Regression Bugs Are In Your Model!
Measuring, Reducing and Analyzing Regressions In NLP Model Updates

Yuqing Xie[*1]**, Yi-An Lai**[2]**, Yuanjun Xiong**[2]**, Yi Zhang**[2]**, Stefano Soatto**[2]

[1]University of Waterloo
[2]Amazon AWS AI
`yuqing.xie@uwaterloo.ca`
`{yianl,yuanjx,yizhngn,soattos}@amazon.com`

Abstract

Behavior of deep neural networks can be inconsistent between different versions. Regressions[1] during model update are a common cause of concern that often over-weigh the benefits in accuracy or efficiency gain. This work focuses on quantifying, reducing and analyzing regression errors in the NLP model updates. Using *negative flip rate* as regression measure, we show that regression has a prevalent presence across tasks in the GLUE benchmark. We formulate the regression-free model updates into a constrained optimization problem, and further reduce it into a relaxed form which can be approximately optimized through knowledge distillation training method. We empirically analyze how model ensemble reduces regression. Finally, we conduct CHECKLIST behavioral testing to understand the distribution of regressions across linguistic phenomena, and the efficacy of ensemble and distillation methods.

1 Introduction

Regression-free model update is a desirable system property which guarantees interoperability of a new system with a legacy version, also known as backward compatibility. Regression occurs when the newly updated system stops functioning as intended.

As advances in deep learning spark industrial applications in AI areas such as natural language processing, the long-term maintenance of such systems is becoming ever more challenging. While models with complex neural architectures and huge parameter space continue to reach higher accuracy, the lack of interpretability and functional decomposibility in these models make it infeasible to apply traditional software regression testing

---

* Work done while at Amazon AWS AI.

[1] Here *regression* refers to bugs in software testing instead of the statistical estimation method.

Figure 1: Prediction flip scenarios on NLP classification tasks when updating from old to new models. Examples from paraphrase classification task.

methods such as unit tests. As result, validating and mitigating regressions during model update is often a long and painful engineering process, which often over-shadows the benefits of a new model.

The model regression issue in deep learning first comes into sight in Shen et al. (2020), where they inspect compatible representation learning for image retrieval. Yan et al. (2020) proposed the positive-congruent training (PCT) for image classification that minimizes prediction errors and model regression at the same time. To our best knowledge, the model update regression has not been studied on NLP tasks.

Following Yan et al. (2020), in this work we measure the model update regression in NLP by *negative flips*. In Figure 1, we demonstrate prediction flip scenarios. Negative flips are shown in the upper-right quadrant where the old model makes correct predictions and the new model predictions are wrong. As we will show in Section 2, regression are prevalent in NLU model updates even with the slightest changes in the new model training process.

To develop a model with minimum regression, we first formulate the learning task into a constrained optimization problem by taking the regression-free conditions as constraints. We apply the Lagrangian relaxation to bring the regression-free constraint into the optimization objective as an

additional penalty loss, and provide approximate solution via knowledge distillation. Yan et al. (2020) also observed that model ensemble can also reduce negative flips without explicit input from the old model. We evaluate both distillation and ensemble based methods on a diverse set of NLP tasks.

To further understand how the above methods contribute to reducing it, we utilize CHECKLIST (Ribeiro et al., 2020) to quantify linguistic behavioral changes before and after applying proposed methods. We find that regressions are prevalent in NLP tasks, and their distribution correlates with different linguistic phenomena.

Our main contributions are as follows:

- We provide empirical evidence to show that the model update regression occurs across text classification tasks in NLP;

- We formulate the regression-free model updates into a constrained optimization problem, and further reduce into a relaxed form which can be approximately optimized through knowledge distillation training method;

- We also explore the model ensemble as another method to reduce regression, and analyzed its efficacy;

- We analyze the source of the regressions in NLP tasks through linguistic behavioural testing, compare reduction in both distillation and ensemble methods.

2 Measuring Regression in NLP Model Update

In this section, we first formulate the measure of model update regression on classification tasks. Then we benchmark on GLUE tasks (Wang et al., 2018) and show that there is a prevalent presence of regression when updating models in NLP.

2.1 Regression Measurement on Classification Tasks

Similar to software regression testing, we need to collect a group of test cases when measuring regression. We start from a regression set: $\mathcal{D}_{reg} = \{x_i, y_i\}_{i=1}^N$, $y_i \in \{l_1, l_2, ..., l_C\}$, where l_i is the i-th label and C is the number of classes. In practice, we can use the development set or compile a collection of critical use cases as $\mathcal{D}_{reg}$.

In a classification task, given a input x_i, a neural network model f, parameterized by ϕ,

approximates the posterior probabilistic distribution $p(y_i|x_i)$ over all possible labels: $\vec{f}_\phi(x_i) = (p_\phi(y = l_1|x_i), ..., p_\phi(y = l_C|x_i))^\top$. To simplify, we denote the final prediction of a model to be $f_\phi(x) = arg\max_{l_j} p_\phi(l_j|x)$.

The regression $\mathcal{R}_{NF}$ between two models $f_{\phi_{old}}$ and $f_{\phi_{new}}$ on $\mathcal{D}_{reg}$ can be defined as the portion of negative flip cases:

$$\mathcal{R}_{NF}(\mathcal{D}_{reg}, \vec{f}_{\phi_{old}}, \vec{f}_{\phi_{new}})$$
$$= \frac{|\{x|f_{\phi_{old}} = y, f_{\phi_{new}} \neq y\}|}{|\mathcal{D}_{reg}|}.$$

We use negative flip $\mathcal{R}_{NF}$ as our regression measurement for classification tasks. Lower $\mathcal{R}_{NF}$ for a new models means better compatibility with the old model.

2.2 Benchmark Severity of Regression

The success of Transformer (Vaswani et al., 2017) and BERT (Devlin et al., 2019) have made pre-training then fine-tuning a standard paradigm in NLP systems. When updating these systems, differences can come from various aspects:

- Changes in the fine-tuning hyperparameters (e.g. random seed, learning rate schedule, epoch, etc.)

