, D)$ is the higher the better.

Learnability. We want to compare perturbations in terms of how well the model *learns* to identify them with a small amount of evidence. We cast learnability estimation as a perturbation classification task, where a model is trained to identify the perturbation in an example. We define that the learnability estimation consists of three steps, namely ① **assigning random labels**, ② **perturbing with probabilities**, and ③ **estimating model**

Exp No.	Measurement	Label	Perturbation	Training Examples	Test Examples
0	Standard	original	$l \in \varnothing$	$(x_i, 0), (x_j, 1)$	$(x_i, 0), (x_j, 1)$
1	Robustness	original	$l \in \{0, 1\}$	$(x_i, 0), (x_j, 1)$	$(x_i^*, 0), (x_j^*, 1)$
2	Data Augmentation	original	$l \in \{0, 1\}$	$(x_i, 0), (x_j, 1)$ $(x_i^*, 0), (x_j^*, 1)$	$(x_i^*, 0), (x_j^*, 1)$
3	Learnability	random	$l' \in \{1'\}$	$(x_j, 0'), (x_i^*, 1')$	$(x_i^*, 1')$
4		random	$l' \in \{1'\}$	$(x_j, 0'), (x_i^*, 1')$	$(x_i, 1')$

Table 1: Example experiment settings for measuring learnability, robustness and improvement by data augmentation. We perturb an example if its label falls in the set of label(s) in "Perturbation" column. $\varnothing$ means no perturbation at all. Training/test examples are the expected input data, assuming we have only one negative $(x_i, 0)$ and positive $(x_j, 1)$ example in our original training/test set. l' is a random label and x^* is a perturbed example.

performance. Below we introduce the procedure and intuition for each step. This estimation framework is further grounded in concepts from the causality literature in Section 3, which justifies our motivations. We summarize our estimation approach formally in Algorithm 1 (Appendix A).

1. **Assigning Random Labels.** We randomly assign pseudo labels to each training example regardless of its original label. Each data point has equal probability of being assigned to positive ($l' = 1$) or negative ($l' = 0$) pseudo label. This results in a randomly labeled dataset $D'_{train} = \{(x_1; l'_1), ..., (x_n, l'_n)\}$, where $L' \sim Bernoulli(1, 0.5)$. In this way, we ensure that there is no difference between the two pseudo groups since the data are randomly split.

2. **Perturbing with Probabilities.** We apply the perturbation $g(\cdot)$ to each training example in one of the pseudo groups (e.g., $l' = 1$ in Algorithm 1)[1]. In this way, we create a correlation between the existence of perturbation and label (i.e., the perturbation occurrence is predictive of the label). We control the perturbation probability $p \in [0, 1]$, i.e., an example has a specific probability p of being perturbed. This results in a perturbed training set $D'^{*}_{train} = \{(x_1^*, l'_1), ..., (x_n^*, l'_n)\}$, where the perturbed example x_i^* is:

$$Z \sim U(0, 1), \forall i \in \{1, 2, ..., n\}$$
$$x_i^* = \begin{cases} g(x_i) & l'_i = 1 \wedge z < p, \\ x_i & \text{otherwise.} \end{cases} \quad (3)$$

Here Z is a random variable drawn from a uniform distribution $U(0, 1)$. Due to randomization in the formal step, now the only difference between the two pseudo groups is the occurrence of perturbation.

3. **Estimating Model Performance.** We train a model on the randomly labeled dataset with perturbed examples. Since the only difference between the two pseudo groups is the existence of the perturbation, the model is trained to identify the perturbation. The original test examples D_{test} are also assigned random labels and become D'_{test}. We perturb all of the test examples in one pseudo group (e.g., $l' = 1$, as in step 1) to produce a perturbed test set D'^{*}_{test}. Finally, the perturbation learnability is calculated as the difference of accuracies on D'^{*}_{test} and D'_{test}, which indicates how much the model learns from the perturbation's co-occurrence with pseudo label:

$$\text{learnability}(f, g, p, D) = \mathcal{A}_4(f, g, p, D'^{*}_{test}) - \mathcal{A}_3(f, g, p, D'_{test}). \quad (4)$$

$\mathcal{A}_4(f, g, p, D'^{*}_{test})$ and $\mathcal{A}_3(f, g, p, D'_{test})$ are accuracies measured by Experiment 4 and 3 of Table 1, respectively.

[1] Because the training data is randomly split into two pseudo groups, applying perturbations to any one of the groups should yield same result. We assume that we always perturb into the first group ($l' = 1$) hereafter.

We observe that the learnability depends on perturbation probability p. For each model–perturbation pair, we obtain multiple learnability estimates by varying the perturbation probability (Figure 4). However, we expect that learnability of the perturbation (as a concept) should be independent of perturbation probability. To this end, we use the $\log AUC$ (area under the curve in log scale) of the p – learnability curve (Figure 4), termed as "average learnability", which summarizes the overall learnability across different perturbation probabilities $p_1, ..., p_t$:

$$\text{avg\_learnability}(f, g, D) := \log AUC(\{(p_i, \text{learnability}(f, g, p_i, D)) \mid i \in \{1, 2, ..., t\}\}) \tag{5}$$

We use $\log AUC$ rather than AUC because we empirically find that the learnability varies substantially between perturbations when p is small, and a log scale can better capture this nuance. We also introduce learnability at a specific perturbation probability (Learnability @ p) as an alternate summary metric and provide a comparison of this metric against $\log AUC$ in Appendix E.

2.2 Hypothesis

With the above-defined terminologies, we propose hypotheses for RQ1 and RQ2 in Section 1, respectively.

