Learning Non-linguistic Skills without Sacrificing Linguistic Proficiency

Mandar Sharma
Virginia Tech
mandarsharma@vt.edu

Nikhil Muralidhar
Stevens Institute of Technology
nmurali1@stevens.edu

Naren Ramakrishnan
Virginia Tech
naren@cs.vt.edu

Abstract

The field of Math-NLP has witnessed significant growth in recent years, motivated by the desire to expand LLM performance to the learning of non-linguistic notions (numerals, and subsequently, arithmetic reasoning). However, non-linguistic skill injection typically comes at a cost for LLMs: it leads to catastrophic forgetting of core linguistic skills, a consequence that often remains unaddressed in the literature. As Math-NLP has been able to create LLMs that can closely approximate the mathematical skills of a grade-schooler or the arithmetic reasoning skills of a calculator, the practicality of these models fail if they concomitantly shed their linguistic capabilities. In this work, we take a closer look into the phenomena of catastrophic forgetting as it pertains to LLMs and subsequently offer a novel framework for non-linguistic skill injection for LLMs based on information-theoretic interventions and skill-specific losses that enable the learning of strict arithmetic reasoning. Our model outperforms the state-of-the-art both on *injected non-linguistic skills* and on *linguistic knowledge retention*, and does so with a fraction of the non-linguistic training data ($1/4$) and zero additional synthetic linguistic training data. Our pre-trained models and experimentation codebases are hosted online[1].

1 Introduction

Numeracy, involving the comprehension of sizes, magnitudes, and order, is the most prevalent form of *non-linguistic* information embedded in textual corpora (Joram et al., 1995). Thus, the case for numerically-capable LLMs is rather easy to make: as numerals grant objectivity to language (Porter, 1996), numerically-capable language models are key to optimal performance in a host of downstream tasks such as information extraction (Madaan et al., 2016), inference (Naik et al., 2018), and data-to-text generation (Sharma et al., 2021, 2022a).

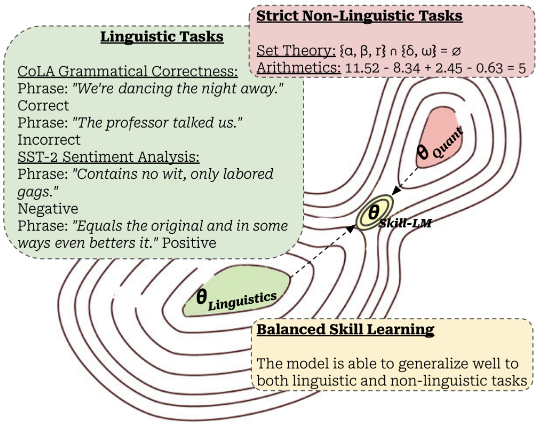

Linguistic Tasks
CoLA Grammatical Correctness:
Phrase: "We're dancing the night away." Correct
Phrase: "The professor talked us." Incorrect
SST-2 Sentiment Analysis:
Phrase: "Contains no wit, only labored gags." Negative
Phrase: "Equals the original and in some ways even betters it." Positive
Strict Non-Linguistic Tasks
Set Theory: {α, β, r} ∩ {δ, ω} = ∅
Arithmetics: 11.52 - 8.34 + 2.45 - 0.63 = 5
θ Quant
θ Skill-LM
θ Linguistics
Balanced Skill Learning
The model is able to generalize well to both linguistic and non-linguistic tasks

Figure 1: LLMs trained for dissimilar skillsets have different convergence points for their parameters - the parameterization space for an LLM trained for linguistic skills $\theta_{Linguistics}$ lives in the green space while the parameterization space for an LLM trained for quantitative reasoning θ_{Quant} lives in the red space. The goal of this work is to approximate a locality of parameterization $\theta_{Skill-LM}$ (yellow) where the model reliably learns a non-linguistic skill (quantitative reasoning) without sacrificing its linguistic proficiency.

