Model-tuning Via Prompts Makes NLP Models Adversarially Robust

Anonymous Authors[1]

Abstract

In recent years, NLP practitioners have converged on the following practice: (i) import an off-the-shelf pretrained (masked) language model; (ii) append a multilayer perceptron atop the CLS token's hidden representation (with randomly initialized weights); and (iii) fine-tune the entire model on a downstream task (`MLP-FT`). This procedure has produced massive gains on standard NLP benchmarks, but these models remain brittle, even to mild adversarial perturbations, such as word-level synonym substitutions. In this work, we demonstrate surprising gains in adversarial robustness enjoyed by Model-tuning Via Prompts (`MVP`), an alternative method of adapting to downstream tasks. Rather than modifying the model (by appending an MLP head), `MVP` instead modifies the input (by appending a prompt template). Across three classification datasets, `MVP` improves performance against adversarial word-level synonym substitutions by an average of 8% over standard methods and even outperforms adversarial training-based state-of-art defenses by 3.5%. By combining `MVP` with adversarial training, we achieve further improvements in robust accuracy while maintaining clean accuracy. Finally, we conduct ablations to investigate the mechanism underlying these gains. Notably, we find that the main causes of vulnerability of `MLP-FT` can be attributed to the misalignment between pre-training and fine-tuning tasks, and the randomly initialized MLP parameters.[1]

1. Introduction

Pre-trained NLP models (Devlin et al., 2019; Liu et al., 2019) are typically adapted to downstream tasks by (i) appending a randomly initialized multi-layer perceptron to their topmost representation layer; and then (ii) fine-tuning the resulting model on downstream data (`MLP-FT`). More recently, work on large language models has demonstrated comparable performance without fine-tuning, by just prompting the model with a prefix containing several examples of inputs and corresponding target values (Brown et al., 2020). More broadly, prompting approaches recast classification problems as sequence completion (or mask infilling) tasks by embedding the example of interest into a prompt template. The model's output is then mapped to a set of candidate answers for final prediction. Prompting has emerged as an effective strategy for large-scale language models (Lester et al., 2021), and its utility has also been demonstrated for masked language models (Gao et al., 2021).

While fine-tuned models perform well on in-distribution data, a growing body of work demonstrates that they remain brittle to adversarial perturbations (Jin et al., 2020; Li et al., 2020; Morris et al., 2020a). Even small changes in the input text, such as replacement with synonyms (Ebrahimi et al., 2018b), and adversarial misspellings (Ebrahimi et al., 2018a; Pruthi et al., 2019) drastically degrade the accuracy of text classification models. While prompting has emerged as a popular approach for adapting pretrained models to downstream data, little work has considered interactions between adaptation strategies and adversarial robustness.

In this work, we demonstrate surprising benefits of Model-tuning Via Prompts (`MVP`) in terms of adversarial robustness to word substitution attacks, compared to fine-tuning models with an MLP head (`MLP-FT`). In `MVP`, all of the parameters of the model are fine-tuned through prompts. Surprisingly, `MVP`, which does not utilize any sort of adversarial training or prompt optimization[2] already yields higher adversarial robustness compared to the state-of-the-art methods utilizing adversarial training by an average of 3.5% across three tasks, two models and two attacks (§4). Moreover, we find that combining `MVP` with single-step adversarial training can further boost adversarial robustness, resulting in combined robustness gains of more than 10% over the baselines. This happens without any loss in clean accuracy, indicating how the objective of adversarial training couples well with `MVP`.

So far, prior works have not explored the idea of full-model full-data fine-tuning via prompts. We only see instances of

[1]Link to code has been removed to preserve anonymity.

[2]The process of finding optimal prompts that maximize downstream performance is referred to as prompt engineering.

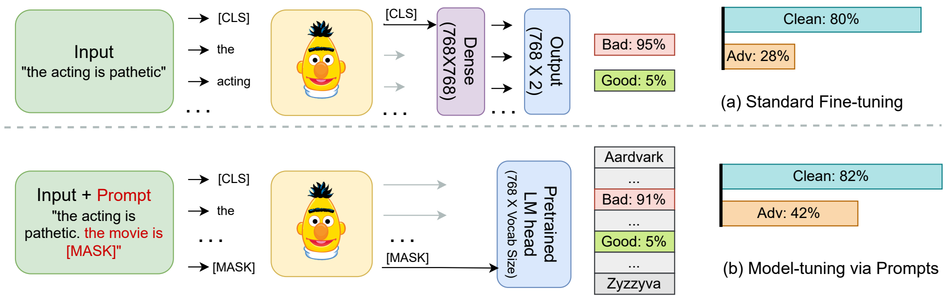


Figure 1: An illustration of (a) Standard Finetuning, and (b) Model-tuning via Prompts. The adjoining accuracy metrics correspond to a RoBERTa model trained on the BoolQ dataset.

(i) few-shot full-model fine-tuning via prompts (Gao et al., 2021), or (ii) partial-model full-data finetuning (Li & Liang, 2021) (in the context of large language models). The idea of full-model full-data fine-tuning via prompts has not been used until now, possibly because clean accuracy improvements for `MVP` over `MLP-FT` are negligible, and the robustness advantages of `MVP` were previously undiscovered.

Additionally, we show (§4.1) that `MVP` is more (i) sample efficient (requires fewer training samples to achieve the same clean accuracy), and (ii) has higher effective robustness than `MLP-FT` (for any given clean accuracy, the robust accuracy of `MVP` is higher than `MLP-FT`). Through ablation studies (§4.2), we find that adding (i) multiple prompt templates makes it harder to fool the model; and (ii) multiple candidate answers has a small but positive impact on the robustness.

To explain our observations, we test a set of hypotheses (§5), including (i) *random parameter vulnerability*—is adding a randomly initialized linear head the source of adversarial vulnerability for `MLP-FT`?; (ii) *pretraining task alignment*—can the gains in robustness be attributed to the alignment between the fine-tuning and pretaining tasks in `MVP`?; and (iii) *candidate semanticity*—are predictions by `MVP` more robust because the candidate answer is semantically similar to the class label? Through experiments designed to test these hypotheses, we find that (i) in the absence of pretraining, `MVP` and `MLP-FT` have similar robustness performance, supporting the hypothesis of pretraining task alignment; (ii) adding extra uninitialized parameters to `MVP` leads to a sharp drop in robustness, whereas removing the dense $(768, 768)$ randomly initialized weight matrix from `MLP-FT` improves the robustness of the model significantly; (iii) even random candidate answers such as 'jack', and 'jill' result in similar robustness gains, suggesting that when fine-tuning through prompts, the choice of candidate answers is inconsequential (in contrast, the choice of candidates is known to be important for few-shot classification).

We also perform a user study (§F) to assess the quality of adversarial examples on which `MVP` + Adv fails. We find that human annotators were 23% more likely to find adversarial examples to have been perturbed as opposed to clean examples. Moreover, humans achieved 11% lower accuracy on adversarial examples as compared to clean examples with average confidence on the label of perturbed examples being 15% lower. This highlights that a large fraction of adversarial examples are already detected by humans, and often change the true label of the input, signifying that `MVP` is more robust than crude statistics discussed in §4. Future work will benefit from developing better evaluation strategies for the robustness of NLP models.

In summary, we demonstrate that models tuned via prompts (`MVP`) are considerably more robust than the models adapted through the conventional approach of fine-tuning with an MLP head while maintaining similar clean performance. Our findings suggest that practitioners adopt `MVP` as a means of fine-tuning, regardless of the training data size (few-shot or full data) and model capacity.

2. Method

We consider the task of supervised text classification, where we have a dataset $\mathcal{S} = \{x^{(i)}, y^{(i)}\}^n$, with $x^{(i)} \in \mathcal{X}$ and $y^{(i)} \in \{1, \ldots, k\}$ for a k-class classification problem. We train a classifier f to predict y based on input x. *Prompt Template* (t) is the input string that we append at the beginning or end of the input. For example, we may append the following template at the end of a movie review—"This movie is `[MASK]`". *Candidate Answers* $(\mathcal{A})$ is a set of tokens corresponding to each class. For example, the positive sentiment class can have $\mathcal{A}$ ={great, good, amazing} (Liu et al., 2021).

Adversarial Attacks We are concerned with perturbations to the input x that change the model prediction. Let $\Delta(x)$ be the set of all feasible perturbed inputs, then

$$x_{adv} = \arg\max_{\hat{x} \in \Delta(x)} \ell(\hat{x}, y, f).$$

In case of adversarial attacks confined to synonym substitutions, $\Delta(x) = \tilde{S}_1 \times \tilde{S}_2 \times \cdots \times \tilde{S}_k$, where $\tilde{S}_i$ is the set of permissible synonyms of the word x_i including itself.