Hypothesis 1 (H1): *A model for which a perturbation is more learnable is less robust against the same perturbation at the test time.*

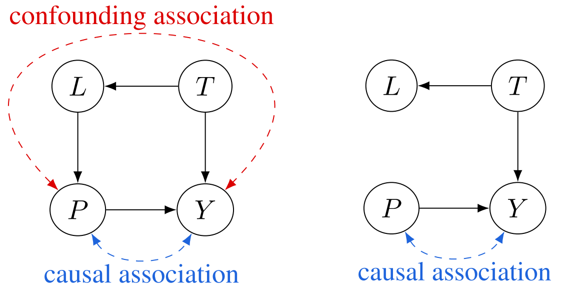
This is *not* obvious because the model encounters this perturbation during training in learnability estimation while they do not in robustness measurement.

Hypothesis 2 (H2): *A model for which a perturbation is more learnable experiences bigger robustness gains with data augmentation along such a perturbation.*

We validate both Hypotheses 1 and 2 with experiments on several perturbations and models described in Section 4.1 and 4.2.

3 A Causal View on Perturbation Learnability

In Section 2.1, we introduce the term "learnability" in an intuitive way. Now we map it to a formal, quantitative measure in standard statistical frameworks. Learnability is actually motivated by concepts from the causality literature. We provide a brief introduction to basic concepts of causal inference in Appendix C. In fact, learnability is the causal effect of perturbation on models, which is often difficult to measure due to the confounding latent features. In the language of causality, this is *"correlation is not causation"*. Causality provides insight on how to fully decouple the effect of perturbation and other latent features. We introduce the causal motivations for step 1 and 3 of learnability estimation in the following Section 3.1 and 3.2, respectively.



(a) Before randomization. (b) After randomization.

Figure 2: Causal graph explanation for decoupling perturbation and latent feature with randomization. P is the perturbation and T is the latent feature. L is the original label and Y is the correctness of the predicted label.

3.1 A Causal Explanation for Random Label Assignment

Natural noise (simulated by perturbations in this work) usually co-occurs with latent features in an example. If we did not assign random labels and simply perturbed one of the *original* groups, there would be confounding latent features that would prevent us from estimating the causal effect of the perturbation. Figure 2a illustrates this scenario. Both perturbation P and latent feature T may affect the outcome Y,[2] while the latent feature is predictive of label L. Since we make the perturbation P on examples with the same label, P is decided by L. It therefore follows that T is a confounder of the effect of P on Y, resulting in non-causal association flowing along the path $P \leftarrow L \leftarrow T \rightarrow Y$. However, if we do randomize the labels, P no longer has any causal parents (i.e., incoming edges) (Figure 2b). This is because perturbation is purely

[2] Y is later defined in Section 3.2

Perturbation	Example Sentence
None	His quiet and straightforward demeanor was rare then and would be today.
duplicate_punctuations	His quiet and straightforward demeanor was rare then and would be today..
butter_fingers_perturbation	His quiet and straightforward demeanor was rarw then and would be today.
shuffle_word	quiet would and was be and straightforward then demeanor His today. rare
random_upper_transformation	His quiEt and straightForwARd Demeanor was rare TheN and would be today.
insert_abbreviation	His quiet and straightforward demeanor wuz rare then and would b today.
whitespace_perturbation	His quiet and straightforward demean or wa s rare thenand would be today.
visual_attack_letters	Hiṣ qủiẽt ȁnd strḁighṭḟorwǎ̊rḍ dɛmeanoŕ wâṣ rȧre tḫen and wouḷd þə tɵḏầȳ.
leet_letters	His qui3t and strai9htfor3ard d3m3an0r 3as rar3 t43n and 30uld 63 t0da4.

Figure 3: An example sentence with different types of perturbations.

random. Without the path represented by $P \leftarrow L$, all of the association that flows from P to Y is causal. As a result, we can directly calculate the causal effect from the observed outcomes.

3.2 Learnability is a Causal Estimand

We identify learnability as a causal estimand. In causality, the term "identification" refers to the process of moving from a causal estimand (Average Treatment Effect, ATE) to an equivalent statistical estimand. We show that the difference of accuracies on D'^{*}_{test} and D'_{test} is actually a causal estimand. We define the outcome Y of a test example x_i as the correctness of the predicted label:

$$Y_i(0) \coloneqq \mathbf{1}_{\{f(x_i)=l'_i\}} \tag{6}$$

where $\mathbf{1}_{\{\cdot\}}$ is the indicator function. Similarly, the outcome Y of a perturbed test example x_i^* is:

$$Y_i(1) \coloneqq \mathbf{1}_{\{f(x_i^*)=l'_i\}} \tag{7}$$

According to the definition of Individual Treatment Effect (ITE, see Equation 9 of Appendix C), we have $ITE_i = \mathbf{1}_{\{f(x_i^*)=l'_i\}} - \mathbf{1}_{\{f(x_i)=l'_i\}}$. We then take the average over all the perturbed test examples (half of the test set)[3]. This is our Average Treatment Effect (ATE):

$$\begin{aligned} ATE &= E[Y(1)] - E[Y(0)] \\ &= E[\mathbf{1}_{\{f(x^*)=l'\}}] - E[\mathbf{1}_{\{f(x)=l'\}}] \\ &= P(f(x^*)=l') - P(f(x)=l') \\ &= \mathcal{A}(f,g,p,D'^{*}_{test}) - \mathcal{A}(f,g,p,D'_{test}) \end{aligned} \tag{8}$$

where $\mathcal{A}(f,g,p,D)$ is the accuracy of model $f(\cdot)$ trained with perturbation $g(\cdot)$ at perturbation probability p on test set D. Therefore, we show that ATE is exactly the difference of accuracy on the perturbed and unperturbed test sets with random labels. And the difference is learnability according to Equation 4.

We discuss another means of identification of ATE in Appendix D, based on the prediction probability. We compare between the probability-based and accuracy-based metrics there. We find that our accuracy-based metric yields better resolution, so we report this metric in the main text of this paper.

4 Experiments

4.1 Perturbation methods

Criteria for Perturbations. We select various character-level and word-level perturbation methods in existing literature that simulate different types of noise an NLP model may encounter in real-world situations. These perturbations are non-adversarial, label-consistent, and can be automatically generated at scale. We note that our perturbations do not require access to the model internal structure. We also assume that the feature of perturbation does not exist in the original data. Not all perturbations in the existing literature are suitable for our task. For example, a perturbation that swaps gender words (i.e., female → male, male → female) is not suitable for our experiments since we cannot distinguish the perturbed text from an unperturbed one. In other words, the perturbation function $g(\cdot)$ should be *asymmetric*, such that $g(g(x)) \neq x$.