1.1 Re-thinking the Objective of Math-NLP

Progress in Math-NLP: Several notable publications in the Math-NLP space have made rapid strides in numeracy-tinged language-modeling (Thawani et al., 2021) - from investigations of the inherent deficiency of numerical reasoning skills in LLMs induced through unsupervised training, both for numerals that appear in the training corpus (Zhang et al., 2020) and OOD (out-of-domain) numerals (Wallace et al., 2019; Razeghi et al., 2022), to interventions that strengthen the numerical reasoning skills of these models (Spithourakis and Riedel, 2018; Jiang et al., 2020; Geva et al., 2020). Further, advances in chain-of-thought prompting in few-shot learning settings (Li et al., 2022) and

[1] https://github.com/Mandar-Sharma/Skill-LM

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 6178–6191
July 9-14, 2023 ©2023 Association for Computational Linguistics

Model	CoLA	STS-B	MNLI	MNLI$_{MM}$	MRPC	QNLI	QQP	RTE	SST-2	WNLI
BERT	*0.59*	*0.89*	*83.85*	*84.05*	*86.76*	*90.55*	*90.61*	*65.34*	*91.62*	*56.33*
BERT$_{Arith}$	0.08	0.80	32.73	32.95	70.34	50.53	70.49	47.29	88.07	56.33

Table 1: *LLMs trained for niche non-linguistic skill-sets forget linguistics:* Comparative analysis between the performance of the base BERT model and the same model further trained on an arithmetic reasoning corpus on the set of 9 GLUE tasks for natural language understanding. All tasks except WNLI suffer severe performance degradation as a consequence of continued training on a non-linguistic corpus.

task-specific fine-tuning (Lewkowycz et al., 2022) have shown significant gains in the capacity for quantitative reasoning in LLMs.

Linguistic evaluation remains important: As notable as these accomplishments are, the goal remains not to replicate the reasoning capabilities of a grade-schooler or to proxy a calculator, but rather build LLMs that are *empowered* with these skills. As such, an area that often goes unaddressed in the Math-NLP space is how these models perform as *general language modelers*. With the advent and popularity of generative conversational models (OpenAI, 2022), the goal is to have one model capable of a host of skills - not to load separate models for conversation/assistance and reasoning. As depicted in Figure 1, whether a model is designed to perform strict non-linguistic tasks or semi-linguistic tasks, it should never come at the cost of core linguistic competency. After all, language models are intended to *model language*.

1.2 Necessitating the Re-thinking

LLMs injected with non-linguistic skills forgo their linguistic skills: Consider the task of strict arithmetic reasoning as shown in Figure 1, a subset of possible quantitative reasoning tasks. If a base BERT model (Devlin et al., 2019) is further trained on this non-linguistic task, it suffers significant degradation on 8/9 GLUE tasks (Wang et al., 2018) that evaluate the natural language understanding (NLU) capabilties of the model, as showcased in Table 1. This observation has long been known in the deep learning literature as *catastrophic forgetting* (Kirkpatrick et al., 2017), wherein when a model pre-trained on task A is further trained on task B, the parameters in the model vital for task A adapt their values to meet the requirements of task B.

LLMs exhibit unconventional forgetting: What is interesting, based on our findings, is that in the case of LLMs, the forgetting of linguistic skills is not evenly spread - the forgetting is rather *task-specific*. Akin to other neural network applications, the forgetting of linguistic skills may likely be grouped as performance loss over a single task A; however, as seen in Table 1, the GLUE tasks suffer various ranges of degradation - the task of finding the referent of a pronoun (WNLI, Levesque et al. (2012)) does not seem to suffer at all, while the grammatical correctness assessment task (CoLA, Warstadt et al. (2019)) suffers severe degradation.

As proponents for *skill-empowered* LLMs, we thus make a case for disclosing the performance on general NLU tasks when models are trained for superior performance on niche skill-sets such as non-linguistics, an area left wanting in the Math-NLP front. Because of this task-specific forgetting, quantitative reasoning models trained in a Q&A fashion may not showcase degradation in similarly modeled downstream tasks such as SQuAD (Rajpurkar et al., 2016) and DROP (Dua et al., 2019) - thus disclosing performance across a range of NLU tasks is crucial.

Substantiating forgetting on the basis of parameter sharing: To establish that observed performance degradation can indeed be accredited to catastrophic forgetting, we take an information theoretic lens to pry into parameter-sharing tendencies across tasks with the aid of Fisher information (Rissanen, 1996). For a single sample y drawn from a distribution with probability desnity $f(y;\theta)$, the Fisher information index $I(\theta)$ (1) quantifies the sensitivity of the parameter θ to the data instance y. Thus, given a task-specific training corpus $(X, Y) \in D_{task}$, we can estimate the sensitivity of each model parameter $\theta_i \in \theta$ for the given task.