- Changes in model size or architecture (e.g. from $BERT_{base}$ to $BERT_{large}$)

- Changes in pre-training procedure or objective (e.g. BERT to ROBERTA (Liu et al., 2020), to $BERT_{whole-word-masking}$ or to ELECTRA (Clark et al., 2020))

- Changes in pre-trained model architecture (e.g. BERT to ALBERT (Lan et al., 2020))

While accuracy or efficient improvements are strong motivations for these model updates, they could also introduce behavioral incongruence when compared to the previous model. To benchmark the severity of regression, we apply a general setup: Fine-tune various pre-trained language models (LM) on GLUE and calculate $\mathcal{R}_{NF}$ when updating from $BERT_{base}$ to other LMs . We use dev sets as $\mathcal{D}_{reg}$. Results in Table 1 show that:

1. *Model update regression is prevalent on NLU tasks.* A minimum of 1.98% $\mathcal{R}_{NF}$ is observed across diverse classification tasks and model update scenarios, while the average accuracy gain is only 1.4%.

2. *Minor changes such as random seeds can introduce significant regression.* Shown in

		CoLA	MRPC	QQP	MNLI-m
Train size		8.6k	3.7k	360k	390k
Dev size		1k	0.4k	40k	9.8k
Old: BERT$_{base}$	Acc	82.84%	86.03%	90.76%	83.82%
$\rightarrow$ **BERT$_{base}$**	Acc	83.80%(+0.96%)	86.03%(-0.00%)	90.56%(-0.20%)	83.55%(-0.27%)
	$\mathcal{R}_{NF}$	$\hookleftarrow$3.36%	$\hookleftarrow$ 4.17%	$\hookleftarrow$2.32%	$\hookleftarrow$ 3.56%
$\rightarrow$ **BERT$_{large}$**	Acc	85.43%(+2.59%)	87.75%(+1.72%)	91.11%(+0.35%)	86.10%(+1.97%)
	$\mathcal{R}_{NF}$	$\hookleftarrow$ 3.16%	$\hookleftarrow$ 5.88%	$\hookleftarrow$2.82%	$\hookleftarrow$ 3.95%
$\rightarrow$ **ROBERTA$_{base}$**	Acc	84.85%(+2.01%)	89.22%(+3.19%)	91.25%(+0.52%)	87.58% (+4.42%)
	$\mathcal{R}_{NF}$	$\hookleftarrow$4.67%	$\hookleftarrow$ 4.66%	$\hookleftarrow$1.98%	$\hookleftarrow$ 2.64%
$\rightarrow$ **ELECTRA$_{base}$**	Acc	85.81%(+2.97%)	86.03%(+0.00%)	91.35%(+0.59%)	88.87%(5.05%)
	$\mathcal{R}_{NF}$	$\hookleftarrow$ 5.18%	$\hookleftarrow$ 5.39%	$\hookleftarrow$ 3.20%	$\hookleftarrow$3.50%
$\rightarrow$**ALBERT$_{base}$**	Acc	76.51%(-6.33%)	86.27%(+0.24%)	90.73%(-0.03%)	85.26%(+1.44%)
	$\mathcal{R}_{NF}$	$\hookleftarrow$10.74%	$\hookleftarrow$ 6.86%	$\hookleftarrow$ 3.78%	$\hookleftarrow$ 5.22%

Table 1: Regression measurement when performing different model updates on GLUE benchmark. Old: BERT$_{base}$ is our base version of model and we update it to other new models. We also show the accuracy gain in the parenthesis. All numbers are 5-seed average.[2]

$\rightarrow$BERT$_{base}$, even when we only alter the initialization random seed, this can lead to up to 3.56% negative flip.

3. *Negative flip rates are often much higher than the accuracy gains.* When updating to BERT$_{large}$ on QQP, $\mathcal{R}_{NF}$ is about 8x the accuracy gain. This implies reducing error rate alone does not ensure the decrease in regression.

4. *Pre-training objective or architecture updates often lead to higher regressions than those caused by model size or random seeds.* The regressions are higher when updating to ALBERTA, compared with updating to a larger model BERT$_{large}$ or a different random seed. This implies systematic regression could be introduced if the backbone models are different.

3 Reducing Regression in Model Update

In this section, we first formulate regression-free model update as a constrained optimization problem, then further reducing it to a joint optimization objective combining the training loss on the original task and a distillation loss with respect to the old model's behavior.

Unlike typical optimizations in neural model training where we minimizes a loss function on a training set, the regression-free model update requires the model to learn the target task as well as comply with conditions posed by the old model.

We can cast the regression-free model update as a constrained optimization problem by writing down the classification loss as the optimization objective and the regression-free conditions as constraints:

$$\min_{\phi_{new}} \sum_{x\in\mathcal{D}_{train}} \mathcal{L}_{CE}(x, \phi_{new}) \tag{1}$$
$$s.t.\ \mathcal{R}_{NF}(f_{\phi_{old}}, f_{\phi_{new}}, \mathcal{D}_{reg}) = 0.$$

where $\mathcal{D}_{train}, \mathcal{D}_{reg}$ represent the training and regression sets, respectively.

The constraint in Equation 1 asks for zero regression on $\mathcal{D}_{reg}$. It would be difficult to ensure the constraint is satisfied along the model training. We instead relax the hard constraint into a soft inequality condition that allows the regression measure to be less than a constant $\mathcal{C}$:

$$\min_{\phi_{new}} \sum_{x\in\mathcal{D}_{train}} \mathcal{L}_{CE}(x, \phi_{new}) \tag{2}$$
$$s.t.\ \mathcal{C} - \mathcal{R}_{NF}(f_{\phi_{old}}, f_{\phi_{new}}, \mathcal{D}_{reg}) \geq 0.$$

Training a model directly with the regression-free constraint still remains difficult in that signals from old predictions are sparse and $\mathcal{R}_{NF}$ is non-differentiable. Here, we propose two proxies of $\mathcal{R}_{NF}$ to measure regression in continuous space.