Figure 3 shows an example sentence with different perturbations. Perturbation of "duplicate_punctuation" doubles the punctuation by appending a duplicate after each punctuation, e.g.,

[3] The other half of the test set ($l' = 0$) is left unperturbed, following the same procedure in Section 2.1. Model predictions will not change for unperturbed ones, resulting in ITEs with zero values. Therefore, we do not take them into account for ATE calculation.

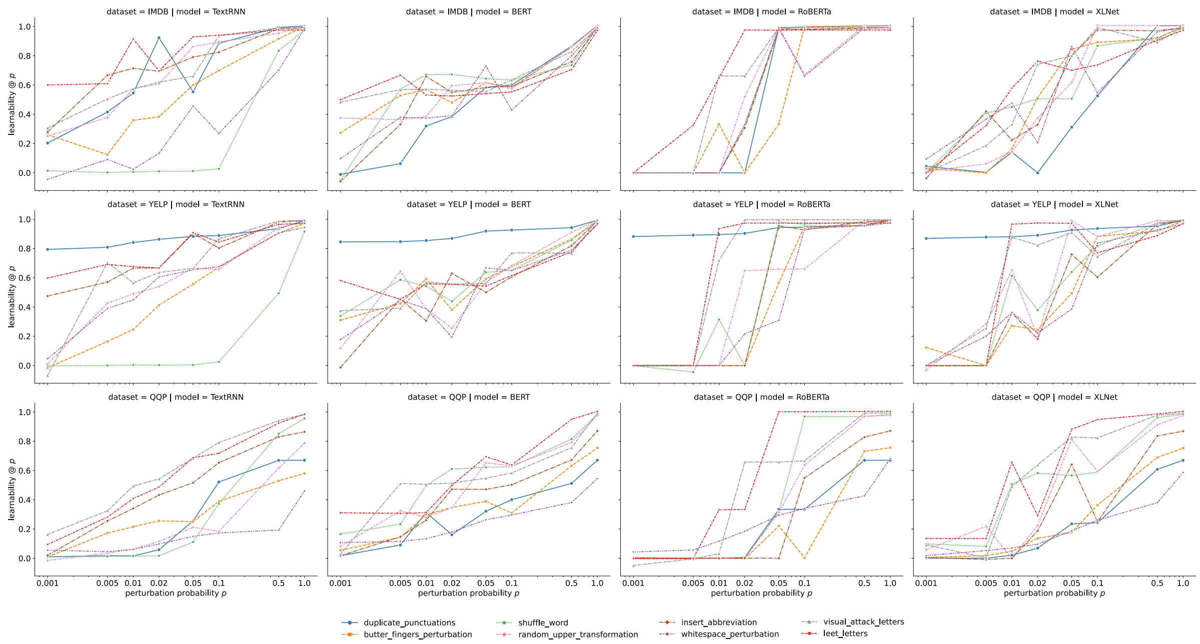


Figure 4: Learnability of eight perturbations for four NLP models on three datasets, as a function of perturbation probability.

"," → ",,"; "butter_fingers_perturbation" misspells some words with noise erupting from keyboard typos; "shuffle_word" randomly changes the order of word in the text (Moradi and Samwald, 2021); "random_upper_transformation" randomly adds upper cased letters (Wei and Zou, 2019); "insert_abbreviation" implements a rule system that encodes word sequences associated with the replaced abbreviations; "whitespace_perturbation" randomly removes or adds whitespaces to text; "visual_attack_letters" replaces letters with visually similar, but different, letters (Eger et al., 2019); "leet_letters" replaces letters with leet, a common encoding used in gaming (Eger et al., 2019).

4.2 Experimental Settings

To test the learnability, robustness and improvement by data augmentation with different NLP models and perturbations, we experiment with four modern and representative neural NLP models: TextRNN (Liu et al., 2016), BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019b) and XLNet (Yang et al., 2019). For TextRNN, we use the implementation by an open-source text classification toolkit NeuralClassifier (Liu et al., 2019a). For the other three pretrained models, we use the `bert-base-cased`, `roberta-base`, `xlnet-base-cased` versions from Hugging Face (Wolf et al., 2020), respectively. These two platforms support most of the common NLP models, thus facilitating extension studies of more models in future. We use three common binary text classification datasets — IMDB movie reviews (IMDB) (Pang and Lee, 2005), Yelp polarity reviews (YELP) (Zhang et al., 2015), Quora Question Pair (QQP) (Iyer et al., 2017) — as our testbeds. IMDB and YELP datasets present the task of sentiment analysis, where each sentence is labelled as positive or negative sentiment. QQP is a paraphrase detection task, where each pair of sentences is marked as semantically equivalent or not. To control the effect of dataset size and imbalanced classes, all datasets are randomly subsampled to the same size as IMDB (50k) with balanced classes. The training steps for all experiments are the same as well. We implement perturbations $g(\cdot)$ with two self-designed ones and six selected ones from the NL-Augmenter[4] library. For perturbation probabilities, we choose 0.001, 0.005, 0.01, 0.02, 0.05, 0.10, 0.50, 1.00. We run all experiments across three random seeds and report the average results.