$$I(\theta_i) = E_{y \in Y}(\frac{d \log f(y;\theta_i)}{d\theta_i})^2 \tag{1}$$

$$= -E_{y \in Y}(\frac{d^2 \log f(y;\theta_i)}{d\theta_i^2}) \tag{2}$$

Using this formulation, we compute the Fisher parameter sensitivities $I(\theta)$ for four different models based on continued training of the base BERT model on four datasets:

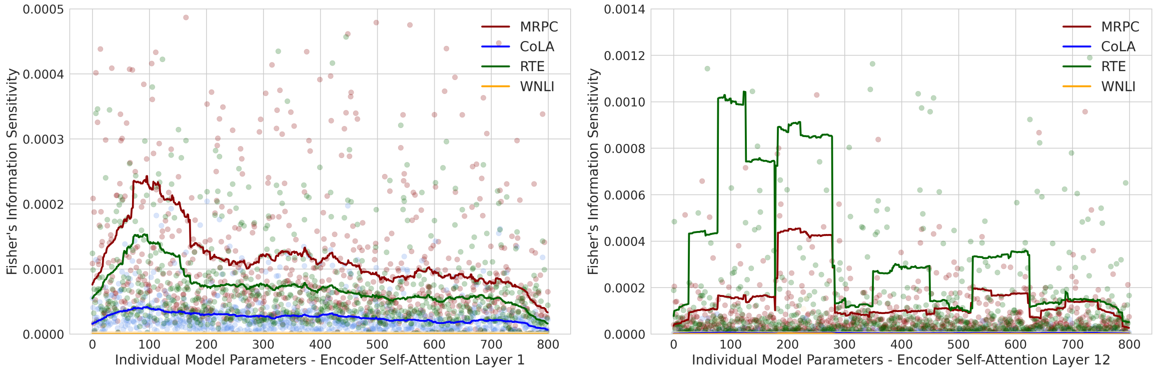


Figure 2: *NLU tasks MRPC, RTE, and CoLA, suffer the most as a consequence of non-linguistic skill-injection, because these tasks deem the same subset of model parameters to be vital as the non-linguistic (arithmetic) task:* Given Fisher parameter sensitivities $I(\theta)$ of the first (left) and last (right) self-attention encoder layers for four different models based on continued training of the base BERT model on four datasets: $I_{arith}(\theta)$ on an arithmetic reasoning and $I_{CoLA}(\theta)$, $I_{MRPC}(\theta)$, $I_{RTE}(\theta)$ on GLUE tasks CoLA, MRPC, and RTE respectively, this plot takes the $n = 800$ most crucial parameters based on $I_{arith}(\theta)$ and showcases how sensitive those *same* parameters are to the GLUE tasks based on $I_{CoLA}(\theta)$, $I_{MRPC}(\theta)$, and $I_{RTE}(\theta)$. These plots correlate to the task-specific performance degradation showcased in Table 1.

- $I_{arith}(\theta)$: for BERT trained on an arithmetic reasoning dataset (Geva et al., 2020)
- $I_{CoLA}(\theta)$, $I_{MRPC}(\theta)$, $I_{RTE}(\theta)$: for BERT trained on three GLUE (Wang et al., 2018) tasks CoLA, MRPC, and RTE respectively

To ground our hypothesis of task-specific forgetting as a consequence of parameter-sharing, first, we select $n = 800$ parameters deemed most sensitive for arithmetic reasoning from $I_{arith}(\theta)$, and compare how important those *same* parameters are for the three GLUE tasks based on their respective Fisher scores $I_{CoLA}(\theta)$, $I_{MRPC}(\theta)$, $I_{RTE}(\theta)$ (see Appendix §A.1.1 for details on Fisher score computations). As seen in Figure 2, for the first and last self-attention encoder layers, the sensitivities of the parameters across tasks correlate well with the findings of Table 1 - the NLU task that suffers the least performance degradation (WNLI) also has the least sensitivity to these (shared) parameters across the encoder layers, while the NLU tasks that do suffer from performance degradation (MRPC, CoLA, RTE) have varying ranges of shared sensitivities across the encoder self-attention layers. These findings hold consistent across all 12 encoder layers of the BERT model (see Appendix §A.1.2).

Our contributions: In line with the above observations, we offer the following contributions in the form of our proposed model, *Skill-LM*, for non-linguistic skill injection in LLMs:

- Novel multi-task skill-injection loss that infuses a sense of numeral structure in the learned representations, leading to better generalization performance than the state-of-the-art, all with a significantly lower fraction ($\frac{n}{4}$) of training data.
- Weight consolidation schemes for LLMs for better linguistic retention with *0* additional linguistic samples compared to *1 million* synthetic textual training samples used by the state-of-the-art.
- Through exhaustive qualitative and quantitative evaluations, we demonstrate the improved generalization performance of Skill-LM over the state-of-the-art. Our experiments also highlight the need for disclosing linguistic performance for models trained on highly-niche non-linguistic tasks.