2.1. Model-tuning Via Prompts (MVP)

We present the overall pipeline of MVP in Figure 1(b), and describe individual component below.

Input Modification Consider a prompt template $t = t_1, t_2, \ldots$ [MASK]$, \ldots t_m$. For any input x, the prompt input (x_t) can be constructed by appending the template at the beginning or end of the input. The final output is based on the most likely substitution for the [MASK] token, as given by the language model. Typically, we use a set of prompt templates denoted by $\mathcal{T}$.

Inference For a given class c, consider the candidate answer set $\mathcal{A}_c = \{a_{1,c}, a_{2,c}, \ldots, a_{k_c,c}\}$. The output logit for class c is computed as follows:

$$p(y = c|x) = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \max_{i \in [k_c]} p(\texttt{[MASK]} = a_{i,c}|x_t).$$

We use the language modeling head to calculate $p(\texttt{[MASK]} = a_{i,c}|x_t)$. The final predicted output label is $\hat{y} = \mathrm{argmax}_c\ p(y = c|x)$. In other words, we do the following: (i) select the candidate corresponding to the highest probability for a given class label; (ii) take mean of the probabilities of the selected candidates over all the templates to compute the final logit of the given class label; (iii) predict the class having the highest final logit.

2.2. MVP + Single-step Adv

Based on the Fast Gradient Sign Method (FGSM) by Goodfellow et al. (2014), we perform single-step adversarial training. Note that the input tokens are discrete vectors, and hence it is not possible to perturb the inputs directly. Instead, we pass the inputs through the embedding layer of the model and then perform adversarial perturbations in the embedding space. We do not perturb the embeddings corresponding to the prompt tokens. We find that performing single-step perturbations with the ℓ_2 constraint leads to more stable training than in the ℓ_∞ norm ball, and use the same for all our experiments. Similar (but not equivalent) approaches have also been studied in literature (Si et al., 2021a).

3. Experimental Setup

Detailed information about training and attack hyperparameters is provided in Appendix E.

Datasets and Models We perform our experiments on five different datasets—AG News (Zhang et al., 2015b) (4-class topic classification), SST-2 (Socher et al., 2013) (binary sentiment classification), BoolQ (Clark et al., 2019) (boolean question answering), DBPedia14 (Zhang et al., 2015a) (14-class topic classification), and MRPC (Dolan & Brockett, 2005) (paraphrase detection). Results on DBPedia14 and MRPC are presented in Appendix D.1. All models are trained with the BERT-Base (Devlin et al., 2019) and RoBERTa-Base (Zhuang et al., 2021) backbone. Experiments on GPT-2 are included in Appendix D.2

Attack Strategies We perturb the input examples using the TextAttack library (Morris et al., 2020b). In particular, we use 1 character-level attack and 3 word-level attacks. Word-level attacks include the TextFooler (Jin et al., 2020), TextBugger (Li et al., 2018) and BertAttack (Li et al., 2020) attack strategies.[3] They are greedy word substitution attacks that replace words with neighboring words based on counter-fitted GloVe embeddings. For character-level, we use adversarial misspellings (Pruthi et al., 2019). Results on Adversarial Misspellings and BERTAttack are in Appendix D.1.

3.1. Baseline Methods

For our evaluations, we compare our method to MLP-FT, MLP-FT + Adv, FreeLB++ (Li et al., 2021), InfoBert (Wang et al., 2021a) and AMDA (Si et al., 2021b), the details of which are provided in §A

4. Results

For the task of Boolean question answering (BoolQ), we find that fine-tuning a RoBERTa model with an MLP head (MLP-FT) achieves an accuracy of 28.2% on adversarial examples obtained through the TextFooler attack strategy (Table 1). Whereas, the corresponding accuracy for tuning the model via prompts (MVP) is 42.9% which is a considerable improvement over MLP-FT. Additionally, MVP leads to more robust models compared to adversarial training baselines like MLP-FT + Adv and InfoBERT that attain accuracies of 39.0% and 38.1% respectively. Further, MVP can be combined with adversarial training (MVP + adv), and doing so leads to an accuracy of 52.2% which is about a 10% improvement over MVP, without any loss in clean performance.

Similar to BoolQ, the robustness advantages of MVP hold across all tasks we examine. The individual performance statistics are detailed in Table 1. Overall, across two models (BERT & RoBERTa), two attack strategies, and three datasets, we report that MVP improves over MLP-FT by 8%. Remarkably, even in the absence of any adversarial training MVP achieves the state-of-the-art adversarial performance improving baseline adversarial training methods by 3.5%. Moreover, it can be coupled with single-step adversarial training, resulting in an overall 7% improvement over state-

[3] In line with previous benchmark (Li et al., 2021) we only use the word-substitution transformation in TextBugger.

	AG News					
	BERT-Base			RoBERTa-Base		
	Clean Acc	TextFooler	TextBugger	Clean Acc	TextFooler	TextBugger
MLP-FT	93.76 ± 0.46	37.53 ± 0.67	58.97 ± 0.67	94.50 ± 0.40	42.86 ± 0.74	61.80 ± 0.30
MLP-FT + Adv	93.23 ± 0.23	44.34 ± 0.98	64.12 ± 0.23	94.40 ± 0.61	47.67 ± 0.51	65.60 ± 0.78
Free LB++	93.40 ± 0.20	43.53 ± 0.21	63.43 ± 0.78	94.37 ± 0.68	46.93 ± 1.60	65.56 ± 1.00
AMDA	92.83 ± 0.55	41.80 ± 0.87	62.63 ± 1.04	94.10 ± 0.62	44.30 ± 1.41	62.90 ± 0.51
InfoBERT	93.83 ± 0.30	43.97 ± 1.60	64.08 ± 0.78	94.50 ± 0.89	48.00 ± 2.25	65.63 ± 1.20
MVP	93.70 ± 0.46	46.27 ± 1.15	65.97 ± 0.35	94.33 ± 0.21	51.46 ± 2.06	68.73 ± 0.70
MVP + Adv	93.97 ± 0.59	**53.73 ± 0.06**	**69.17 ± 1.27**	94.43 ± 0.81	**62.73 ± 2.35**	**75.33 ± 1.60**
	BoolQ					
	BERT-Base			RoBERTa-Base		
	Clean Acc	TextFooler	TextBugger	Clean Acc	TextFooler	TextBugger
MLP-FT	71.13 ± 1.34	21.77 ± 4.38	36.80 ± 3.00	80.60 ± 1.56	28.23 ± 1.68	38.36 ± 1.09
MLP-FT + Adv	70.98 ± 0.91	29.78 ± 0.78	42.78 ± 1.34	78.86 ± 1.26	39.00 ± 0.72	44.40 ± 1.25
Free LB++	70.73 ± 0.15	29.50 ± 0.61	42.83 ± 0.63	80.63 ± 0.49	37.27 ± 1.47	43.23 ± 1.05
AMDA	71.06 ± 0.91	25.37 ± 0.76	41.60 ± 0.61	79.20 ± 0.95	32.03 ± 0.32	41.10 ± 0.20
InfoBERT	71.77 ± 0.55	29.86 ± 0.25	42.60 ± 0.56	81.50 ± 0.70	38.07 ± 1.37	42.47 ± 0.96
MVP	71.43 ± 1.00	31.13 ± 1.27	44.40 ± 2.78	82.00 ± 0.60	42.93 ± 0.57	49.86 ± 1.67
MVP + Adv	71.27 ± 0.72	**43.10 ± 0.70**	**49.93 ± 0.90**	81.07 ± 0.60	**52.23 ± 1.62**	**56.46 ± 1.60**
	SST2					
	BERT-Base			RoBERTa-Base		
	Clean Acc	TextFooler	TextBugger	Clean Acc	TextFooler	TextBugger
MLP-FT	91.97 ± 0.20	38.32 ± 1.01	60.41 ± 0.48	93.58 ± 0.40	40.25 ± 0.94	65.37 ± 0.28
MLP-FT + Adv	90.98 ± 0.34	42.89 ± 1.23	62.34 ± 0.52	93.63 ± 0.63	44.04 ± 1.24	68.47 ± 1.47
Free LB++	92.16 ± 0.84	42.25 ± 1.01	63.05 ± 0.71	94.05 ± 0.09	43.37 ± 1.00	67.15 ± 0.64
AMDA	92.18 ± 0.89	41.72 ± 0.57	60.96 ± 0.44	93.84 ± 0.42	41.85 ± 0.46	66.06 ± 0.17
InfoBERT	91.79 ± 0.67	43.15 ± 0.81	64.69 ± 0.66	94.00 ± 0.40	43.63 ± 0.52	66.58 ± 1.77
MVP	91.78 ± 0.46	44.67 ± 0.76	65.16 ± 0.05	93.92 ± 0.70	46.88 ± 0.50	69.80 ± 0.51
MVP + Adv	91.80 ± 0.74	**47.67 ± 0.58**	**67.77 ± 0.39**	93.82 ± 0.12	**53.78 ± 0.72**	**71.73 ± 0.85**

Table 1: Adversarial performance of BERT-base and RoBERTa-base models on 3 different datasets averaged over 3 seeds.

of-art methods. Lastly, the robustness benefits come only at a 2x computation cost of standard training, as opposed to past works which need 5–10x computation cost of standard training due to additional adversarial training.