4.3 Perturbation Learnability Analysis

Figure 4 shows learnability as a function of perturbation probability. Learnability @ p generally increases as we increase the perturbation probability, and when we perturb all the examples (i.e., p = 1.0), every model can easily identify it well,

[4] https://github.com/GEM-benchmark/NL-Augmenter

Perturbation	XLNet	RoBERTa	BERT	TextRNN	Average over models
whitespace_perturbation	1.638	1.436	1.492	0.878	1.361
shuffle_word	1.740	1.597	1.766	0.594	1.424
duplicate_punctuations	1.086	1.499	1.347	2.050	1.495
butter_fingers_perturbation	1.590	1.369	1.788	1.563	1.578
random_upper_transformation	1.583	1.520	1.721	2.039	1.716
insert_abbreviation	1.783	1.585	1.564	2.219	1.788
visual_attack_letters	**1.824**	1.921	**1.898**	2.094	1.934
leet_letters	1.816	**2.163**	1.817	**2.463**	**2.065**

Table 2: Average learnability (log AUC of corresponding curve in Figure 4) of each model–perturbation pair on IMDB dataset. Rows are sorted by average values over all models. The perturbation for which a model is most learnable is highlighted in **bold** while the following one is underlined.

ρ	IMDB	YELP	QQP
Avg. learnability vs. robustness	-0.643*	-0.821*	-0.695*
Avg. learnability vs. post aug Δ	0.756*	0.846*	0.750*

Table 3: Correlations of average learnability vs. robustness vs. post data augmentation Δ. ρ is Spearman correlation. * indicates high significance (p-value < 0.001).

resulting in the maximum learnability of 1.0. This shows that neural NLP models master these perturbations eventually. At lower perturbation probabilities, some models still learn that perturbation alone predicts the label. In fact, the major difference between different p – learnability curves is the area of lower perturbation probabilities and this provides motivation for using $\log AUC$ instead of AUC as the summarization of learnability at different p (Section 2.1).

Table 2 shows the average learnability over all perturbation probabilities of each model–perturbation pair on IMDB dataset in Figure 4.[5] It reveals the most learnable perturbation for each model. For example, the learnability of "visual_attack_letters" and "leet_letters" are very high for all four models, likely due to their strong effects on the tokenization process. Perturbations like "white_space_perturbation" and "duplicate_punctuations" are less learnable for pretrained models, probably because they have little effect on the subword level tokenization, or they may have encountered similar noise in the pretraining corpora. We observe that "duplicate_punctuations" already exists in the original text of YELP dataset (e.g., "*The burgers are awesome!!*"), thus violating our assumptions for perturbations in Section 4.1. As a result, the curve for this perturbation substantially deviates from others in Figure 4. We do not count this perturbation on YELP dataset in the following analysis. The perturbation learnability experiments provide a clean setup for NLP practitioners to analyze the effect of textual perturbations on models.

4.4 Empirical Findings

We observe a negative correlation between learnability (Equation 4) and robustness (Equation 1) across all three datasets in Table 2, validating Hypothesis 1. Table 2 also quantifies the trend that data augmentation with a perturbation the model is *less* robust to has *more* improvement on robustness (Hypothesis 2).[6] Both the correlations between 1) learnability vs. robustness and 2) learnability vs. improvement by data augmentation are strong (Spearman $|\rho| > 0.6$) and highly significant (p-value < 0.001), which firmly supports our hypotheses. Our findings provide insight about when the model is less robust and when data augmentation works better for improving robustness.

Figure 1 shows that the more learnable a perturbation is for a model, the greater the likelihood that its robustness can be improved through data augmentation along this perturbation. We argue that this is not simply because there is more room for

[5] Please refer to Appendix F for benchmark results on YELP (Table 5) and QQP (Table 6) datasets.

[6] For visualizations of correlations, please refer to Figure 5 for IMDB, Figure 6 for YELP and Figure 7 for QQP in Appendix F.

improvement by data augmentation. From a causal perspective, learnability acts as a common cause (confounder) for both robustness and improvement by data augmentation. This indicates a potential limitation of using data augmentation for improving robustness to perturbations (Jha et al., 2020): for unlearnable perturbations, data augmentation may be of little help. Approaches that go beyond simple data augmentation are required to combat such perturbations.

5 Discussion

Potential Impacts. Our findings seem intuitive but are non-trivial. The NLP models were not trained on perturbed examples when measuring robustness, but still they display a strong correlation with perturbation learnability. Understanding these findings are important for a more principled evaluation of and control over NLP models (Lovering et al., 2020). Specifically, the learnability metric complements to the evaluation of newly designed perturbations by revealing model weaknesses in a clean setup. Reducing perturbation learnability is promising for improving robustness of models. Contrastive learning (Gao et al., 2021; Yan et al., 2021) that pulls the representations of the original and perturbed text together, makes it difficult for the model to identify the perturbation (reducing learnability) and thus may help improve robustness. Moreover, learnability may facilitate the development of model architectures with explicit inductive biases (Warstadt and Bowman, 2020; Lovering et al., 2020) to avoid sensitivity to noisy perturbations. Grounding the learnability within the causality framework inspires future researchers to incorporate the causal perspective into model design (Zhang et al., 2020), and make the model robust to different types of perturbations.

Limitations. We note that this work has not established that the relationship between learnability and robustness is *causal*. This could be explored with other approaches in causal inference for deconfounding besides simulation on randomized control trial, such as working with real data but stratifying it (Frangakis and Rubin, 2002), to bring the learnability experiment closer to more naturalistic settings. Although we restrict to balanced, binary classification for simplicity in this pilot study, our framework can also be extended to imbalanced, multi-class classification. We are aware that computing average learnability is expensive for large models and datasets, which is further discussed in Appendix B. We provide a greener solution in Appendix E. We could further verify our assumptions for perturbations with a user study (Moradi and Samwald, 2021) which investigates how understandable the perturbed texts are to humans.

6 Related Work

Robustness of NLP Models to Perturbations. The performance of NLP models can decrease when encountering noisy data in the real world. Recent works (Prabhakaran et al., 2019; Ribeiro et al., 2020; Niu et al., 2020; Moradi and Samwald, 2021) present comprehensive evaluations of the robustness of NLP models to different types of perturbations, including typos, changed entities, negation, etc. Their results reveal the phenomenon that NLP models can handle some specific types of perturbation more effectively than others. However, they do not go into a deeper analysis of the reason behind the difference of robustness between models and perturbations.