2 Designing Skill-LM

2.1 Non-Linguistic Learning

Based on probabilistic modeling, language models are trained to output the next sequential token y_t at timestep t based on the n tokens already predicted by the model, formulated as $P(y_t|y_{t-1}, ..., y_{t-n}) = P(y_t|y_{<t})$. This probability distribution P is often optimized through measures of uncertainty such as cross-entropy or KL-divergence. The application of these same loss functions used for learning linguistic token distributions may not necessarily translate to the learning

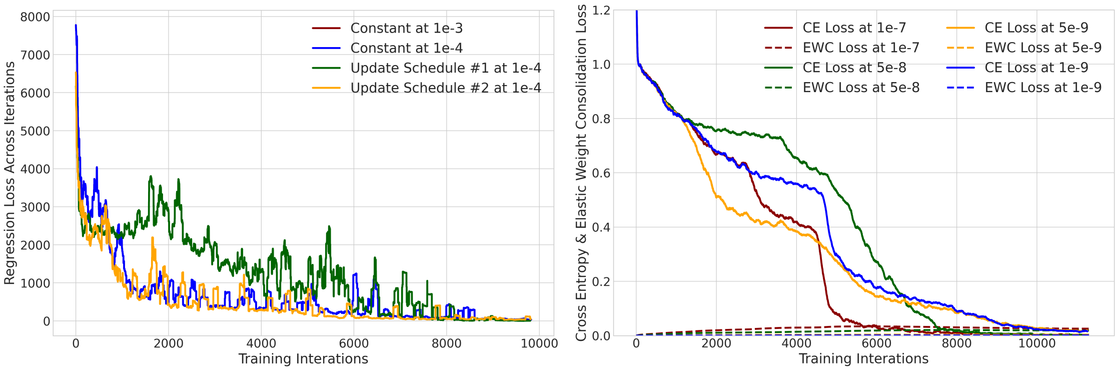


Figure 3: Empirical setting of hyperparameters: For regression loss $\mathcal{L}_{REG}$ (left), the loss convergence is evaluated at four configurations of λ_1 on the validation set with both constant $\{1e^{-3}, 1e^{-4}\}$ and dynamic (update scheduling) initializations. For EWC loss $\mathcal{L}_{EWC}$ (right), the interplay between $\mathcal{L}_{EWC}$ and CE loss $\mathcal{L}_{CE}$ (color-matched) is evaluated through a parameter sweep of λ_2 within $\{1e^{-6}, 1e^{-10}\}$ with the best performing configurations plotted.

of non-linguistic entities.

Unlike linguistic tokens, the magnitude of a numeral is especially tied to its meaning (Dehaene et al., 1998). This magnitude can either be modeled as a continuous linear representation (Dehaene et al., 1990) or a log-compressive representation (Dehaene, 2003). Thus, to inject this numeric-scale representation into a language model, we take a simplistic approach of augmenting the learning of tokens through cross-entropy $\mathcal{L}_{CE}$ with a regression loss $\mathcal{L}_{REG}$[2]. This regression loss is incorporated into the quantitative reasoning loss function $\mathcal{L}_Q$ as represented in (3, 5).

$$\mathcal{L}_Q(\theta) = \mathcal{L}_{CE} + \lambda_1 \,.\, \mathcal{L}_{REG} \tag{3}$$

$$\mathcal{L}_{CE} = -log(P(y_t|y_{<t})) \tag{4}$$

$$\mathcal{L}_{REG} = \sqrt{\sum_{i=1}^{n}(y^2 - \hat{y}^2)} \tag{5}$$

Figure 3 (left) depicts the convergence of $\mathcal{L}_{REG}$ for different configurations of λ_1. Please see Appendix §A.2.1 for further details on the update schedules for hyperparameter tuning.