4.1. Sample Efficiency & Effective Robustness

We investigate the sample efficiency and effective robustness of MVP through experiments on the BoolQ and AG-News datasets using the RoBERTa-base model. We randomly sample small fractions of the dataset, ranging from $5e^{-4}$ to $1e^{-1}$, and train MLP-FT and MVP on the same.

Sample Efficiency We compare the performance of MVP and MLP-FT in low-data regimes. We find that MVP results in models are consistently more robust compared to models trained through MLP-FT in the low data setups (see Figure 2a). In fact, we observe that in extremely low resource case (only 60 examples), it is hard to learn using MLP-FT , but model trained through MVP performs ex-

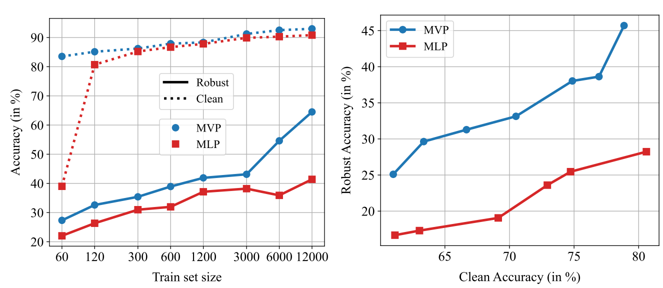


Figure 2: (a) Sample Efficiency: Clean & Robust Accuracy of RoBERTa-base model when trained using different data sizes of the AG News dataset. (b) Effective Robustness: Clean vs Robust Accuracy on the BoolQ dataset. We find that (a) MVP is more sample efficient as compared to MLP-FT , and (b) MVP yields more robustness compared to MLP-FT for the same clean accuracy (see §4.1 for details).

			BoolQ			AGNews		
Experiment	# Templates	Candidate	Clean	TFooler	TBugger	Clean	TFooler	TBugger
Template Expansion	1	Class Label	81.9 ± 0.8	35.9 ± 0.2	44.6 ± 0.5	94.6 ± 0.4	48.6 ± 1.1	67.3 ± 1.1
	2	Class Label	82.3 ± 0.2	37.4 ± 0.3	46.4 ± 0.5	94.5 ± 0.6	50.8 ± 1.6	67.8 ± 0.5
	3	Class Label	82.1 ± 0.3	40.8 ± 1.5	49.5 ± 1.1	94.2 ± 0.2	48.4 ± 3.4	66.2 ± 1.1
	4	Class Label	82.0 ± 0.6	42.9 ± 0.5	49.8 ± 1.6	94.3 ± 0.2	51.4 ± 2.0	68.7 ± 0.7
Candidate Exp.	4	Multiple	81.6 ± 1.2	46.1 ± 1.6	53.0 ± 0.7	93.6 ± 0.4	54.0 ± 0.7	69.8 ± 0.3

Table 2: We study the impact of the number of candidate answers and prompt templates on adversarial performance (see §4.2). Additionally, we also assess the effect of including semantically similar answer candidates (see §5). All values are averaged over 3 seeds.

ceedingly well. We note that the relative benefit of MVP over MLP-FT peaks around 5–10% of the data. Interestingly, the model trained through MVP requires only 5% of samples to achieve similar robustness levels as models trained with MLP-FT on the full dataset. In addition to robustness benefits, we find that MVP achieves considerably higher clean accuracy in low-data regimes (i.e., with < 200 examples). Results for BoolQ are presented in D.3.

Effective Robustness Past work has observed that scaling of both model and data size (Hoffmann et al., 2022; Lester et al., 2021) result in models that perform better in-distribution. Effective robustness (Taori et al., 2021) measures the robust accuracy of models that have the same clean accuracy. This can help determine which training time design decisions will be valuable when scaled up. We measure the effective robustness for models trained through MVP and MLP-FT by training them on different data sizes. We find that even when both MLP-FT and MVP achieve the same clean accuracy, models trained through MVP are more robust (Figure 2b). Results for AG News are presented in D.3

4.2. Ablation Studies

Number of Candidate Answers A larger candidate answer set is shown to improve clean performance in the few-shot setting (Hu et al., 2022). Here, we investigate the impact of the size of the candidate answer set on the adversarial performance of models tuned via prompts. The adversarial accuracy of the model with a single candidate answer is 42.9%, and it increases to 46.2% upon using an answer set comprising 4 candidates.[4] These results correspond to the RoBERTa-base model on BoolQ dataset against perturbations by the TextFooler attack. Overall, we observe an improvement of 1.0–3.5% in adversarial accuracy when we use a larger candidate set across different settings (Table 2).

Number of Prompt Templates Another design choice that we consider is the number of prompt templates used for prediction. We conjecture that the adversary may find it difficult to flip the model prediction when we average logits across multiple templates. To evaluate this, we train MVP with different number of prompt templates (ranging from 1 to 4), and compare the adversarial robustness. We notice a steady improvement in the adversarial accuracy as we increase the number of templates which supports our initial conjecture (see Table 2). While increasing the number of templates improves the robustness of the downstream model, MVP achieves large robustness gains even with a single template (compared to MLP-FT). Hence, using multiple prompt templates is not the fundamental reason for the improved robustness of MVP.

[4]Details about candidates and templates are in Appendix C

5. Why Does MVP Improve Robustness?

Random Parameter Vulnerability One plausible explanation for the observed adversarial vulnerability of MLP-FT is the randomly-initialized linear head used for downstream classification. The intuition behind this effect is that *fine-tuning a set of randomly-initialized parameters may lead to feature distortion of the pretrained model* as is demonstrated in Kumar et al. (2022). This phenomenon has also been observed in CLIP models (Radford et al., 2021), where the authors found that fine-tuning the model using a randomly initialized linear prediction head reduces the out-of-distribution robustness of the model. The phenomenon is unexplored in the context of adversarial robustness. We study this effect through three experiments.

1. ProjectCLS: First, we reduce the number of random parameters by removing the dense layer of weights (768×768 parameters) from the standard MLP architecture. We call this ProjectCLS, and only use a projection layer of dimensions $768 \times C$ parameters, with C being the number of classes (see Figure 4(a)). We find that ProjectCLS is on average $\sim 8\%$ more robust than MLP-FT which suggests that reducing the number of randomly initialized parameters helps to increase model robustness (see Table 3).

2. CLSPrompt: Second, we train another model, CLSPrompt, where instead of using the probabilities corresponding to the [MASK] token as in MVP, we use the

		BoolQ			AGNews		
Hypothesis	Setting	Clean	TFooler	TBugger	Clean	TFooler	TBugger
Random Parameter	MLP-FT	80.6 ± 1.5	28.2 ± 1.6	38.3 ± 1.0	94.5 ± 0.4	42.8 ± 0.7	61.8 ± 0.3
	ProjectCLS	81.3 ± 0.5	37.4 ± 1.2	45.6 ± 1.2	93.7 ± 0.4	46.7 ± 1.3	65.2 ± 3.3
	CLSPrompt	82.4 ± 0.3	36.5 ± 0.4	46.0 ± 1.2	94.7 ± 0.2	47.2 ± 1.9	66.7 ± 2.0
	DenseLPFT	81.3 ± 0.4	33.9 ± 1.4	42.6 ± 1.2	94.5 ± 0.6	44.2 ± 0.8	64.5 ± 1.1
	LPFT	81.6 ± 1.2	37.5 ± 1.1	46.4 ± 1.2	94.5 ± 0.1	46.5 ± 0.9	67.2 ± 1.0
Task Alignment	Untrained MVP	67.5 ± 0.9	11.7 ± 2.7	14.9 ± 2.7	90.1 ± 0.8	12.2 ± 2.9	20.6 ± 2.2
	Untrained MLP-FT	67.0 ± 0.6	14.8 ± 4.3	17.5 ± 1.1	89.5 ± 0.4	13.4 ± 1.2	19.4 ± 0.8
Candidate Semantics	Random (MVP)	80.9 ± 0.3	42.1 ± 0.4	48.1 ± 2.2	93.4 ± 0.3	50.3 ± 1.2	68.3 ± 0.3

Table 3: Adversarial performance of the RoBERTa model for experiments corresponding to the random parameter vulnerability and task alignment hypotheses (§5).

probabilities of the candidate answers corresponding to the `[CLS]` token (see Figure 4(b)). The key difference between `CLSPrompt` and `MLP-FT` is that there are no randomly initialized MLP parameters in `CLSPrompt`, and we use the probabilities corresponding to the candidate answers, instead of projecting the representations with new parameters. From Table 3, we observe that `CLSPrompt` is once again on average $\sim 8\%$ more robust than `MLP-FT` which provides strong evidence in favor of our hypothesis of random parameter vulnerability.