Interpretation of Data Augmentation. Although data augmentation has been widely used in CV (Sato et al., 2015; DeVries and Taylor, 2017; Dwibedi et al., 2017) and NLP (Wang and Yang, 2015; Kobayashi, 2018; Wei and Zou, 2019), the underlying mechanism of its effectiveness remains under-researched. Recent studies aim to quantify intuitions of how data augmentation improves model generalization. Gontijo-Lopes et al. (2020) introduce affinity and diversity, and find a correlation between the two metrics and augmentation performance in image classification. In NLP, Kashefi and Hwa (2020) propose a KL-divergence–based metric to predict augmentation performance. Our proposed learnability metric implies when data augmentation works better and thus acts as a complement to this line of research.

7 Conclusion

This work targets at an open question in NLP: why models are less robust to some textual perturbations than others? We find that learnability, which causally quantifies how well a model learns to identify a perturbation, is predictive of the model robustness to the perturbation. In future work, we will investigate whether these findings can generalize to other domains, including computer vision.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Terrance DeVries and Graham W Taylor. 2017. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*.

Debidatta Dwibedi, Ishan Misra, and Martial Hebert. 2017. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1310–1319. IEEE Computer Society.

Steffen Eger, Gözde Gül Şahin, Andreas Rücklé, Ji-Ung Lee, Claudia Schulz, Mohsen Mesgar, Krishnkant Swarnkar, Edwin Simpson, and Iryna Gurevych. 2019. Text processing like humans do: Visually attacking and shielding NLP systems. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1634–1647, Minneapolis, Minnesota. Association for Computational Linguistics.

Constantine E Frangakis and Donald B Rubin. 2002. Principal stratification in causal inference. *Biometrics*, 58(1):21–29.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Karan Goel, Nazneen Fatema Rajani, Jesse Vig, Zachary Taschdjian, Mohit Bansal, and Christopher Ré. 2021. Robustness gym: Unifying the nlp evaluation landscape. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*, pages 42–55.

Raphael Gontijo-Lopes, Sylvia Smullin, Ekin Dogus Cubuk, and Ethan Dyer. 2020. Tradeoffs in data augmentation: An empirical study. In *International Conference on Learning Representations*.

Paul W Holland. 1986. Statistics and causal inference. *Journal of the American statistical Association*, 81(396):945–960.

Shankar Iyer, Nikhil Dandekar, and Kornel Csernai. 2017. First quora dataset release: Question pairs.

Rohan Jha, Charles Lovering, and Ellie Pavlick. 2020. Does data augmentation improve generalization in nlp? *arXiv preprint arXiv:2004.15012*.

Omid Kashefi and Rebecca Hwa. 2020. Quantifying the evaluation of heuristic methods for textual data augmentation. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 200–208.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457.

Zhenhao Li and Lucia Specia. 2019. Improving neural machine translation robustness via data augmentation: Beyond back-translation. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 328–336, Hong Kong, China. Association for Computational Linguistics.

Liqun Liu, Funan Mu, Pengyu Li, Xin Mu, Jing Tang, Xingsheng Ai, Ran Fu, Lifeng Wang, and Xing Zhou. 2019a. NeuralClassifier: An open-source neural hierarchical multi-label text classification toolkit. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 87–92, Florence, Italy. Association for Computational Linguistics.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 2873–2879.

Xiao Liu, Da Yin, Yansong Feng, Yuting Wu, and Dongyan Zhao. 2021. Everything has a cause: Leveraging causal inference in legal text analysis. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1928–1941.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Charles Lovering, Rohan Jha, Tal Linzen, and Ellie Pavlick. 2020. Predicting inductive biases of pre-trained models. In *International Conference on Learning Representations*.

Junghyun Min, R Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. 2020. Syntactic data augmentation increases robustness to inference heuristics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2339–2352.

Milad Moradi and Matthias Samwald. 2021. Evaluating the robustness of neural language models to input perturbations.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, Online. Association for Computational Linguistics.

Brady Neal. 2020. Introduction to causal inference from a machine learning perspective. *Course Lecture Notes (draft)*.

Xing Niu, Prashant Mathur, Georgiana Dinu, and Yaser Al-Onaizan. 2020. Evaluating robustness to input perturbations for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8538–8544.

Bo Pang and Lillian Lee. 2005. Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 115–124.

Vinodkumar Prabhakaran, Ben Hutchinson, and Margaret Mitchell. 2019. Perturbation sensitivity analysis to detect unintended model biases. 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 5740–5745.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912.

Donald B Rubin. 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688.

Ikuro Sato, Hiroki Nishimura, and Kensuke Yokoi. 2015. Apac: Augmented pattern classification with neural networks. *arXiv preprint arXiv:1505.03229*.

Samson Tan and Shafiq Joty. 2021. Code-mixing on sesame street: Dawn of the adversarial polyglots. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3596–3616, Online. Association for Computational Linguistics.

Lifu Tu, Garima Lalwani, Spandana Gella, and He He. 2020. An empirical study on robustness to spurious correlations using pre-trained language models. *Transactions of the Association for Computational Linguistics*, 8:621–633.

William Yang Wang and Diyi Yang. 2015. That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using #petpeeve tweets. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2557–2563, Lisbon, Portugal. Association for Computational Linguistics.

Xiao Wang, Qin Liu, Tao Gui, Qi Zhang, et al. 2021. Textflint: Unified multilingual robustness evaluation toolkit for natural language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 347–355, Online. Association for Computational Linguistics.

Alex Warstadt and Samuel R Bowman. 2020. Can neural networks acquire a structural bias from raw linguistic data? In *Proceedings of the Annual Meeting of the Cognitive Science Society*.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. 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 6382–6388.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. Consert: A contrastive framework for self-supervised sentence representation transfer. *arXiv preprint arXiv:2105.11741*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Cheng Zhang, Kun Zhang, and Yingzhen Li. 2020. A causal view on robustness of neural networks. *Advances in Neural Information Processing Systems*, 33.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28:649–657.