2.2 Linguistic Retention

Among prominent strategies for multitask learning, a system-level consolidation scheme consists of stitching-together amalgamated datasets constituting multiple-shared tasks (Kumaran et al., 2016). However, due to the limitless range of possible downstream tasks that LLMs are often employed for, the paradigm consists of building large models that hold linguistic prowess and are intended to be fine-tuned on a single downstream task (Devlin et al., 2019; Brown et al., 2020), thus suited for a continual learning paradigm. As depicted in Figure 1, the high degree of parameterization of these models leads to the belief that there is a solution for {task B, θ_B}, a non-linguistic skill, that is proximal to the linguistic solution space for the model {task A, θ_A} (Sharma et al., 2022b). To enable this continual learning, we adapt the elastic weight consolidation (EWC) regularization to LLMs - elastic as it functions as a spring, anchoring the solution space closer to θ_A (Kirkpatrick et al., 2017). Thus, EWC penalizes changes to specific network weights deemed vital for linguistics while injecting non-linguistic skills into the model.

In line with our task-specific parametric observations from Introduction §1.2, we compute $F = I_{BERT}(\theta)$, the Fisher information index for the base BERT model based on a portion of its original pre-training corpus - WikiText (Merity et al., 2016), thus approximating its posterior distribution. Let us assume that θ^*_{ling} represents the set of parameters of a converged base-BERT model pre-trained for *linguistics*. We now introduce the quadratic penalty $\mathcal{L}_{EWC}$ (6, 7) that penalizes changes to any model parameter i crucial to the core linguistic functionality of the pre-trained model.

$$\mathcal{L}(\theta) = \mathcal{L}_Q(\theta) + \lambda_2 \,.\, \mathcal{L}_{EWC} \tag{6}$$

$$\mathcal{L}_{EWC} = \sum_i \frac{1}{2}\, F_i(\theta_i - \theta^*_{ling,i})^2 \tag{7}$$

In this loss formulation, the hyperparameter λ_2

[2] As in the initial phases of model training, incorrect predictions of target numerals can lead to exceedingly large values of $\mathcal{L}_{REG}$, thus our choice of seed values for λ_1 were set to $\{1e^{-3}, 1e^{-4}\}$ as not to exceed the range of $\mathcal{L}_{CE}$ $\{0, 1\}$ by values greater than an order of magnitude.

is crucial as it dictates both model convergence and balances the learning of quantitative reasoning skills θ_Q with linguistic prowess θ_{ling}. To evaluate the sensitivity of model convergence with respect to λ_2, we perform a hyperparameter sweep between $\{1e^{-6}, 1e^{-10}\}$ - Figure 3 (right) showcases the interplay between $\mathcal{L}_{CE}$ and $\mathcal{L}_{EWC}$ (color-matched) for the best performing values of λ_2 on the validation set. The first sign of model convergence is observed at $\lambda_2 = 1e^{-7}$, and although slight improvements to model convergence are noted for even smaller values of λ_2, the smallest value that allows for convergence, theoretically, allows for balanced learning of θ_Q with θ_{ling}.

3 Experiment Setup and Results

3.1 Tasks and Datasets

The goal of Skill-LM is to empower LLMs with non-linguistic skills in a manner that avoids catastrophic forgetting of linguistic skills without the aid of additional synthetic linguistic training. Thus, we have two categories of tasks that Skill-LM, along with the baselines, should be evaluated on:

3.1.1 Quantitative Reasoning

To hold fair comparisons to GenBERT (Geva et al., 2020), we both train and evaluate all models with the arithmetic reasoning portion of their dataset. The data instances take the form of the sample arithmetic task demonstrated in Figure 1. The corpus consists of $N_{train} = 165,000$ training samples and $N_{val} = 1666$ validation samples, where the numerals are in the range $\{1, 20^3\}$ with numeral ranges stratified between the training and validation sets. For our models, we randomly sample $\frac{N_{train}}{4}$ instances for training.

OOD Performance: The out-of-domain (OOD) performance of all models are evaluated on data instances generated in the same manner but for numeral ranges $\{20^3, 10^6\}$ that are unseen for all models evaluated.

3.1.2 Natural Language Understanding

Following standard protocols, we employ all 9 tasks in the GLUE benchmark (Wang et al., 2018) as metrics for linguistic prowess of a model. The tasks, as per the benchmark, are categorized into three groups:

- Single Sentence Tasks: CoLA (the Corpus of Linguistic Acceptability) (Warstadt et al., 2019) for grammatical fidelity (Matthews correlation), SST-2 (the Stanford Sentiment Treebank) (Socher et al., 2013) for sentiment prediction

- Similarity and Paraphrase Tasks: MRPC (the Microsoft Research Paraphrase Corpus) (Dolan and Brockett, 2005), QQP (the Quora Question Pairs), and STS-B (the Semantic Textual Similarity Benchmark) (Cer et al., 2017) for semantic equivalence

- Inference Tasks: MNLI (the Multi-Genre Natural Language Inference Corpus) (Williams et al., 2018) and RTE (Recognizing Textual Entailment) for textual entailment, QNLI (the Stanford Question Answering Dataset) (Rajpurkar et al., 2016) for Q&A, and WNLI (the Winograd Schema Challenge) (Levesque et al., 2012) for pronoun referent selection.