Proxy from Prediction Probabilities. We use the KL divergence between the predicted probabilities of both models as one soft regression measure:

$$\mathcal{R}_{\text{KL-div}}(f_{\phi_{old}}, f_{\phi_{new}}, \mathcal{D}_{reg})$$
$$= \sum_{x\in\mathcal{D}_{reg}} D_{KL}(p_{\phi_{old}}(y|x)||p_{\phi_{new}}(y|x)). \tag{3}$$

---

[2]Full results on GLUE can be found in Appendix A

Proxy from Deep Representations. We can also use the l_2 distance between models' sentence representations, e.g. `[CLS]` embedding in BERT as another soft regression measure:

$$
\begin{aligned}
&\mathcal{R}_{l_2}(f_{\phi_{old}}, f_{\phi_{new}}, \mathcal{D}_{reg}) \\
&= \sum_{x \in \mathcal{D}_{reg}} l_2(\vec{f}_{\phi_{old}}(x), \vec{f}_{\phi_{new}}(x)).
\end{aligned} \quad (4)
$$

A linear projection is used to align the representations if they initially lie in different spaces.

Reduce to Knowledge Distillation. Finally, we apply the Lagrangian relaxation to bring the regression-free constraint into the optimization objective as an additional penalty loss:

$$
\begin{aligned}
\min_{\phi_{new}} &\sum_{x \in \mathcal{D}_{train}} \mathcal{L}_{CE}(x, \phi_{new}) \\
&- \alpha * (\mathcal{C} - \mathcal{R}_{soft}(f_{\phi_{old}}, f_{\phi_{new}}, \mathcal{D}_{reg})),
\end{aligned} \quad (5)
$$

where α is a positive penalty scaling parameter and $\mathcal{R}_{soft}$ can be chosen from $\mathcal{R}_{\text{KL-div}}$ or $\mathcal{R}_{l_2}$. Then, the above optimization problem can be cast into a joint learning of the original target task and knowledge distillation from the old model. The distillation loss acts as a surrogate of the model update regression measure. The joint learning process minimizes this term as an approximation of minimizing the overall model update regression.

4 Experiments

4.1 Implementation Details

Since we usually update models from elementary ones to improved ones, in the experiments we take origin BERT$_{base}$ (12-layer, 768-hidden, 12-heads, 110M parameters) (Devlin et al., 2019) as the old model's backbone and update it to a homogeneous model, e.g. BERT$_{base}$ with different fine-tuning random seeds or parameters, or a heterogeneous models with improvements such as BERT$_{large}$ (24-layer, 1024-hidden, 16-heads, 340M parameters). We fine-tune the pre-trained LMs without any constraint as our baselines. We use the GLUE datasets to benchmark the effectiveness of proposed techniques. Details of each GLUE task can be found in Appendix D. For investigative experiments, we use the Microsoft Research Paraphrase Corpus (MRPC) (Dolan and Brockett, 2005), a paraphrase identification dataset that aims to classify whether two sentences are the paraphrase of each other. Pre-trained

model artifacts and the GLUE dataset processing procedures are brought from Hugging Face[3] and experiments are done in PyTorch (Paszke et al., 2019) with Tesla V100 GPUs. Cross-entropy is used for fine-tuning on target tasks with batch size 16 for 4 to 6 epochs. The learning rate is searched among $2e^{-5}$, $3e^{-5}$ and $5e^{-5}$.

During joint training of classification and knowledge distillation, we take the fine-tuned old models as the teacher, and distill with batch size 16 for 6 to 8 epochs. We set $\mathcal{D}_{reg} = \mathcal{D}_{train}$ when training models with the constraint and use $\mathcal{D}_{reg} = \mathcal{D}_{dev}$ for reporting results. To encourage constraint satisfaction and reduce regression, we only include the distillation penalty into our loss on the examples where the current model makes negative flips.

4.2 Ensemble

Yan et al. (2020) reported an intriguing finding on image classification tasks that model ensemble can reduce model update regressions without explicit regularization from the old model. This was attributed to the reduction of variance in ensemble model predictions, making it less prone to overfitting and indirectly reducing regressions. Here we include model ensemble as an alternative approach to reduce regression, with further analysis on how ensemble reduces regression in Section 5.1.

4.3 Main Results

Table 2 shows the efficacy of distillation method and model ensemble on reducing NLP classification task model update regressions. On average, the distillation method reduces $\mathcal{R}_{NF}$ by 30.6% and 36.3% while the ensemble method reduces $\mathcal{R}_{NF}$ by 55.9% and 20.6% when updating to BERT$_{base}$ and to BERT$_{large}$, respectively. Both distillation and ensemble methods can significantly bring down negative flips across GLUE tasks compared with the baselines. The ensemble seems to work better when the old and new models share the same underlying pre-trained LM. In the update BERT$_{base} \rightarrow$ BERT$_{base}$, the ensemble method outperforms the distillation on reducing the regression. On the other hand, the distillation method seems to be more effective on reducing regression under the heterogeneous model update setting. In the update BERT$_{base} \rightarrow$ BERT$_{large}$, distillation reduce more regression, with especially

---

[3] https://huggingface.co

		CoLA	SST-2	MRPC	QQP	MNLI-m	QNLI	RTE	Average
Train size		8.6k	67k	3.7k	360k	390k	100k	2.5k	
Dev size		1k	0.9k	0.4k	40k	9.8k	5.5k	0.3k	
Old: BERT$_{base}$	Acc	82.26%	91.17%	86.03%	90.76%	83.82%	91.07%	67.15%	84.61%
→BERT$_{base}$ - Baseline	Acc	82.93%	91.63%	86.03%	90.56%	83.55%	90.65%	63.18%	84.08%
	$\mathcal{R}_{NF}$	4.41%	1.95%	4.17%	2.32%	3.56%	2.35%	11.43%	4.31%
→BERT$_{base}$ - Distillation	Acc	**84.47%**	**92.09%**	**87.01%**	**91.14%**	83.77%	91.16%	68.95%	**85.81%**
	$\mathcal{R}_{NF}$	**1.92%**	**0.80%**	1.72%	1.69%	4.32%	2.47%	8.30%	2.99%
→BERT$_{base}$ - Ensemble	Acc	82.17%	91.63%	86.03%	91.06%	**84.35%**	**91.62%**	**70.76%**	85.37%
	$\mathcal{R}_{NF}$	2.59%	0.92%	**1.23%**	**1.18%**	**1.66%**	**1.06%**	**4.69%**	**1.90%**
→BERT$_{large}$ - Baseline	Acc	85.62%	92.89%	87.75%	91.11%	86.10%	92.53%	66.43%	86.06%
	$\mathcal{R}_{NF}$	2.68%	1.72%	5.88%	2.82%	3.95%	2.64%	12.27%	4.57%
→BERT$_{large}$ - Distillation	Acc	**85.62%**	92.89 %	88.73%	91.50%	86.73 %	92.15%	**73.65%**	**87.33%**
	$\mathcal{R}_{NF}$	2.49%	**1.26%**	2.45%	2.46%	3.76%	2.54 %	**5.42%**	**2.91%**
→BERT$_{large}$ - Ensemble	Acc	84.95%	**93.12%**	**89.46%**	**91.66%**	**87.05%**	**93.08%**	67.87%	86.74%
	$\mathcal{R}_{NF}$	2.78%	1.61%	2.45%	**2.20%**	**3.24%**	**2.27%**	10.83%	3.63%