3. `LPFT` (linear probe, then fine-tune): For our third experiment, we train two new models namely `LPFT` and `DenseLPFT` (see Figure 4(c,d)). In both these models, we do the following: (i) fit a logistic regression to the hidden states corresponding to the `[CLS]` token (linear probing); (ii) initialize the final layer of the classification head with the learned $768 \times C$ (where C is the number of classes) matrix of the fitted logistic regression model; and (iii) fine-tune the whole model as in `MLP-FT`. The only difference between `LPFT` and `DenseLPFT` is that `DenseLPFT` has an additional randomly initialized dense layer of dimensions 768×768 unlike `LPFT`. In contrast to Kumar et al. (2022), we test `LPFT` against adversarial manipulations. We note from Table 3 that `DenseLPFT` is more robust than `MLP-FT` (by over 10%) but it demonstrates lower robustness as compared to `LPFT` (by over 2%). This provides further evidence that randomly initialized parameters add to the vulnerability.

Pretraining Task Alignemnt The task of mask infilling aligns more naturally with the pretraining objective of the language model and we posit that finetuning via mask infilling as in `MVP` results in robustness gains. To test this hypothesis, we use an untrained RoBERTa model, and measure the clean accuracy and robustness of `MVP` and `MLP-FT` models. We observe that in the absence of pre-training, `MVP` trained with a single template does not achieve any additional robustness over the baseline, and infact `MLP-FT` performs better than `MVP`(Table 3) whereas in the presence of pre-training, `MVP` outperforms `MLP-FT` (Table 2) in all the settings. Note that this does not contradict the hypothesis about vulnerability due to randomly-initialized parameters, as that hypothesis only applies for pretrained models. This suggests that the alignment of `MVP` with the pre-training task is crucial for adversarial robustness on downstream task.

Semantically Similar Candidates We question whether the improvement in robustness can be attributed to the semanticity between candidate answers and the class labels. To answer this question, we change the candidate answers to random proper nouns ('jack', 'john', 'ann', 'ruby') for the 4-class classification problem of AG-News and ('jack', 'john') for the 2-class classification task of BoolQ. All of these words are unrelated to the class labels. We find that irrespective of whether we use semantically related candidates or not, the robust accuracy of the model is within 1% of each other, thereby implying that using semantically similar candidates is not a factor behind the robustness gains of `MVP` (Table 3). While the choice of candidate answers is crucial in the pre-train, prompt, and predict paradigm (Hu et al., 2022), it is irrelevant in the pre-train, prompt, and fine-tune paradigm. With sufficient fine-tuning over the downstream corpus, a model can learn to associate any candidate word with any class, irrespective of its semanticity.

6. Conclusion

In this work, we benchmark the robustness of masked language models when adapted to downstream classification tasks through prompting. Remarkably, `MVP`—which does note even utilize any sort of adversarial training or prompt engineering—already outperforms the state-of-the-art methods in adversarially robust text classification by over 3.5% on average. Moreover, we find that `MVP` is sample efficient and also exhibits high *effective* robustness as compared to the conventional approach (`MLP-FT`). We find that the lack of robustness in baseline methods can largely be attributed to the lack of alignment between pre-training and finetuning task, and the introduction of new randomly parameters.

References

Alzantot, M., Sharma, Y., Elgohary, A., Ho, B.-J., Srivastava, M., and Chang, K.-W. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2890–2896, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1316. URL https://aclanthology.org/D18-1316.

Belinkov, Y. and Bisk, Y. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BJ8vJebC-.

Boucher, N., Shumailov, I., Anderson, R., and Papernot, N. Bad characters: Imperceptible nlp attacks. In *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 1987–2004. IEEE, 2022.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M., and Toutanova, K. BoolQ: Exploring the surprising difficulty of natural yes/no questions. 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)*, pp. 2924–2936, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1300. URL https://aclanthology.org/N19-1300.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. 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)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Dolan, W. B. and Brockett, C. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005. URL https://aclanthology.org/I05-5002.

Ebrahimi, J., Lowd, D., and Dou, D. On adversarial examples for character-level neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 653–663, Santa Fe, New Mexico, USA, August 2018a. Association for Computational Linguistics. URL https://aclanthology.org/C18-1055.

Ebrahimi, J., Rao, A., Lowd, D., and Dou, D. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 31–36, Melbourne, Australia, July 2018b. Association for Computational Linguistics. doi: 10.18653/v1/P18-2006. URL https://aclanthology.org/P18-2006.

Gao, T., Fisch, A., and Chen, D. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 3816–3830, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.295. URL https://aclanthology.org/2021.acl-long.295.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2014.

Goyal, S., Kumar, A., Garg, S., Kolter, Z., and Raghunathan, A. Finetune like you pretrain: Improved finetuning of zero-shot vision models, 2022. URL https://arxiv.org/abs/2212.00638.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. d. L., Hendricks, L. A., Welbl, J., Clark, A., et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Hu, S., Ding, N., Wang, H., Liu, Z., Wang, J., Li, J., Wu, W., and Sun, M. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2225–2240, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.158. URL https://aclanthology.org/2022.acl-long.158.

Jia, R. and Liang, P. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2021–2031, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1215. URL `https://aclanthology.org/D17-1215`.

Jia, R., Raghunathan, A., Göksel, K., and Liang, P. Certified robustness to adversarial word substitutions. 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)*, pp. 4129–4142, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1423. URL `https://aclanthology.org/D19-1423`.

Jiang, Z., Xu, F. F., Araki, J., and Neubig, G. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438, 2020.

Jin, D., Jin, Z., Zhou, J. T., and Szolovits, P. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8018–8025, Apr. 2020. doi: 10.1609/aaai.v34i05.6311. URL `https://ojs.aaai.org/index.php/AAAI/article/view/6311`.

Kumar, A., Raghunathan, A., Jones, R. M., Ma, T., and Liang, P. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=UYneFzXSJWh`.

Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.243. URL `https://aclanthology.org/2021.emnlp-main.243`.

Levine, Y., Wies, N., Jannai, D., Navon, D., Hoshen, Y., and Shashua, A. The inductive bias of in-context learning: Rethinking pretraining example design. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=lnEaqbTJIRz`.

Li, J., Ji, S., Du, T., Li, B., and Wang, T. Textbugger: Generating adversarial text against real-world applications. *arXiv preprint arXiv:1812.05271*, 2018.

Li, L., Ma, R., Guo, Q., Xue, X., and Qiu, X. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6193–6202, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.500. URL `https://aclanthology.org/2020.emnlp-main.500`.

Li, X. L. and Liang, P. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.353. URL `https://aclanthology.org/2021.acl-long.353`.

Li, Z., Xu, J., Zeng, J., Li, L., Zheng, X., Zhang, Q., Chang, K.-W., and Hsieh, C.-J. Searching for an effective defender: Benchmarking defense against adversarial word substitution. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3137–3147, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.251. URL `https://aclanthology.org/2021.emnlp-main.251`.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*, 2021.

Liu, X., Ji, K., Fu, Y., Tam, W., Du, Z., Yang, Z., and Tang, J. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 61–68, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-short.8. URL `https://aclanthology.org/2022.acl-short.8`.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Morris, J., Lifland, E., Yoo, J. Y., Grigsby, J., Jin, D., and Qi, Y. 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*, pp. 119–126, Online, October 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.

emnlp-demos.16. URL https://aclanthology.org/2020.emnlp-demos.16.

Morris, J., Lifland, E., Yoo, J. Y., Grigsby, J., Jin, D., and Qi, Y. 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*, pp. 119–126, 2020b.

Pruthi, D., Dhingra, B., and Lipton, Z. C. Combating adversarial misspellings with robust word recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5582–5591, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1561. URL https://aclanthology.org/P19-1561.

Qin, G. and Eisner, J. Learning how to ask: Querying LMs with mixtures of soft prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5203–5212, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.410. URL https://aclanthology.org/2021.naacl-main.410.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. 2019.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Rubin, O., Herzig, J., and Berant, J. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2655–2671, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.191. URL https://aclanthology.org/2022.naacl-main.191.

Shin, T., Razeghi, Y., Logan IV, R. L., Wallace, E., and Singh, S. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4222–4235, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.346. URL https://aclanthology.org/2020.emnlp-main.346.