Evaluation Metrics: Besides CoLA (evaluated using the Matthews correlation coefficient) and STS-B (evaluated using a combination of the Spearman's and Pearson's correlation coefficients), all result shown represent accuracy for the respective GLUE task.

3.2 Baselines

For assessment of both quantitative reasoning skills and linguistic prowess through natural language understanding, the following three models are used as the baselines for this experimentation. For the training specifics, please see Appendix §A.2.2.

- *BERT*: In this evaluation, this base pre-trained model establishes the standard for natural language understanding that all BERT-derivatives designed for non-linguistic skills should strive to achieve. Thus, its performance on the set of GLUE tasks are *italicized* in Table 3.

- *BERT$_{Arith}$*: This is the model generated from the continued training of the pre-trained BERT model on the quantitative reasoning dataset using the standard cross-entropy $\mathcal{L}_{CE}$ loss. This model showcases the current paradigm of skill-injection where the architecture of a model is left unchanged and the training parameters are often adapted to meet performance requirements in the target task.

- *GenBERT* (Geva et al., 2020): This BERT-based model is trained for numerical reason-

		Model Accuracy			
Model	Training Samples	Validation Set [0,20^3]	OOD [20^3,10^4]	OOD [10^4,10^5]	OOD [10^5,10^6]
GenBERT	165,000 (n)	**100%**	1.32%	0.06%	0.0%
BERT	**41,250 (n/4)**	96.63%	7.20%	**0.12%**	0.0%
Skill-LM (w/o $\mathcal{L}_{EWC}$)	**41,250 (n/4)**	95.67%	9.66%	**0.12%**	0.0%
Skill-LM	**41,250 (n/4)**	98.01%	**19.44%**	**0.12%**	0.0%

Table 2: Comparative analysis of quantitative reasoning performance between the baselines and Skill-LM (that uses $\frac{1}{4}$ of the training data) for both the in-domain validation set $\{1, 20e^3\}$ and out-of-domain (OOD) sets for numeral ranges $\{20e^3, 10e^4\}$, $\{10e^4, 10e^5\}$, and $\{10e^5, 10e^6\}$. Using a fraction ($\frac{n}{4}$) of the original training data, Skill-LM not only closely matches the in-domain performance of GenBERT, it significantly improves the generalization performance to OOD numerals ranges as well.

Model	Training Samples	CoLA	STS-B	MNLI	MNLI$_{MM}$	MRPC
BERT	-	*0.59*	*0.89*	*83.85*	*84.05*	*86.76*
BERT$_{Arith}$	**0**	0.08	0.80	32.73	32.95	70.34
GenBERT	1 Million	$0.54_{0.001}$	$0.88_{0.001}$	$83.00_{0.576}$	$83.40_{1.107}$	$85.04_{0.693}$
Skill-LM	**0**	**0.58**$_{0.041}$	**0.89**$_{0.003}$	**84.07**$_{0.158}$	**84.66**$_{1.123}$	**86.88**$_{1.123}$

Model	Training Samples	QNLI	QQP	RTE	SST-2	WNLI
BERT	-	*90.55*	*90.61*	*65.34*	*91.62*	*56.33*
BERT$_{Arith}$	**0**	50.53	70.49	47.29	88.07	56.33
GenBERT	1 Million	$90.83_{0.012}$	$90.78_{0.316}$	**67.86**$_{2.042}$	$91.51_{0.648}$	$55.63_{0.995}$
Skill-LM	**0**	**91.54**$_{0.207}$	**90.96**$_{0.043}$	$65.70_{1.531}$	**92.37**$_{0.081}$	**56.18**$_{0.216}$

Table 3: Comparative analysis of linguistic performance between the baselines and Skill-LM (that uses 0 additional linguistic training data) for the set of 9 GLUE benchmarks, each addressing a certain aspect of natural language understanding (see §3.1.2). To further authenticate the performance increase of Skill-LM over GenBERT, the results are presented as μ_σ (mean and standard deviation) across two runs of training-validation with different seeds for model initialization.