A Algorithm for Perturbation Learnability Estimation

Algorithm 1 Learnability Estimation

Input: training set $D_{train} = \{(x_1, l_1), ..., (x_n, l_n)\}$, test set $D_{test} = \{(x_{n+1}, l_{n+1}), ..., (x_{n+m}, l_{n+m})\}$, $D = D_{train} \cup D_{test}$, model $f : (x; \theta) \mapsto \{0, 1\}$, perturbation $g : (x; \beta) \rightarrow x^*$, perturbation probability p
Output: learnability(f, g, p, D)

// ① assigning random labels
Initialize an empty dataset D'
for i in $\{1, 2, ..., n+m\}$ **do**
  $l'_i \leftarrow randint[0, 1]$
  $D' \leftarrow D' \cup \{(x_i, l'_i)\}$
end for
// ② perturbing with probabilities
Initialize an empty dataset D'^*
for i in $\{1, 2, ..., n+m\}$ **do**
  $z \leftarrow rand(0, 1)$
  $x^*_i \leftarrow x_i$
  if $l'_i = 1 \wedge z < p$ **then**
    $x^*_i \leftarrow g(x_i)$
  end if
  $D'^* \leftarrow D'^* \cup \{(x^*_i, l'_i)\}$
end for
// ③ estimating model performance
$D'_{train}, D'_{test} \leftarrow D'[1:n], D'[n+1:n+m]$
$D'^*_{train}, D'^*_{test} \leftarrow D'^*[1:n], D'^*[n+1:n+m]$

fit the model $f(\cdot)$ on D'^*_{train}
$\mathcal{A}(f, g, p, D'^*_{test}) \leftarrow f(\cdot)$ accuracy on D'^*_{test}
$\mathcal{A}(f, g, p, D'_{test}) \leftarrow f(\cdot)$ accuracy on D'_{test}
return $\mathcal{A}(f, g, p, D'^*_{test}) - \mathcal{A}(f, g, p, D'_{test})$

B Ethics Statement

Computing average learnability requires training a model for multiple times at different perturbation probabilities, which can be computationally intensive if the sizes of the datasets and models are large. This can be a non-trivial problem for NLP practitioners with limited computational resources. We hope that our benchmark results of typical perturbations for NLP models work as a reference for potential users. Collaboratively sharing the results of such metrics on popular models and perturbations in public fora can also help reduce duplicate investigation and coordinate efforts across teams.

To alleviate the computational efficiency issue of average learnability estimation, using learnability at selected perturbation probabilities may help at the cost of reduced precision (Appendix E). We are not alone in facing this issue: two similar metrics for interpreting model inductive bias, *extractability* and *s-only error* (Lovering et al., 2020) also require training the model repeatedly over the whole dataset. Therefore, finding an efficient proxy for average learnability is promising for more practical use of learnability in model interpretation.

C Background on Causal Inference

Causal Inference. The aim of causal inference is to investigate how a treatment T affects the outcome Y. Confounder X refers to a variable that influences both treatment T and outcome Y. For example, sleeping with shoes on (T) is strongly associated with waking up with a headache (Y), but they both have a common cause: drinking the night before (X) (Neal, 2020). In our work, we aim to study how a perturbation (treatment) affects the model's prediction (outcome). However, the latent features and other noise usually act as confounders.

Causality offers solutions for two questions: 1) how to eliminate the spurious association and isolate the treatment's causal effect; and 2) how varying T affects Y, given both variables are causally-related (Liu et al., 2021). We leverage both of these properties in our proposed method. Let us now introduce Randomized Controlled Trial and Average Treatment Effect as key concepts in answering the above two questions, respectively.

- **Randomized Controlled Trial (RCT).** In an RCT, each participant is randomly assigned to either the treatment group or the non-treatment group. In this way, the only difference between the two groups is the treatment they receive. Randomized experiments ideally guarantee that there is no confounding factor, and thus any observed association is actually causal. We operationalize RCT as a perturbation classification task in Section 3.1.

- **Average Treatment Effect (ATE).** In Section 3.2, we apply ATE (Holland, 1986) as a measure of learnability. ATE is based on Individual Treatment Effect (ITE, Equation 9), which is the difference of the outcome with and without treatment.

$$ITE_i = Y_i(1) - Y_i(0) \qquad (9)$$

Here, $Y_i(1)$ is the outcome Y of individual i that receives treatment ($T = 1$), while $Y_i(0)$ is the opposite. In the above example, waking up with a headache ($Y = 1$) with shoes on ($T = 1$) means $Y_i(1) = 1$.

We calculate the Average Treatment Effect (ATE) by taking an average over ITEs:

$$ATE = E[Y(1)] - E[Y(0)] \tag{10}$$

ATE quantifies how the outcome Y is expected to change if we modify the treatment T from 0 to 1. We provide specific definitions of ITE and ATE in Section 3.2.

D Alternate Definition of Perturbation Learnability

In Section 3.2, we propose an accuracy-based identification of ATE. Now we discuss another probability-based identification and compare between them. We can also define the outcome Y of a test example x_i as the predicted probability of (pseudo) true label given by the trained model $f(\cdot)$:

$$Y_i(0) \coloneqq P_f(L' = l'_i \mid X = x_i) \in (0, 1) \tag{11}$$

Similarly, the performance outcome Y of a perturbed test data point x_i^* is:

$$Y_i(1) \coloneqq P_f(L' = l'_i \mid X = x_i^*) \in (0, 1) \tag{12}$$

For example, for a test example (x_i, l'_i) which receives treatment ($l'_i = 1$), the trained model $f(\cdot)$ predicts its label as 1 with only a small probability 0.1 before treatment (it has not been perturbed yet), and 0.9 after treatment. So the Individual Treatment Effect (ITE, see Equation 9) of this example is calculated as $ITE_i = Y_i(1) - Y_i(0) = 0.9 - 0.1 = 0.8$. We then take an average over all the perturbed test examples (half of the test set)[7] as Average Treatment Effect (ATE, see Equation 10), which is exactly the learnability of a perturbation for a model. To clarify, the two operands in Equation 10 are defined as follows:

$$E[Y(1)] \coloneqq \mathcal{P}(f, g, p, D'^{*}_{test}) \tag{13}$$

It means the average predicted probability of (pseudo) true label given by the trained model $f(\cdot)$ on the perturbed test set D'^{*}_{test}.

$$E[Y(0)] \coloneqq \mathcal{P}(f, g, p, D'_{test}) \tag{14}$$

[7]The other half of the test set ($l' = 0$) is left unperturbed, following the same procedure in Section 2.1. Therefore, we do not take them into account for ATE calculation.