Si, C., Yang, Z., Cui, Y., Ma, W., Liu, T., and Wang, S. Benchmarking robustness of machine reading comprehension models. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 634–644, Online, August 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.56. URL https://aclanthology.org/2021.findings-acl.56.

Si, C., Zhang, Z., Qi, F., Liu, Z., Wang, Y., Liu, Q., and Sun, M. Better robustness by more coverage: Adversarial and mixup data augmentation for robust finetuning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 1569–1576, Online, August 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.137. URL https://aclanthology.org/2021.findings-acl.137.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1170.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. Intriguing properties of neural networks. *CoRR*, abs/1312.6199, 2013.

Taori, R., Dave, A., Shankar, V., Carlini, N., Recht, B., and Schmidt, L. Measuring robustness to natural distribution shifts in image classification. In *NeurIPS 2021 Workshop on ImageNet: Past, Present, and Future*, 2021. URL https://openreview.net/forum?id=DiC4hoZHO_2.

Wang, B., Wang, S., Cheng, Y., Gan, Z., Jia, R., Li, B., and Liu, J. Info{bert}: Improving robustness of language models from an information theoretic perspective. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=hpH98mK5Puk.

Wang, X., Hao, J., Yang, Y., and He, K. Natural language adversarial defense through synonym encoding. In *Uncertainty in Artificial Intelligence*, pp. 823–833. PMLR, 2021b.

Wang, Y. and Bansal, M. Robust machine comprehension models via adversarial training. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 575–581, New Orleans, Louisiana, June 2018. Association for

Computational Linguistics. doi: 10.18653/v1/N18-2091. URL https://aclanthology.org/N18-2091.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6.

Wortsman, M., Ilharco, G., Kim, J. W., Li, M., Kornblith, S., Roelofs, R., Lopes, R. G., Hajishirzi, H., Farhadi, A., Namkoong, H., et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7959–7971, 2022.

Zhang, X., Zhao, J., and LeCun, Y. Character-level convolutional networks for text classification. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015a. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/250cf8b51c773f3f8dc8b4be867a9a02-Paper.pdf.

Zhang, X., Zhao, J. J., and LeCun, Y. Character-level convolutional networks for text classification. In *NIPS*, 2015b.

Zhou, K., Yang, J., Loy, C. C., and Liu, Z. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16816–16825, 2022a.

Zhou, K., Yang, J., Loy, C. C., and Liu, Z. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022b.

Zhu, C., Cheng, Y., Gan, Z., Sun, S., Goldstein, T., and Liu, J. Freelb: Enhanced adversarial training for natural language understanding. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BygzbyHFvB.

Zhuang, L., Wayne, L., Ya, S., and Jun, Z. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pp. 1218–1227, Huhhot, China, August 2021. Chinese Information Processing Society of China. URL https://aclanthology.org/2021.ccl-1.108.

Supplementary Material
Model-tuning Via Prompts Makes NLP Models More Robust

A. Baseline Methods

MLP-FT : This is the "base" model for classification via standard non-adversarial training, and is utilized by all the baselines discussed in this section. Given a pretrained model, we perform downstream fine-tuning by adding an MLP layer to the output corresponding to [CLS] token as illustrated in Figure 1(a). This hidden representation is of size 768×1. In the case of the BERT model, there is a single dense layer of dimension 768×2, whereas in the case of RoBERTa model, we have a two-layer MLP that is used to make the final prediction.

MLP-FT + Adv: This is identical to the method used for adversarial training in Section 2.2, wherein we perform adversarial perturbations in the embedding space of the MLP-FT model, rather than MVP.

FreeLB++ (Li et al., 2021): Free Large-Batch (FreeLB) adversarial training (Zhu et al., 2020) performs multiple Projected Gradient Descent (PGD) steps to create adversarial examples, and simultaneously accumulates parameter gradients which are then used to update the model parameters (all at once). FreeLB++ improves upon FreeLB by increasing the number of adversarial training steps to 10 and the max adversarial norm to 1.

InfoBERT (Wang et al., 2021a): InfoBERT uses an Information Bottleneck regularizer to suppress noisy information that may occur in adversarial attacks. Alongside, an 'anchored feature regularizer' tries to align local stable features to the global sentence vector. InfoBERT is additionally combined with adversarial training using Free LB++.

AMDA (Si et al., 2021b): Adversarial and Mixup Data Augmentation (AMDA) improves robustness to adversarial attacks by increasing the number of adversarial samples seen during training. This method interpolates training examples in their embedding space to create new training examples. The label assigned to the new example is the linear interpolation of the one hot encodings of the original labels.

B. Related Work

Adversarial Attacks and Defenses Inspired by the brittleness of vision models to adversarial examples (Szegedy et al., 2013; Goodfellow et al., 2014), researchers have found similar vulnerabilities to also exist in language models (Alzantot et al., 2018; Belinkov & Bisk, 2018). Unlike vision, the goal in NLP is to develop (i) semantically viable substitutions or deletions (Ebrahimi et al., 2018b); (ii) character-level misspellings (Zhang et al., 2015b; Ebrahimi et al., 2018a; Pruthi et al., 2019); or (iii) imperceptible homoglyphs (Boucher et al., 2022).

The discovery of such adversarial examples span several tasks such as classification (Zhang et al., 2015b; Alzantot et al., 2018), NMT (Belinkov & Bisk, 2018), and question-answering (Jia & Liang, 2017), but they are restricted to small models such as LSTMs and RNNs. Among others, Jin et al. (2020); Li et al. (2020) show that despite producing massive gains on standard NLP benchmarks, BERT style pretrained models are susceptible to adversarial attacks when finetuned on downstream tasks. Subsequently, multiple works have attempted at developing fast and semantically meaningful attacks (Li et al., 2018) and scalable defenses (Wang & Bansal, 2018; Jia et al., 2019; Wang et al., 2021b; Si et al., 2021b; Zhu et al., 2020) for masked language models. Despite these efforts, NLP models suffer a significant drop in robust accuracy, when compared to clean accuracy on the same task.

Prompting NLP Models Prompting gained traction from GPT-3 (Brown et al., 2020) where it was primarily used in the zero-shot and few-shot settings and required manual trials to increase performance. In the zero-shot setting, no labeled examples are provided to the model and the language model is kept frozen. The model needs to output its prediction using the prompt that is provided. Whereas, in the few-shot setting, a few task-specific labeled examples are also provided for the frozen model in addition to the prompt (also known as in-context learning) (Rubin et al., 2022; Levine et al., 2022). A lot of work has gone into improving the prompts that are used in the zero-shot and few-shot settings, including mining-based methods to automatically augment prompts (Jiang et al., 2020), gradient-based search (Shin et al., 2020), using generative language models (Gao et al., 2021) and others (Hu et al., 2022). In the full data setting, previous works have explored prompting via prompt tuning (Liu et al., 2022; Li & Liang, 2021; Qin & Eisner, 2021) where the model is injected with additional tunable parameters.

Robust Fine-tuning and Adaptation In the vision literature, prior works have also tried to use prompting to improve out-of-distribution robustness in the zero-shot and few-shot settings (Zhou et al., 2022a;b). Kumar et al. (2022) observed that fine-tuning worsens the out-of-distribution (OOD) performance of models due to the bias introduced via a randomly-initialized head on top of the CLIP model, and instead suggest a procedure (LPFT) that first fits the linear head and then finetunes the model. Later works have shown that this ID/OOD performance trade-off could be mitigated by averaging model weights between the original zero-shot and fine-tuned model (Wortsman et al., 2022) and/or by finetuning using an objective similar to that used for pretraining (Goyal et al., 2022). However, this work has been applied only to vision–language models, and secondly only deals with "natural" robustness evaluations rather than the adversarial manipulations we consider here.

C. Candidate Answers & Prompt Templates

We enumerate all the prompt templates and candidate answers used for our experiments on `MVP`. We prefix the prompt template with the `[SEP]` token at the beginning. Note that since Causal Language models are not bidirectional, for GPT-2 experiments, all the prompt templates will be appended at the end of the input.