ing with a multitask setup wherein a conjunction 1 million synthetic numerical reasoning samples (165,000 of which are strict arithmetic) is used for numeric skill injection while an additional 1 million synthetic textual samples are used to avoid catastrophic forgetting of linguistics as a consequence of the non-linguistic skill injection. Please note, that for this experimentation, the *pre-trained GenBERT model has been used as-is, thus ensuring no performance degradation as a consequence of in-house replication.*

3.3 Quantitative Results

3.3.1 Numerical Reasoning

From Table 2, we observe that using only $\frac{1}{4}$th of the training dataset, Skill-LM closely resembles the performance of GenBERT while significantly improving the performance on out-of-domain numeral ranges. This leads to two deductions:

- It is known in the literature that LLMs often struggle to extrapolate numeral ranges that are absent from the training corpus (OOD) (Wallace et al., 2019; Razeghi et al., 2022). The significant improvement in quantitative reasoning in OOD numerals from Skill-LM (w/o $\mathcal{L}_{EWC}$) (row 3) establishes the vital role that skill-specific regression loss $\mathcal{L}_{REG}$ plays in not just learning the correct tokens to predict in response to a quantitative prompt, but capturing the magnitude of each numeral tokens in their representations.

- The significant jump in OOD improvement in addition to the increased in-domain performance from Skill-LM (row 4) suggests that $\mathcal{L}_{EWC}$ not only minimizes the loss of linguistic prowess, but also acts as a universal regularizer that prevents the model from over-fitting on the target task.

3.3.2 Natural Language Understanding

Recall that our goal with Skill-LM is to prevent the loss of linguistic prowess as a consequence of non-linguistic skill injection. The premise therein is that BERT-derivatives, empowered with non-linguistic skills, should at least strive to have linguistic performances of the base model. Thus, the performance of the base BERT model is *italicized* in Table 3.

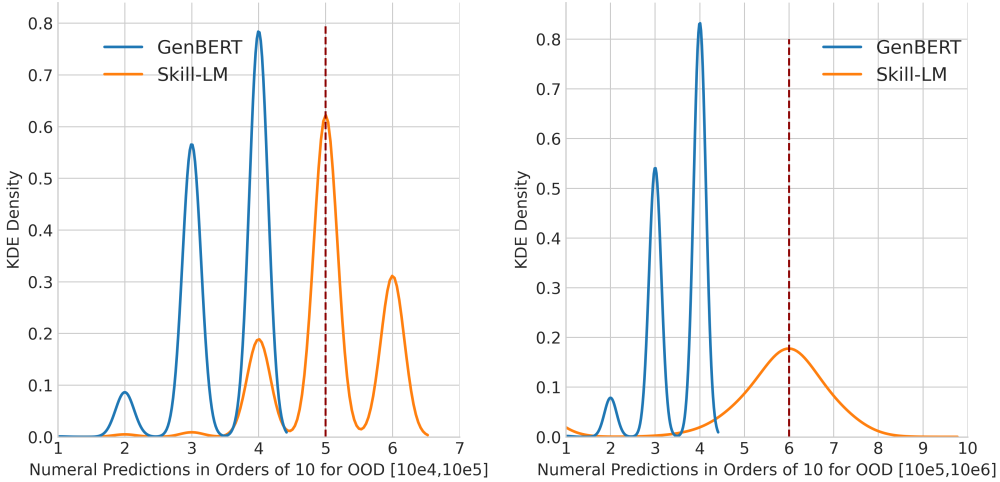


Figure 4: *Skill-LM consistently predicts the correct order of magnitude for the numerals:* From Table 2, Skill-LM significantly improves predictive performance for OOD range $[20^3, 10^4]$, however, all models see a drastic drop in performance for the OOD ranges $[10^4, 10^5]$ and $[10^5, 10^6]$. Although these models are unable to predict the *exact* numerals, the KDE plots above showcase how *close* these models are to predicting the correct order of magnitude for the numerals - $[10^4, 10^5]$ on the left and $[10^5, 10^6]$ on the right. Skill-LM consistently predicts the correct order of magnitude for the numerals as marked by the vertical dashed red line. This is evident from the fact that the largest mode of Skill-LM coincides with the correct order of magnitude (red line).