Similarly, this is the average predicted probability on the randomly labeled test set D'_{test}.

Notice that the accuracy-based definition of outcome Y (Equation 6) can also be written in a similar form to the probability-based one (Equation 11):

$$Y_i(0) \coloneqq \mathbf{1}_{\{f(x_i)=l'_i\}} = \mathbf{1}_{\{P_f(L'=l'_i|X=x_i)>0.5\}} \in \{0, 1\} \tag{15}$$

because the correctness of the prediction is equal to whether the predicted probability of true (pseudo) label exceeds a certain threshold (i.e., 0.5).

The major difference is that, accuracy-based ITE is a discrete variable falling in $\{-1, 0, 1\}$, while probability-based ITE is a continuous one ranging from -1 to 1. For example, if a model learns to identify a perturbation and thus changes its prediction from wrong (before perturbation) to correct (after perturbation), accuracy-based ITE will be $1 - 0 = 1$ while probability-based ITE will be less than 1. That is to say, accuracy-based ATE tends to vary more drastically than probability-based if inconsistent predictions occur more often, and thus can better capture the nuance of perturbation learnability. Empirically, we find that accuracy-based average learnability varies greatly ($\sigma = 0.375$, Table 4) and thus can better distinguish between different model-perturbation pairs than probability-based one ($\sigma = 0.288$, Table 4). As a result, we choose accuracy-based ATE as the primary measurement of learnability in this paper.

E Investigating Learnability at a Specific Perturbation Probability

Inspired by Precision @ K in Information Retrieval (IR), we propose a similar metric dubbed Learnability @ p, which is the learnability of a perturbation for a model at a specific perturbation probability p. We are primarily interested in whether a selected p can represent the learnability over different perturbation probabilities and correlates well with robustness and post data augmentation Δ.

We calculate the standard deviation (σ) of Learnability @ p and average learnability ($\log AUC$) over all model-perturbation pairs to measure how well it can distinguish between different models and perturbations. Table 4 shows that average learnability is more diversified than all Learnability @ p and diversity (σ) peaks at $p = 0.01$ for accuracy-based/probability-based measurement. Accuracy-based Learnability @ p is generally more diversi-

p	Accuracy-based Learnability @ p				Probability-based Learnability @ p			
	σ	Avg Learn.	Robu.	Post Aug Δ	σ	Avg Learn.	Robu.	Post Aug Δ
Avg.	0.375	1.000*	-0.643*	0.756*	0.288	1.000*	-0.652*	0.727*
0.001	0.182	0.426*	-0.265	0.259	0.114	0.367*	-0.279	0.288
0.005	0.235	0.637*	-0.383*	0.522*	0.192	0.925*	-0.620*	0.702*
0.01	0.263	0.741*	-0.530*	0.635*	0.192	0.893*	-0.567*	0.586*
0.02	0.257	0.816*	-0.636*	0.743*	0.192	0.886*	-0.686*	0.690*
0.05	0.236	0.279	-0.158	0.136	0.121	0.576*	-0.371*	0.350*
0.1	0.241	0.354*	-0.162	0.192	0.115	0.543*	-0.288	0.258
0.5	0.094	0.024	0.155	-0.179	0.037	-0.080	0.114	-0.258
1.0	0.011	-0.199	0.252	-0.332	0.019	-0.220	0.294	-0.402*

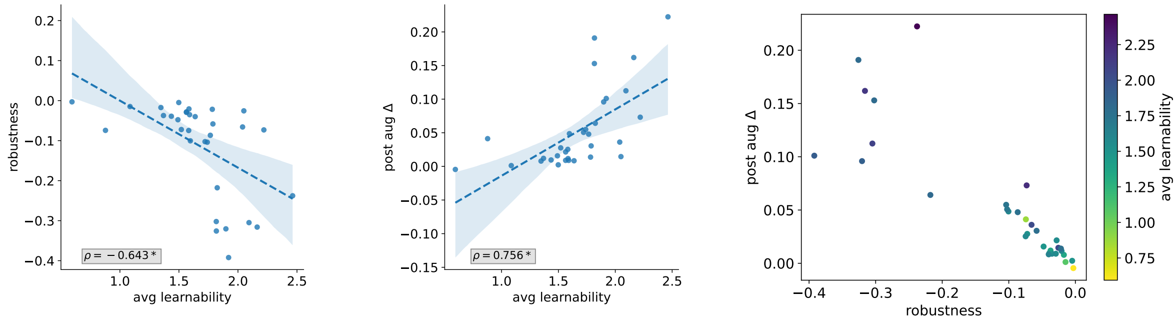
Table 4: Standard deviations (σ) of Learnability @ p and Spearman correlations between accuracy-based/probability-based learnability @ p vs. average learnability/robustness/post data augmentation Δ over all model-perturbation pairs on IMDB dataset. * indicates significance (p-value < 0.05).

fied across models and perturbations than its counterpart.

To investigate the strength of the correlations, we also calculate Spearman ρ between accuracy-based/probability-based learnability @ p vs. average learnability/robustness/post data augmentation Δ over all model-perturbation pairs. Table 4 shows that generally average learnability has stronger correlation than Learnability @ p. Correlations with both robustness and post data augmentation Δ peak at p = 0.02 for accuracy-based/probability-based measurements, and the correlations with average learnability (0.816*/0.886*) are also strong at these perturbation probabilities.