Table 2: Results of fine-tuning with distillation and ensemble on GLUE benchmark. Baseline denotes directly fine-tuning new pre-trained models on target tasks. We show the distillation results with $\mathcal{R}_{\text{KL-div}}$, and the ensemble results with 5 model majority vote. Due to page limitation, we only show the matched results on MNLI (Williams et al., 2018).

large reductions on small datasets such as CoLA and SST-2. We hypothesize that it's because the ensemble focuses on reducing the variance in model predictions, while distillation enables the explicit alignment in either probability distribution or representation space between the old and the new model. When the new model is very different from the old one, it can implicitly align new model's behavior with the old one.

4.4 Variants in Distillation Objective

As introduced in Section 3, we can have several variants of distillation loss to be used to constrain new model training on the old model. We explore and benchmark the following variants on the MRPC task:

- **Distillation - $\mathcal{R}_{\text{KL-div}}$, Logits** calculates the distillation loss as the KL divergence between the two Bernoulli distributions set by the old and new model prediction probabilities;

- **Distillation - $\mathcal{R}_{l_2}$, [CLS]** uses the `[CLS]` token embedding from the final layer as sentence representations and calculates the distillation loss as the Euclidean distance between the two vectors;

- **Distillation - $\mathcal{R}_{l_2}$, All [CLS]** also calculates the Euclidean distance between the old and new sentence representation vectors, but with concatenated `[CLS]` token embeddings from all layers instead of the final layer.

Pre-trained models could have different layers. For BERT$_{base}$ →BERT$_{large}$ in the **All [CLS]** setup, we align representations from BERT$_{large}$'s even layers with the corresponding BERT$_{base}$ layers, e.g. 14-th layer in BERT$_{large}$ is aligned with 7-th in BERT$_{base}$.

Table 3 shows the results. In the homogeneous setup, the most effective variant is to align the prediction probabilities via $\mathcal{R}_{\text{KL-div}}$, where it achieves up to 58% $\mathcal{R}_{NF}$ reduction, i.e. from 4.17% to 1.72%. For $\mathcal{R}_{l_2}$ setup, aligning at all layers can further reduce $\mathcal{R}_{NF}$ compared with only aligning at the final layer. This implies a deeper alignment can help the new model more effectively learn to behave similarly as the old one when fine-tuning the same architecture with a different random seed.

In the heterogeneous setup, $\mathcal{R}_{l_2}$, **[CLS]** works the best for BERT$_{large}$ that achieves 62% $\mathcal{R}_{NF}$ reduction, with $\mathcal{R}_{\text{KL-div}}$ having a comparable performance. Overall, $\mathcal{R}_{\text{KL-div}}$ produces consistent regression reductions across different setups, which we pick it as our default setting in the distillation method.

From Table 3, we can also observe that the deeper alignment seems to hurt $\mathcal{R}_{NF}$ in the heterogeneous update setup. The reason might be that differences between pre-trained models are too significant. The distillation with simple all-layer alignment could mess up pre-trained representations rather than effectively encourage

Old: BERT$_{base}$	Acc: 86.03%			
New:	→**BERT**$_{base}$		→**BERT**$_{large}$	
	Acc	$\mathcal{R}_{NF}$	**Acc**	$\mathcal{R}_{NF}$
Baseline	86.03%	4.17%	87.75%	5.88%
Distillation - $\mathcal{R}_{\textbf{KL-div}}$, Logits	**87.01**%	1.72%	88.73%	2.45%
Distillation - $\mathcal{R}_{l_2}$, [CLS]	85.54%	3.19%	88.73%	**2.21**%
Distillation - $\mathcal{R}_{l_2}$, All [CLS]	85.54%	2.45%	87.99%	4.90%
Ensemble	86.0%	**1.23**%	**89.46**%	2.45%

Table 3: Accuracy and regression results on MRPC with BERT$_{base}$ →BERT$_{base}$ and BERT$_{base}$ →BERT$_{large}$ updates using variants of distillation and ensemble methods. Baseline is fine-tuning with different random seeds. Model accuracy and negative flip rates are averaged across 5 seeds.

new models to learn where the old model performs well.

Another interesting finding is that the simple model ensemble is a competitive solution comparing to the distillation. In the BERT$_{base}$ →BERT$_{base}$ setup, the ensemble even outperforms all the other distillation variants. This is indeed a bit counter-intuitive as the distillation explicitly encourages the new model to pick up old models' correct predictions while the ensemble does not involve the old model in the process. We conduct deeper analysis trying to understand on which aspects that these methods work to reduce the regression in the next section.

5 Analyzing Regression in Model Updates

In this section, we first analyze the model ensemble and present our hypothesis on how it reduces regression. Next, we conduct behavioral testing across diverse linguistic phenomena to see where the reduced and remaining regressions reside.