AG News The prompt templates used for MLMs:

1. A `[MASK]` news
2. `[SEP]` This topic is about `[MASK]`
3. Category : `[MASK]`
4. `[SEP]` The category of this news is `[MASK]`

The prompt templates used for GPT-2 are:

1. `[SEP]` This topic is about `[MASK]`
2. `[SEP]` The category of this text is `[MASK]`
3. `[SEP]` Category : `[MASK]`
4. `[SEP]` This is a news from `[MASK]`

The candidate answers used are the same as the class labels, namely, politics, business, sports, and technology for all the experiments except the larger candidate set ablation study. For that ablation, we use the following candidate answer set:

1. {politics, world, government, governance}
2. {sports, competition, games, tournament}
3. {business, corporation, enterprise, commerce}
4. {technology, science, electronics, computer}

BoolQ The prompt templates used for MLMs are:

1. Answer to the question is `[MASK]`
2. `[SEP]` `[MASK]`
3. I think `[MASK]`
4. `[SEP]` The answer is `[MASK]`

The prompt templates used for GPT-2 are the same as above except every template is appended to the end of the input. As in AG News, the candidate answers used are the same as the class labels, namely false and true, except when performing the larger candidate set experiment, in which case we use the following candidate answer set:

1. False: false, wrong, incorrect, invalid
2. True: true, correct, valid, accurate

SST-2 The prompt templates used for MLMs are:

1. Sentiment of the statement is `[MASK]` .
2. `[SEP]` `[MASK]`
3. This is a `[MASK]` statement
4. `[SEP]` The statement is `[MASK]`

Similar to AG News and BoolQ, we use the class labels (i.e., negative and positive) as the candidate answers.

DBPedia14 The prompt templates used for MLMs are:

1. Content on `[MASK]`
2. `[SEP]` This topic is about `[MASK]`
3. Category : `[MASK]`
4. `[SEP]` The content is about `[MASK]`

MRPC The prompt templates used for MLMs are:

1. The two sentences are `[MASK]`
2. `[SEP]` First sentence is `[MASK]` to second sentence
3. Two `[MASK]` sentences

SEP The two sentences have `[MASK]` meanings

D. Extended Experiments

D.1. Results on Additional Datasets and Attacks

Additional Attacks In the main paper, we evaluated our method on two popular word substitution attacks. These included the TextFooler and TextBugger attack strategies. They are word substitution attacks that replace words with "similar" neighboring words (where similarity is based on counterfitted GloVe embeddings). TextFooler greedily searches in a large set of neighbors (in the embedding space) for each word, so long as they satisfy some constraints on

	GPT2					
	BoolQ			AG News		
	Clean Acc	TextFooler	TextBugger	Clean Acc	TextFooler	TextBugger
MLP-FT	61.0 ± 2.1	20.2 ± 0.6	24.9 ± 1.4	93.7 ± 0.2	27.6 ± 1.2	58.2 ± 0.9
MLP-FT +Adv	60.5 ± 0.4	22.0 ± 1.1	31.8 ± 1.8	92.4 ± 0.3	39.6 ± 0.5	61.3 ± 0.7
MVP	72.5 ± 1.0	28.7 ± 1.6	38.3 ± 1.6	93.8 ± 0.3	31.4 ± 0.5	61.0 ± 0.8
MVP +Adv	71.8 ± 0.8	**30.1 ± 0.6**	**41.2 ± 0.8**	93.7 ± 0.3	**44.0 ± 0.2**	**64.4 ± 1.2**

Table 4: Adversarial Robustness results on BoolQ and AG News dataset using GPT-2 model. All experiments are run on 3 different seeds and the performance is reported over a fixed test set of size 1000. The best-performing robust accuracies are bolded and the second best robust accuracies are underlined.

embedding similarity and sentence quality. An additional constraint requires the substituted word to match the POS of the original word. TextBugger, on the other hand, restricts the search space to a small subset of neighboring words and only uses sentence quality as a constraint. To control the amount of change made by an attack, we limit the adversary to perturbing a maximum of 30% words in the AG News dataset and 10% in all other datasets. We do not modify any other constraints (such as the query budget) and run the attacks on 1000 examples from the test set. In the appendix, we further extend our evaluation on one character-level, and another word substitution attack. For character-level attack, we use the adversarial misspellings attack introduced by Pruthi et al. (2019), and we additionally evaluate the popular BertAttack (Li et al., 2020). Results on RoBERTa and BERT-base models are presented in Tables 5, 6 respectively.

MVP without adversarial training improves over MLP-FT by an average of 6% on BERT-Attack and 5% on Adversarial-Misspellings across 2 models and multiple datasets.

Additional Datasets We further extend our results on two diverse datasets—DBPedia14 (Zhang et al., 2015a), a 14-class news classification dataset, and MRPC (Dolan & Brockett, 2005), a paraphrase detection dataset. Results on these are presented for the MLP-FT and MVP training schemes for RoBERTa-base model in Table 5.

The experiments provide additional evidence to support our findings about the adversarial robustness conferred by model-tuning via prompts (MVP) as opposed to the conventional approach of MLP-FT. Without adversarial training, MVP improves over MLP-FT by an average of 6% on the MRPC dataset across 4 different attacks. Results on the DBPedia dataset also show consistent improvements of MVP over MLP-FT . In particular, we find that MVP improves on average (across 4 different attacks) by 10% over MLP-FT, and MVP + adv improves by 16% over the adversarial training counterpart of MLP-FT. In a setting where the number of labels is many, we in fact see a larger relative gain by using MVP over the conventional approach of MLP-FT.

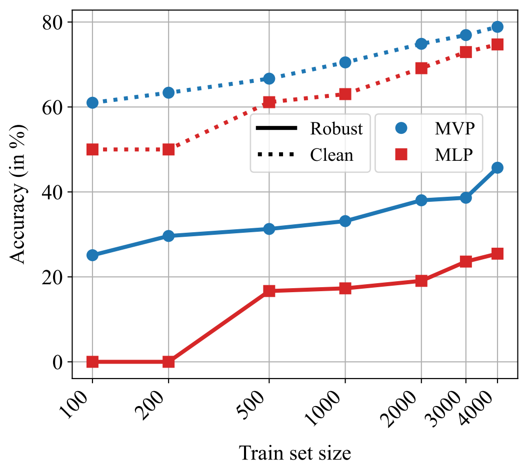
D.2. Results on Causal Language Models

Causal Language Models have not been traditionally fine-tuned for downstream classification tasks. This is evident also from the exclusion of fine-tuning results in the original GPT-2 paper (Radford et al., 2019). In this work, we try to evaluate the clean and adversarial robustness of GPT-2 models, when adapted to downstream tasks. To implement MVP, we use the Causal Language Modeling (CLM) head to get the next word prediction logits. Since we are using the CLM head, it is imperative that the prompt templates are appended at the back and have the [MASK] as the last token.

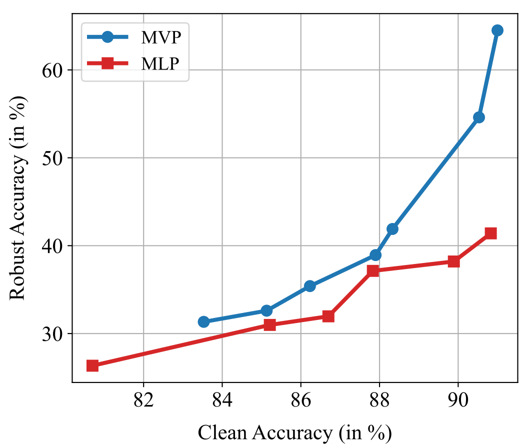
We find that on the BoolQ dataset MLP-FT achieves a robust accuracy of 20.2% and MVP achieves a robust accuracy of 28.7% (Table 4), which is a large improvement. Similar to our findings in the main paper, 1-step adversarial training on MVP (MVP + Adv) yields a robust accuracy of 30.1% which is a massive improvement over MLP-FT and MLP-FT + Adv which obtains a robust accuracy of 22.0%. Interestingly, we also notice that for MLP-FT and MLP-FT + Adv, it is difficult to achieve a good clean generalization performance whereas MVP and MVP + Adv perform much better on the clean test set. These observations are in line with the results in our main paper. On the AG News dataset, MVP performs significantly better than MLP-FT and MVP + Adv performs better than MLP-FT + Adv. These results show that MVP is not only a good way of finetuning BERT-like MLMs but can also improve Causal Language Models both in terms of clean accuracy and robustness to adversarial perturbations.

D.3. Additional Sample Efficiency and Effective Robustness

We demonstrate the sample efficiency of MVP on the BoolQ dataset (Figure 3a) in addition to the discussion about AG News in §4.1. Interestingly we find that MLP-FT is unable



(a) Clean and adversarial accuracies of RoBERTa-base model on BoolQ dataset for varying amounts of training data.



(b) Clean vs adversarial performance of RoBERTa base model for the AG News dataset. We find that models tuned via prompts (MVP) yield more robust models compared to fine-tuning MLP heads for the same clean accuracy.

Figure 3: (a) Models trained with MVP are significantly more sample efficient as compared to those with MLP-FT . (b) We find that models tuned via prompts (MVP) yield more robust models compared to fine-tuning MLP heads for the same clean accuracy (see §4.1 for details).

to achieve better accuracy compared to even random classifiers with 200 examples but MVP performs much better in the low data regime ($<$ 200 examples). We also provide more evidence on the effective robustness of MVP by presenting the effective robustness results on AG News (Figure 3b). Even for AG News, we notice that the curve is much steeper for MVP than MLP-FT.