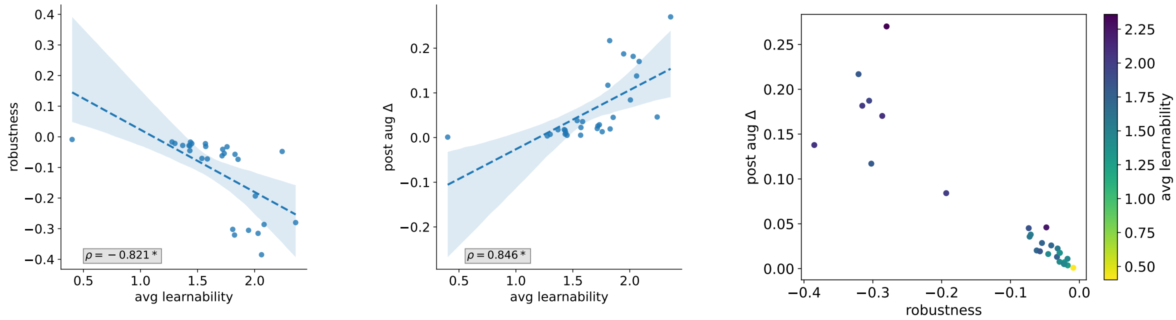
Overall, Learnability @ p with higher standard deviation correlates better with average learnability, robustness and post data augmentation Δ. Our analysis shows that if p is carefully selected by σ, Learnability @ p is also a promising metric, though not as accurate as average learnability. One advantage of Learnability @ p over average learnability is that it costs less time to obtain learnability at a single perturbation probability. We plan to explore other efficient proxies of average learnability in future.

F Additional Experiment Results



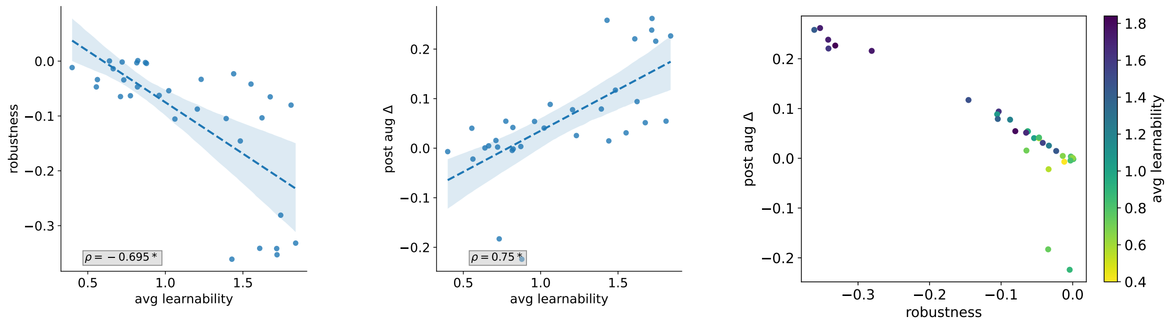
(a) Learnability vs. Robustness (b) Learnability vs. Post Aug Δ (c) Learn. vs. Robu. vs. Post Aug Δ

Figure 5: Linear regression plots of learnability vs. robustness vs. post data augmentation Δ on IMDB dataset. Each point in the plots represents a model-perturbation pair. ρ is Spearman correlation. * indicates high significance (p-value < 0.001).



(a) Learnability vs. Robustness (b) Learnability vs. Post Aug Δ (c) Learn. vs. Robu. vs. Post Aug Δ

Figure 6: Linear regression plots of learnability vs. robustness vs. post data augmentation Δ on YELP dataset. Each point in the plots represents a model-perturbation pair. ρ is Spearman correlation. * indicates high significance (p-value < 0.001).



(a) Learnability vs. Robustness (b) Learnability vs. Post Aug Δ (c) Learn. vs. Robu. vs. Post Aug Δ

Figure 7: Linear regression plots of learnability vs. robustness vs. post data augmentation Δ on QQP dataset. Each point in the plots represents a model-perturbation pair. ρ is Spearman correlation. * indicates high significance (p-value < 0.001).

Perturbation	RoBERTa	XLNet	TextRNN	BERT	Average over models
shuffle_word	1.538	1.586	0.401	**1.854**	1.345
butter_fingers_perturbation	1.301	1.433	1.425	1.758	1.479
whitespace_perturbation	1.276	1.449	1.720	1.569	1.504
insert_abbreviation	1.437	1.370	<u>2.241</u>	1.572	1.655
random_upper_transformation	1.432	1.828	1.733	1.715	1.677
visual_attack_letters	<u>2.060</u>	**2.006**	2.030	1.808	<u>1.976</u>
leet_letters	**2.083**	<u>1.947</u>	**2.359**	<u>1.824</u>	**2.053**

Table 5: Average learnability ($\log AUC$ of corresponding curve in Figure 4) of each model–perturbation pair on YELP dataset. Rows are sorted by average values over all models. The perturbation for which a model is most learnable is highlighted in **bold** while the following one is <u>underlined</u>.

Perturbation	RoBERTa	TextRNN	XLNet	BERT	Average over models
whitespace_perturbation	0.732	0.399	0.562	0.711	0.601
duplicate_punctuations	0.722	0.823	0.640	0.872	0.764
butter_fingers_perturbation	0.555	0.878	0.775	1.022	0.808
insert_abbreviation	0.820	1.440	0.960	1.206	1.107
random_upper_transformation	1.062	0.664	1.392	1.483	1.150
shuffle_word	1.231	0.816	1.552	<u>1.623</u>	1.306
visual_attack_letters	<u>1.429</u>	**1.810**	<u>1.744</u>	1.608	<u>1.648</u>
leet_letters	**1.720**	<u>1.676</u>	**1.840**	**1.718**	**1.738**

Table 6: Average learnability ($\log AUC$ of corresponding curve in Figure 4) of each model–perturbation pair on QQP dataset. Rows are sorted by average values over all models. The perturbation for which a model is most learnable is highlighted in **bold** while the following one is <u>underlined</u>.