5.1 Analysis of Updating to Model Ensemble

Similar to the findings of Yan et al. (2020), we observe in Table 2 and 3 that a simple ensemble of models trained with different random initialization before finetuning can reduce regression in some cases. We fine-tune BERT$_{base}$ on MRPC with 20 random seeds as our old base models, and another 20 seeds as our new single models, and another 100 seeds for building 20 ensemble models. Next, we calculate $\mathcal{R}_{NF}$ on the dev set in each model update setup, i.e. 400 update pairs. Figure 2 plots their model update regression $\mathcal{R}_{NF}$ distributions. We observe that the ensemble can not only bring down $\mathcal{R}_{NF}$ but also reduce its variance.
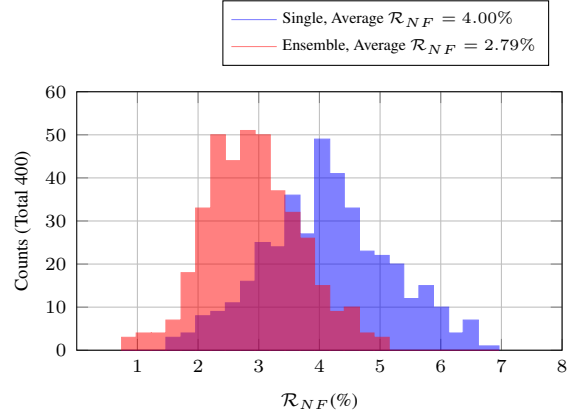
From Figure 2, we conjecture that each single

Figure 2: Pair-wise $\mathcal{R}_{NF}$ distributions of update BERT$_{base}$ →BERT$_{base}$ on MRPC, with new model being single models fine-tuned with different seeds (blue) or ensemble models (red). We train 20 old and 20 new models to calculate 400 pair-wise $\mathcal{R}_{NF}$.

model could learn a subset of all possible patterns in the data to achieve comparable accuracy on the task. Models fine-tuned with different seeds could rely on different sets of patterns for predictions, leading to behavioral difference and regression. On the other hand, ensemble aggregates distinct and complementary behaviors from individual models, leading to less eccentric behavior and increased compatible with individual models on average. In a parallel work, Zeyuan and Li (2020) provides a theoretical framework of how ensemble works from the multi-view perspective. They show that single models can pick up multiple but different views of the data, and the ensemble naturally collects more view features, leading to a higher accuracy. Our hypothesis concurs with their findings.

However, ensemble is not required to achieve moderate model behavior. To verify this, we designed the following simple model selection

		Acc	$\mathcal{R}_{NF}$
Old: BERT$_{base}$	Single	85.81($\pm$1.07)%	-
$\rightarrow$**BERT**$_{base}$	Single	85.39($\pm$1.43)%	4.30($\pm$1.53)%
	Ensemble	86.18($\pm$1.12)%	3.08($\pm$1.14)%
	Centric	87.75%	2.79($\pm$0.69)%
$\rightarrow$**BERT**$_{large}$	Single	86.32($\pm$2.50)%	5.37($\pm$2.69)%
	Ensemble	87.65($\pm$1.34)%	3.64($\pm$1.21)%
	Centric	87.25%	4.24($\pm$0.78)%

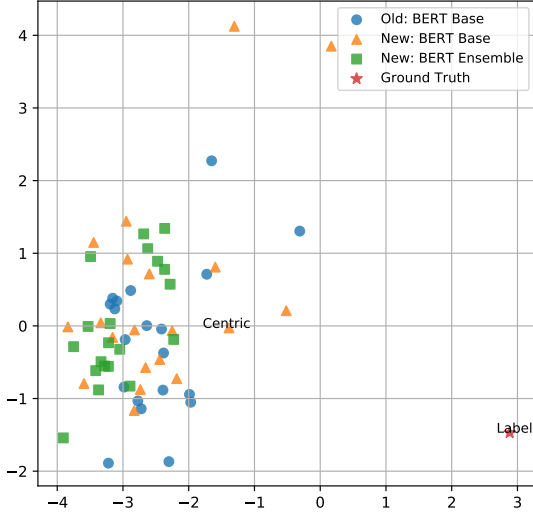Table 4: The selected single model *centric* can achieve similar accuracy and negative flip as ensemble.

Figure 3: 2D visualization by PCA of old, new single, and new ensemble models based on dev set predictions.

procedure. We first train 20 new single models, among which we compute for each model the average $\mathcal{R}_{NF}$ on the first half of dev set when updating from the other 19 models. We then select the single model with the lowest average $\mathcal{R}_{NF}$ as the *centric*. Results in Table 4 show the accuracy and $\mathcal{R}_{NF}$ on the second half of the dev set. Indeed the single *centric* model achieves substantial reduction in $\mathcal{R}_{NF}$ comparable to model ensemble. We further plot all the BERT$_{base}$ models based on their class predictions down-projected by PCA (Hotelling, 1933). Figure 3 shows that single models tend to spread while ensembles are more concentrated and close together. We can also see that the *centric* indeed sits near the center of single model cluster. In essence, the *centric* model is a single model that requires much less compute resource than the ensemble model during inference, yet can achieve comparable performance and reductions in regression.

5.2 Analyzing Regression with Linguistic Behavioral Testing

To further understand where the regression happens and how the above methods contribute to reducing regression, we conduct qualitative analysis across diverse linguistic phenomena. More precisely, we leverage the CHECKLIST (Ribeiro et al., 2020) behavioral testing and construct regression sets for relevant linguistic capabilities and tests based on perturbations and provided templates. For example, to test the capability of dealing with lexical taxonomy in the paraphrase detection task, we replace adjectives in one sentence with their synonyms with the label unchanged and expect the model can still predict correctly. We manually set the templates, apply CHECKLIST to automatically generate testing sentence pairs, and calculate $\mathcal{R}_{NF}$ for each linguistic test. Detailed linguistic behavioral testing setups with examples can be found in Appendix C.

Table 5 shows the linguistic behavioral testing results when updating from BERT$_{base}$ - 1 Seed to BERT$_{large}$. Each row denotes one specific behavioral test and 500 cases are sampled in each test. We focus on negative flips where the new model fails the test while the old model passes. We can observe that the vanilla fine-tuned BERT$_{large}$ has significant regressions on switching with synonyms, asymmetric ordering, and active-passive swap related to people names (see Appendix C). Also, we observe that models tend to either fail or pass almost all cases in a test, which leads to high variances in $\mathcal{R}_{NF}$. This implies that models fine-tuned with different seeds can have different behavioral patterns, which could be one source of regression.

Furthermore, Table 5 shows that the distillation can effectively reduce regressions across almost all types of behavioral tests. This demonstrates that minimizing the surrogate regression measure, formulated as a knowledge distillation objective, reduces the regression through actually aligning new model's behavior with the old model.