E. Hyperparameter Details

Attack Hyperparameters TextFooler and TextBugger use a word substitution attack that searches for viable substitutions of a word from a set of synonyms. We restrict the size of the synonym set to 50 for TextFooler which is the default value used by Jin et al. (2020) and to 5 which is the default value used by Li et al. (2018). Both TextFooler and TextBugger use a Universal Sentence Encoder (USE), that poses a semantic similarity constraint on the perturbed sentence. We use the default value of 0.84 as the minimum semantic similarity. Another important attack parameter is the maximum percentage of modified words (ρ_{max}). As discussed in (Li et al., 2021), we use $\rho_{\text{max}} = 0.3$ for AG News and use $\rho_{\text{max}} = 0.1$ for BoolQ and SST2 in all our experiments.

Training Hyperparameters & Model Selection We train all models including the baselines with patience of 10 epochs, for a maximum of 20 epochs, and choose the best model based on validation accuracy. For the datasets that do not contain a publicly available validation set, we set aside 10% of the training set for validation. In the case of baseline defenses that use adversarial training, we perform model selection based on adversarial accuracy rather than clean accuracy. We use a candidate answer set containing only the class label names and we average over 4 prompt templates in all the MVP models. We use a batch size of 32 for MLP-FT and a batch size of 8 for MVP models. The learning rate is set as 1e$-$5 for all the models. We use the AdamW optimizer along with the default linear scheduler (Wolf et al., 2020). In all the MVP + Adv and MLP-FT + Adv models, we use a use 1-step adversarial training with max ℓ_2 norm of 1.0. For the state-of-the-art baselines, we use the same hyperparameters as prescribed by the original papers.

F. Human Study

Despite the improvements brought to adversarial robustness by our proposed modification (MVP + Adv), we note that there is still a significant drop in robust accuracy as opposed to the clean accuracy of the model. We conduct a human study in order to (i) assess the viability of adversarial attacks, and (ii) estimate human performance against adversarial attacks. More specifically, we provide machine learning graduate students 250 input examples and ask the following questions:

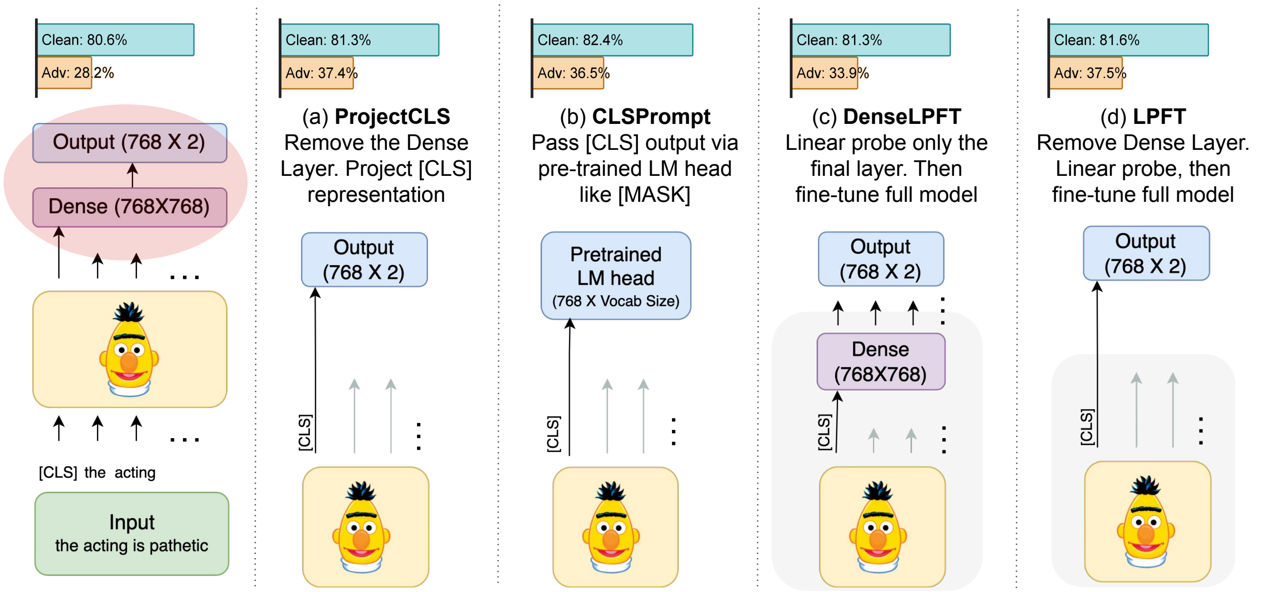


Figure 4: Various model tuning strategies for RoBERTa model trained on the BoolQ dataset. The corresponding clean and robust accuracies (under TextFooler attack) are also shown above each model paradigm. The left-most diagram shows the standard fine-tuning paradigm of `MLP-FT` , and each subsequent column modifies the architecture, helping us confirm the hypothesis of that randomly initialized parameters are a cause of vulnerability.

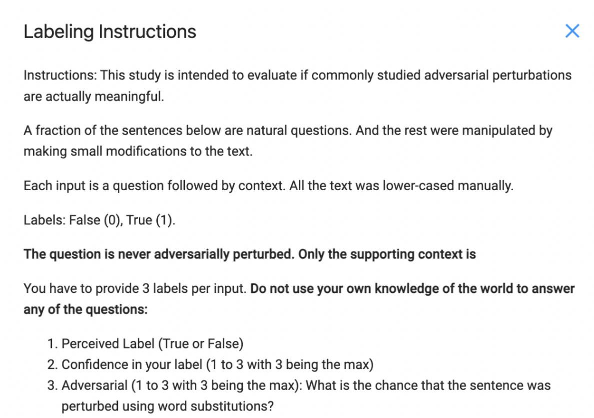
1. What is the perceived label of the sentence? (Answer options: True or False)

2. On a scale of 1 to 3, what is their confidence about this label?

3. Was this sentence adversarially manipulated? (Answer options: Yes, Unsure, or No)

We use the BoolQ dataset and strictly instruct our annotators to not use any external knowledge but the only context of the given passage. We use samples that were successfully attacked by TextFooler for `MVP` + Adv model with a RoBERTa backbone. As a control for the study, 33% of all sentences are unperturbed sentences from the original dataset. The underlying model achieves a clean accuracy of 81.7% and a robust accuracy of 54.0%.

First, we find that humans achieved 11% lower accuracy on adversarial examples as compared to clean examples (85% $\rightarrow$ 74%) with average confidence on the label of perturbed examples being 15% lower (90% $\rightarrow$ 75%) (Table 7). Next, we also discover that human annotators suspect 29% of adversarial examples to be perturbed as opposed to only 6% of clean examples. Through this study, we also find that in 47% of the cases, the input is either manipulated so significantly that it is easily detectable or the original label is not preserved, signifying that `MVP` may be more robust than what statistics suggest in §4. Further details are available in Appendix F.1.


Labeling Instructions

Instructions: This study is intended to evaluate if commonly studied adversarial perturbations are actually meaningful.

A fraction of the sentences below are natural questions. And the rest were manipulated by making small modifications to the text.

Each input is a question followed by context. All the text was lower-cased manually.

Labels: False (0), True (1).

The question is never adversarially perturbed. Only the supporting context is

You have to provide 3 labels per input. **Do not use your own knowledge of the world to answer any of the questions:**

1. Perceived Label (True or False)
2. Confidence in your label (1 to 3 with 3 being the max)
3. Adversarial (1 to 3 with 3 being the max): What is the chance that the sentence was perturbed using word substitutions?

Figure 5: A snapshot of the instructions for completing our study.

F.1. Details of Interface

We present a snapshot of our interface that provides detailed instructions for our users (Figure 5). We provide a detailed overview of the questions asked in the user study. Annotators were provided with a boolean question and an accompanying context to answer the question and asked were asked to annotate the following:

1. What should be the answer to the question? (only use the context) Given the boolean question and the context, we ask the annotators whether the answer to the question is True or False. We also request the annotators only use the given context and refrain from using any external knowledge.

2. How confident are you about the label above? Once the annotator has answered question 1, we ask them to rate how confident they feel about the label they assigned to the input. The options provided are "Uncertain", "Somewhat Certain" and "Certain". Based on their response we assign a confidence of 1, if the annotator was certain, assign 0.5 if the annotator was somewhat certain, and assign 0 if the annotator was uncertain to calculate the average confidence.

3. Do you think that the sentence is adversarially perturbed? (using word substitutions) Do not use your own knowledge of the world to answer this question. We also ask the annotators, if the input was adversarially perturbed. The options provided to the user are "No", "Unsure" and "Yes".