For the ensemble, although it can reduce significant regressions in the benchmark, we observe that it can only improve the model update compatibility on a handful of capabilities. We hypothesize that the ensemble mostly improves the compatibility with the underlying constituent models. Without an explicit alignment, it cannot proactively reduce the regression on certain behavior tests when updating

	Old: BERT$_{base}$ (Error %)	New: BERT$_{large}$ ($\mathcal{R}_{NF}$)					
	1 Seed	1 Seed	KD 1 Seed	5 Seeds	KD 5 Seeds	Centric	Ensemble
Coref - He/She	0.0%	13.4%	22.4%	12.3($\pm$16.1)%	41.2($\pm$47.0)%	0.0	0.0($\pm$0.0)%
Vocab - People	0.0%	13.8%	**1.9%**	59.5($\pm$34.8)%	34.1($\pm$47.1)%	89.0%	55.2($\pm$50.8)%
Vocab - More/Less	100.0%	0.0%	0.0%	0.0($\pm$0.0)%	0.0($\pm$0.0)%	0.0%	0.0($\pm$0.0)%
Taxonomy - Synonym	0.0%	42.3%	**0.0%**	77.0($\pm$26.2)%	20.3($\pm$44.5)%	73.1%	61.9($\pm$54.1)%
SRL - Pharaphrase	0.0%	12.5%	99.9%	63.2($\pm$43.4)%	42.0($\pm$48.3)%	26.9%	61.9($\pm$54.1))%
SRL - Asymmetric Order	0.0%	47.1%	**0.0%**	70.7($\pm$20.0)%	22.3($\pm$43.7)%	92.0%	58.6($\pm$52.2)%
SRL - Active/Passive 1	0.1%	9.3%	65.2%	58.0($\pm$43.2)%	52.7($\pm$50.1)%	98.5%	42.4($\pm$51.5)%
SRL - Active/Passive 2	0.1%	90.0%	**0.7%**	95.5($\pm$4.8)%	23.8($\pm$43.3)%	99.6%	65.2($\pm$56.5)%
SRL - Active/Passive 3	99.9%	0.1%	0.0%	0.1($\pm$0.0)%	**0.0**($\pm$0.0)%	0.1%	0.1($\pm$0.1)%
Temporal - Before/After	100.0%	0.0%	0.0%	0.0($\pm$0.0)%	0.0($\pm$0.0)%	0.0%	0.0%
Average	38.5%	19.9%	**14.8%**	38.2%	**20.3%**	43.8%	33.3%

Table 5: Behavioral tests with CHECKLIST. Second column shows the error rate of the old model. Remaining columns are $\mathcal{R}_{NF}$ compared with the old model. The columns with *1 Seed* represent the results with random seed equals to 0, while columns with *5 seeds* represent 5 seed average. The columns with *KD* are the models after distillation. The columns with *Centric* means the single selected with the method mentioned in Section 5.1.

from other distinct models.

6 Related work

6.1 Model Update Regression and Solutions

The backward compatibility representation learning first comes into sight in Shen et al. (2020) on learning inter-operabile visual embeddings for image retrieval tasks. Later, Yan et al. (2020) formalize the model update regression problem in machine learning and explore solutions on image classification tasks. They suggest negative flip (NF) as the empirical measurement of regression and propose a specialized knowledge distillation loss (Hinton et al., 2015) as a surrogate of regression for joint optimizations. Our work investigates the model update regression in NLP classification tasks, which involve discrete signals and rich linguistic structures. We formulate our solutions from the perspective of constraint satisfaction and verify their efficacy on scenarios including update to distinct architectures.

6.2 Transfer Learning, Lifelong Learning and Concept Drifting

Pre-training a model on large corpora and fine-tuning on downstream tasks has emerged as a standard paradigm in NLP (Devlin et al., 2019; Lan et al., 2020; CONNEAU and Lample, 2019; Raffel et al., 2020; Brown et al., 2020; Clark et al., 2020). Our work follows this transfer learning paradigm but our main focus is to investigate the regression phenomenon when updating backbone pre-trained models. Another related stream of research is lifelong learning (Lopez-Paz and Ranzato, 2017;

Yoon et al., 2018; Delange et al., 2021; Sun et al., 2019; Chuang et al., 2020), incremental learning (Rebuffi et al., 2017; Chaudhry et al., 2018; Prabhu et al., 2020), or concept drifting (Schlimmer and Granger, 1986; Tsymbal, 2004; Klinkenberg, 2005; Žliobaitė I., 2016) which aims to accumulate knowledge learned either in previous tasks or from data with changing distribution. The model update regression problem differs in that models are trained on the same task and dataset, but we update from one model to another.

6.3 Behavioral Testing of NLP Models

To analyze whether a fine-tuned model can handle linguistic phenomena for a specific end task, perturbation techniques are often used (Belinkov and Bisk, 2018; Ribeiro et al., 2018; Prabhakaran et al., 2019; Wu et al., 2019; Talmor et al., 2020). In particular, CHECKLIST (Ribeiro et al., 2020) leverages and expands those techniques to efficiently evaluate a wide range of linguistic behavioral capabilities of NLP models. Our work applies CHECKLIST to inspect where the model update regressions come from and on which linguistic phenomena our proposed solutions help to reduce regressions.

7 Conclusion

In this work, we investigated the regression in NLP model updates on classification tasks and show that it has a prevalent presence across tasks and models. We formulated the regression-free model update problem as a constrained optimization problem and reduce it into a joint learning objective on target

task while distilling from the old model. Together with the ensemble, these methods can cut down the regression by 60% at best. Experiments on the GLUE benchmark showed that ensemble can be effective in reducing the regression when updating to homogeneous models. On the other hand, knowledge distillation produced more significant regression reductions under the heterogeneous setting. Through linguistic behavioral testing we showed that distillation can reduce the regression across a wider range of linguistic phenomena than ensemble method. While the regression reduction achieved by the discussed methods are promising, they are far from reaching regression-free. We leave the design of more advanced regression-reduction methods as future works.

Acknowledgments

References

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*.

Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, and Bernardo Magnini. 2009. The fifth pascal recognizing textual entailment challenge. In *Proceedings of TAC*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14.

Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. 2018. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 532–547.

Z. Chen, H. Zhang, X. Zhang, and L. Zhao. 2018. Quora question pairs.

Yung-Sung Chuang, Shang-Yu Su, and Yun-Nung Chen. 2020. Lifelong language knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2914–2924, Online. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Alexis CONNEAU and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Matthias Delange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. 2021. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1.

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.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Harold Hotelling. 1933. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417.

Gama J. Žliobaitė I., Pechenizkiy M. 2016. Chapter 1 - introduction. In Stefanowski J. Japkowicz N., editor, *An Overview of Concept Drift Applications.*, volume 16 of *Big Data Analysis: New Algorithms for a New Society. Studies in Big Data*. Springer.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. Tiny{bert}: Distilling {bert} for natural language understanding.

Ralf Klinkenberg. 2005. Meta-learning, model selection, and example selection in machine learning domains with concept drift.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Roberta: A robustly optimized bert pretraining approach.

David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. In *Advances in neural information processing systems*, pages 6467–6476.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035.

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 5744–5749.

Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. 2020. Gdumb: A simple approach that questions our progress in continual learning. In *European Conference on Computer Vision*, pages 524–540. Springer.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. 2017. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging nlp models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865.

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, Online. Association for Computational Linguistics.

Jeffrey C. Schlimmer and Richard H. Granger. 1986. Beyond incremental processing: Tracking concept drift. In *Proceedings of the Fifth AAAI National Conference on Artificial Intelligence*, AAAI'86, page 502–507. AAAI Press.

Yantao Shen, Yuanjun Xiong, Wei Xia, and Stefano Soatto. 2020. Towards backward-compatible representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

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

Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. 2019. Lamol: Language modeling for lifelong language learning. In *International Conference on Learning Representations*.

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2020. olmpics-on what language model pre-training captures. *Transactions of the Association for Computational Linguistics*, 8:743–758.

A. Tsymbal. 2004. The problem of concept drift: definitions and related work.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122.

Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel S Weld. 2019. Errudite: Scalable, reproducible, and testable error analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 747–763.

Sijie Yan, Yuanjun Xiong, Kaustav Kundu, Shuo Yang, Siqi Deng, Meng Wang, Wei Xia, and Stefano Soatto. 2020. Positive-congruent training: Towards regression-free model updates. *arXiv preprint arXiv:2011.09161*.

Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. 2018. Lifelong learning with dynamically expandable networks. In *International Conference on Learning Representations*.

Zhu Zeyuan and Yuanzhi Li. 2020. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *arXiv preprint arXiv:2012.09816*.

A Full Results of Regression Between SOTA Model Updates

Due to the page limitation, we present the full regression update comparison between commonly used pre-trained model pairs (Devlin et al., 2019; Liu et al., 2020; Lan et al., 2020; Clark et al., 2020) in Table 6.

We show regression in model updates from BERT_{base} to the other common used pre-trained models; we also show the regression in updates from BERT_{large} to $\text{BERT}_{large-whole-word-masking}$ and updating from ROBERTA_{base} to ELECTRA_{base}.

Other than the universal presence of the regression, Table 6 shows that:

1. **The more difference in pre-training method, the higher regression can be observed.** From the results of updating from BERT_{base} experiment group we can have this conclusion: The updating factors can increase regression in ascending order: hyperparameters (to BERT_{base}), pre-train settings(to ROBERTA_{base}), model size (to BERT_{large}), pre-train objection (to ELECTRA_{base}), model structure(to ALBERT_{base})

2. **Similar updating factor results in similar level of regression.** Updating from BERT_{large} to $\text{BERT}_{large-wwm}$ have similar regression level as in updating from BERT_{base} to ELECTRA_{base}, both are updating in the pre-training objective. Similarly, Updating from ROBERTA_{base} to ELECTRA_{base} have similar regression as in updating from BERT_{base} to ELECTRA_{base}.

B Selection of Regression Set During Training

Here, we explore the difference of regressing set selection during knowledge distillation.

For the regression set used in the model training process, we propose several options:

1. Take the entire training set as our the regression set in training $\mathcal{D}_{reg} = \mathcal{D}_{train}$

2. Training examples where the old model makes correct predictions $\mathcal{D}_{reg} = \mathcal{D}_{correct}$

3. Training examples where the old model gets a higher predict probability on the ground-truth class than the new model $\mathcal{D}_{reg} = \mathcal{D}_{better}$, equivalent to adjusting α dynamically according to the performance of the two models, we set α to zero when $p_{\phi_{old}}(y|x) < p_{\phi_{new}}(y|x)$

4. Extra data from other tasks $\mathcal{D}_{reg} = \mathcal{D}_{extra}$

5. User-provided regression set, which includes examples with high-stakes $\mathcal{D}_{reg} = \mathcal{D}_{user}$

We experiment with all options except for the user-provided regression set, see Table 7.

Dynamically adapting the regression set according to the current performance of the new model in **Distillation** ($\mathcal{R}_{\text{KL-div}}, \mathcal{D}_{better}$) offers the most reduction in the regression without sacrificing the accuracy. We conjecture that it's because we apply the soft regression-free constraint loss precisely on examples where the new model's performance is behind.

C Linguistic Behaviour Test settings

In the linguistic behaviour tests, we go through a variety range of linguistic aspects and design test examples following CheckList(Ribeiro et al., 2020).

In Table 8 we show the tests for linguistic behaviour tests. Please find the example test cases in the third column for each testing.

D GLUE Details

The GLUE datasets are described as follows(Jiao et al., 2020):

MNLI. Multi-Genre Natural Language Inference is a large-scale, crowd-sourced entailment classification task (Williams et al., 2018). Given a pair of $\langle premise, hypothesis \rangle$, the goal is to predict whether the $hypothesis$ is an entailment, contradiction, or neutral with respect to the $premise$.

QQP. Quora Question Pairs is a collection of question pairs from the website Quora. The task is to determine whether two questions are semantically equivalent (Chen et al., 2018).

QNLI. Question Natural Language Inference is a version of the Stanford Question Answering Dataset which has been converted to a binary sentence pair classification task by Wang et al. (2018). Given a pair $\langle question, context \rangle$. The task is to determine whether the $context$ contains the answer to the $question$.