The annotators helped us annotate 250 such examples out of which 167 were adversarially perturbed and 83 were clean. An overview of the responses from this study is presented in Table 7, and the key takeaways are discussed in Section F.

			SST2		
	Clean Acc	TextFooler	TextBugger	BertAttack	Misspellings
`MLP-FT`	93.6 ±0.4	40.2 ±0.9	65.4 ±0.3	70.3 ±0.9	45.2±1.1
`MLP-FT` + Adv	93.6 ±0.6	44.0 ±1.2	68.5 ±1.5	74.3±0.8	49.3 ±0.3
Free LB++	94.0 ±0.1	43.4 ±1.0	67.2 ±0.6	76.2 ±0.6	50.4±1.1
MADA	93.8 ±0.4	41.8 ±0.5	66.1 ±0.2	74.2 ±0.2	45.4 ±0.4
InfoBert	94.0 ±0.4	43.6 ±0.5	66.6 ±1.8	76.1 ±0.6	47.1 ±0.4
`MVP`	93.9 ±0.7	46.9 ±0.5	69.8 ±0.5	78.1 ±0.9	50.5 ±0.7
`MVP` + Adv	93.8 ±0.1	53.8 ±0.7	71.7 ±0.8	81.7 ±0.7	54.9 ±1.3
			AG News		
	Clean Acc	TextFooler	TextBugger	BertAttack	Misspellings
`MLP-FT`	94.5 ±0.4	42.9 ±0.7	61.8 ±0.3	79.1 ±1.3	76.8 ±1.3
`MLP-FT` + Adv	94.4 ±0.6	47.7 ±0.5	65.6 ±0.8	81.1±1.0	78.6 ±0.8
Free LB++	94.4 ±0.7	46.9 ±1.6	65.5 ±1.0	81.4 ±0.9	80.1 ±1.3
MADA	94.1 ±0.6	44.3 ±1.4	62.9 ±0.5	80.4 ±0.2	77.1 ±0.4
InfoBert	94.5 ±0.9	48.0 ±2.2	65.6 ±1.2	82.4±1.2	80.4±1.4
`MVP`	94.3 ±0.2	51.5 ±2.1	68.7 ±0.7	85.3 ±1.3	82.7 ±0.7
`MVP` + Adv	94.4 ±0.8	62.7 ±2.4	75.3 ±1.6	88.2 ±0.9	86.6 ±0.6
			BoolQ		
	Clean Acc	TextFooler	TextBugger	BertAttack	Misspellings
`MLP-FT`	80.6 ±1.5	28.2 ±1.7	38.3 ±1.0	54.3 ±1.8	55.2±1.2
`MLP-FT` + Adv	78.9 ±1.2	39.0 ±0.7	44.4 ±1.2	57.4 ±0.9	57.3 ±1.3
Free LB++	80.6 ±0.4	37.2 ±1.4	43.2 ±1.0	58.3 ±0.5	58.0±1.1
MADA	79.2 ±0.9	32.0 ±0.3	41.1 ±0.2	57.9 ±0.2	56.4 ±0.2
InfoBert	81.5 ±0.7	38.0 ±1.3	42.4 ±0.9	59.1 ±0.5	59.1 ±1.4
`MVP`	82.0 ±0.6	42.9 ±0.5	49.8 ±1.6	64.1 ±0.7	60.1 ±1.6
`MVP` + Adv	81.1 ±0.6	52.2 ±1.6	56.4 ±1.6	68.2 ±0.6	64.3 ±0.3
			DBPedia		
	Clean Acc	TextFooler	TextBugger	BertAttack	Misspellings
`MLP-FT`	97.3±0.7	43.8±1.5	68.7±0.9	72.4±1.2	65.7±1.3
`MLP-FT` + Adv	97.2±0.4	56.1±0.2	76.4±0.3	78.3±0.6	72.2±0.7
`MVP`	97.0±0.5	57.2 ±1.0	77.2±0.5	80.6±0.7	74.3±0.7
`MVP` + Adv	97.3±0.9	82.7±0.4	90.3±0.2	88.5 ±1.8	86.4±0.3
			MRPC		
	Clean Acc	TextFooler	TextBugger	BertAttack	Misspellings
`MLP-FT`	87.9±0.6	41.5±1.2	50.2±1.0	61.1±1.1	51.7±1.0
`MLP-FT` + Adv	87.2±0.4	42.1±0.3	53.4±0.7	64.1±0.1	54.2±0.4
`MVP`	88.4±0.4	44.8 ±0.1	56.6±0.1	68.8±0.5	57.3±0.9
`MVP` + Adv	87.1±1.2	46.6±1.2	60.7±0.4	72.1 ±0.9	65.8 ±0.3

Table 5: Adversarial performance of RoBERTa-base model on 5 different datasets. All accuracy values are reported for a fixed test set of size 1000 and are averaged over 3 different seeds. The highest accuracies are bolded, and the second-best are underlined. `MVP` is the most robust, and preserves (or improves) the clean accuracy.

	SST2				
	Clean Acc	TextFooler	TextBugger	Bertattack	Misspellings
MLP-FT	91.9 ±0.2	38.3 ±1.0	60.4 ±0.4	68.7±0.5	39.2 ±0.4
MLP-FT + Adv	90.9 ±0.3	42.8 ±1.2	62.3 ±0.5	70.1±0.8	42.4 ±0.4
Free LB++	92.1 ±0.8	42.2 ±1.0	63.0 ±0.7	72.0±0.9	43.4 ±0.4
MADA	92.1 ±0.9	41.7 ±0.5	60.9 ±0.4	70.3±0.7	40.2 ±0.7
InfoBert	91.7 ±0.6	43.1 ±0.8	64.6 ±0.6	72.8±0.6	43.1 ±0.7
MVP	91.7 ±0.4	44.6 ±0.7	65.1 ±0.1	75.9±0.7	45.6 ±1.1
MVP + Adv	91.8 ±0.7	47.6 ±0.5	67.7 ±0.3	78.9±0.8	49.2 ±0.9
	AG News				
	Clean Acc	TextFooler	TextBugger	Bertattack	Misspellings
MLP-FT	93.7 ±0.4	37.5 ±0.7	58.9 ±0.6	78.1±1.2	76.8 ±0.8
MLP-FT + Adv	93.2 ±0.2	44.3 ±1.0	64.1 ±0.2	80.1±0.2	78.5 ±0.2
Free LB++	93.4 ±0.2	43.5 ±0.2	63.4 ±0.8	80.9±0.1	79.5 ±0.7
MADA	92.8 ±0.5	41.8 ±0.9	62.6 ±1.0	79.6±0.6	76.9 ±1.3
InfoBert	93.8 ±0.3	44.0 ±1.6	64.1 ±0.8	80.7±0.6	79.6 ±0.7
MVP	93.7 ±0.5	46.3 ±1.2	66.0 ±0.4	82.1±0.7	81.5 ±0.4
MVP + Adv	94.0 ±0.6	53.7 ±0.1	69.2 ±1.3	83.4±0.4	84.3 ±0.3
	BoolQ				
	Clean Acc	TextFooler	TextBugger	Bertattack	Misspellings
MLP-FT	71.1 ±1.3	21.8 ±4.4	36.8 ±3.0	55.7±1.2	55.1 ±1.0
MLP-FT + Adv	71.0 ±0.9	29.8 ±0.8	42.8 ±1.3	57.8±0.7	58.1 ±0.3
Free LB++	70.7 ±0.2	29.5 ±0.6	42.8 ±0.6	58.2±0.9	59.4 ±0.7
MADA	71.1 ±0.9	25.4 ±0.8	41.6 ±0.6	57.8±0.6	56.2 ±0.7
InfoBert	71.8 ±0.6	29.9 ±0.2	42.6 ±0.6	58.9±0.8	59.1 ±0.6
MVP	71.4 ±1.0	31.1 ±1.3	44.4 ±2.8	60.1±0.6	60.1 ±1.0
MVP + Adv	71.3 ±0.7	43.1 ±0.7	49.9 ±0.9	63.2±0.7	63.2 ±0.8

Table 6: Adversarial performance of BERT-base model on 5 different datasets. All accuracy values are reported for a fixed test set of size 1000 and are averaged over 3 different seeds. The highest accuracies are bolded, and the second-best are underlined. MVP is the most robust, and preserves (or improves) the clean accuracy.

		Adversarial Examples	Original Examples
Q1. Annotator Accuracy		74%	85%
Q2. Annotator Confidence		75%	90%
Q3. Perturbed?	No	54%	82%
	Unsure	17%	12%
	Yes	29%	06%

Table 7: Summary of the responses from the user study. The total number of presented examples is 250, out of which 83 are clean and 167 are adversarially manipulated.