SST-2. The Stanford Sentiment Treebank is a binary single-sentence classification task, where

		CoLA	SST-2	MRPC	QQP	MNLI-m	MNLI-mm	QNLI	RTE
Train size		8.6k	67k	3.7k	360k	390k	390k	100k	2.5k
Dev size		1k	0.9k	0.4k	40k	9.8k	9.8k	5.5k	0.3k
Old: BERT$_{base}$	Acc	82.84%	92.20%	86.03%	90.76%	83.82%	84.13%	91.07%	67.15%
→**BERT**$_{base}$	Acc	83.80%	91.93%	86.03%	90.56%	83.55%	83.94%	90.65%	63.18%
	$\mathcal{R}_{NF}$	3.36%	2.10%	4.17%	2.32%	3.56%	3.67%	2.35%	11.43%
→**BERT**$_{large}$	Acc	85.43%	93.23%	87.75%	91.11%	86.10%	86.49%	92.53%	66.43%
	$\mathcal{R}_{NF}$	3.16%	1.95%	5.88%	2.82%	3.95%	3.69%	2.64%	12.27%
→**ROBERTA**$_{base}$	Acc	84.85%	94.11%	89.22%	91.25%	87.58%	87.74%	92.71%	63.17%
	$\mathcal{R}_{NF}$	4.67%	1.22%	4.66%	1.98%	2.64%	2.38%	1.74%	13.1%
→**ELECTRA**$_{base}$	Acc	85.81%	95.41%	86.03%	91.35%	88.87%	88.67%	93.30%	72.92%
	$\mathcal{R}_{NF}$	5.18%	1.38%	5.39%	3.20%	3.50%	3.57%	2.65%	7.58%
→**ALBERTA**$_{base}$	Acc	76.51%	91.86%	86.27%	90.73%	85.26%	85.14%	91.67%	74.73%
	$\mathcal{R}_{NF}$	10.74%	3.67%	6.86%	3.78%	5.22%	5.24%	3.70%	9.03%
Old: BERT$_{large}$	Acc	85.43%	93.23%	87.75%	91.11%	86.10%	86.49%	92.53%	66.43%
→**BERT**$_{large-wwm}$	Acc	85.14%	94.15%	87.01%	91.52%	86.75%	87.24%	93.34%	70.76%
	$\mathcal{R}_{NF}$	5.05%	1.60%	7.82%	3.03%	5.08%	4.67%	2.75%	15.22%
Old: ROBERTA$_{base}$	Acc	84.85%	94.38%	87.25%	91.28%	88.24%	87.63%	92.51%	70.76%
→**ELECTRA**$_{base}$	Acc	85.81%	95.41%	86.03%	91.35%	88.87%	88.67%	93.30%	72.92%
	$\mathcal{R}_{NF}$	4.31%	1.49%	6.62%	2.77%	3.47%	3.48%	2.62%	7.58%

Table 6: Accuracy and regression measures of different model update variants on GLUE benchmark. BERT$_{large-wwm}$ represents the whole-word-masking version.

Old: BERT$_{base}$	**Acc: 86.03%**			
New:	→**BERT**$_{base}$		→**BERT**$_{large}$	
	Acc	$\mathcal{R}_{NF}$	**Acc**	$\mathcal{R}_{NF}$
Baseline	86.03%	4.17%	87.75%	5.88%
Distillation($\mathcal{R}_{KL\text{-}div}, \mathcal{D}_{train}$)	**87.25%**	2.94%	88.73%	3.68%
Distillation($\mathcal{R}_{KL\text{-}div}, \mathcal{D}_{correct}$)	87.01%	3.92%	88.48%	4.90%
Distillation($\mathcal{R}_{KL\text{-}div}, \mathcal{D}_{better}$)	87.01%	**1.72%**	88.73%	2.45%
Distillation($\mathcal{R}_{l_2}, \mathcal{D}_{better}$)	85.29%	2.45%	88.73%	**2.21%**

Table 7: Results on MRPC of our proposed techniques towards regression-free model updates. The model is updated fromBERT$_{base}$ to BERT$_{base}$, e.g. change fine-tune setups, or BERT$_{large}$. $\mathcal{D}_{(\cdot)}$ denotes the regression set for joint training with distillation and classification.

the goal is to predict the sentiment of movie reviews (Socher et al., 2013).

CoLA. The Corpus of Linguistic Acceptability is a task to predict whether an English sentence is a grammatically correct one (Warstadt et al., 2019).

STS-B. The Semantic Textual Similarity Benchmark is a collection of sentence pairs drawn from news headlines and many other domains (Cer et al., 2017). The task aims to evaluate how similar two pieces of texts are by a score from 1 to 5.

MRPC. Microsoft Research Paraphrase Corpus is a paraphrase identification dataset where systems aim to identify if two sentences are paraphrases of each other (Dolan and Brockett, 2005).

RTE. Recognizing Textual Entailment is a binary entailment task with a small training dataset (Bentivogli et al., 2009).

Category	Description	Example	Label
Coref - He/She	Reverse he or she.	If Charles and Jessica were alone , do you think **he** would reject **her**? If Charles and Jessica were alone , do you think **she** would reject **him**?	False
Vocab - People	Add modifiers that preserve sentence semantics.	Wendy is friendly to Kevin. Wendy is **truely** friendly to Kevin.	True
Vocab - More/Less	Swap more with less.	I can become **more** passive. I can become **less** passive.	True
Taxonomy - Synonym	Replace synonym.	I can become more **courageous**. I can become more **brave**.	True
SRL - Pharaphrase	Somebody think → According to Somebody.	Who do conservatives think is the happiest surgeon in the world ? Who is the happiest surgeon in the world according to conservatives ?	True
SRL - Asymmetric Order	Order does matter for asymmetric relations.	**Shannon** is proposing to **Samantha**. **Samantha** is proposing to **Shannon**.	False
SRL - Active/Passive 1	Traditional SRL: active / passive swap	Jeremy missed the game. The game was missed by Jeremy.	True
SRL - Active/Passive 2	Traditional SRL: active / passive swap. with people.	Christian remembers Alyssa. Alyssa is remembered by Christian.	True
SRL - Active/Passive 3	Traditional SRL: wrong active / passive swap.	Sara **took** the castle. Sara **was taken** by the castle.	False
Temporal - Before/After	Before becoming somebody → after becoming somebody.	What was Noah Myers 's life **before** becoming an architect ? What was Noah Myers 's life **after** becoming an architect ?	False

Table 8: Regression Tests details with CheckList.