In §1.2, we established the degradation of linguistic performance in LLMs as a consequence of non-linguistic skill injection. Thus, the goal of weight consolidation $\mathcal{L}_{EWC}$ was to revitalize the linguistic performance of the model back to baseline. However, from Table 3, we observe that employing $\mathcal{L}_{EWC}$ that uses 0 additional training data outperforms GenBERT that uses 1 Million additional linguistic training data on 8/9 of the standardized GLUE benchmarks. To further authenticate these findings, the results are presented as μ_σ (mean and standard deviation) across two runs of training-validation with different seeds for model initialization. Thus Skill-LM showcases improved performance coupled with significant *savings in GPU compute costs* compared to previous related efforts that train on an additional 1 Million linguistic training samples (Geva et al., 2020).

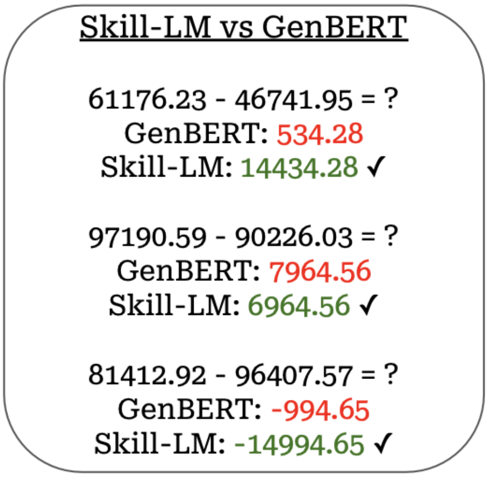
3.4 Qualitative Results

In §2.1, we theorized that regression loss, in the context of numerical skill injection, would inject a sense of numeric scale and magnitude estimation (Dehaene et al., 1998) to the general learning of numerical representations. From Table 2 we quantified the gains from this skill-specific loss in OOD generalization of numerals, however, in this section we further investigate whether the extrapolation to OOD numerals is indeed due to this learnt sense of numeric scale.

OOD numerals closer to the training range: In Table 2, Skill-LM boosts the predictive performance for OOD numerals in the range $[20^3, 10^4]$ from 1.32% to 19.44% - but where does the baseline fail? In Figure 5, as common-case failure scenarios, we showcase 3 sample responses from Skill-LM vs baseline to prompts from the OOD range $[20^3, 10^4]$: while the baseline does capture the nuances in difference of the operands (the numerals closer the decimal are correct), it severely fails to extrapolate to the scale of the operands.

OOD numerals further from the training range: In Table 2, the evaluation metric used is accuracy, thus evaluating the capabilities of these models to output the *exact* token in response to the quantitative reasoning prompts. For larger OOD ranges $[10^4, 10^5]$ and $[10^5, 10^6]$, all models struggle to predict the exact output - *but how close do they get?* Figure 4 showcase the distribution of the predicted output based on their powers of 10s - for the OOD range $[10^4, 10^5]$, the outputs should mostly center around 10^5 (left figure) while for the range $[10^5, 10^6]$ they should center around 10^6 (right figure). Although unable to predict the exact tokens, Skill-LM tends to predict tokens closer in magnitude to the ground truth consistently compared to our baseline.

Figure 5: For the OOD range $[20^3, 10^4]$ immediate to the training numeral range $[0, 20^3]$, this figure showcases, qualitatively, the predictive behaviors of Skill-LM vs GenBERT. Although GenBERT is able to capture the nuances in difference of the operands, it fails to extrapolate to the scale of the operands.



Skill-LM vs GenBERT

61176.23 - 46741.95 = ?
GenBERT: 534.28
Skill-LM: 14434.28 ✓

97190.59 - 90226.03 = ?
GenBERT: 7964.56
Skill-LM: 6964.56 ✓

81412.92 - 96407.57 = ?
GenBERT: -994.65
Skill-LM: -14994.65 ✓

4 Conclusions

Our study shows that LLMs are capable of demonstrating quantiative reasoning without sacrificing the broad palette of linguistic skills that they are traditionally evaluated against. This multi-task framework, together with the weight consolidation strategy, highlights that this framework can be systematized beyond the studies described here. As a result, non-linguistic tasks and linguistic tasks need not be seen as being at odds for LLMs and we can begin thinking about richer integrations of qualitative and quantitative reasoning. Our experimental results also highlight that the improvements showcased here do not require exorbitant training data and in fact require just a fraction of what previous studies have leveraged.

Our future work will be organized in three directions. First, we intend to study at a more fine-grained level the dovetailing of different arithmetic reasoning tasks vis-a-vis linguistic counterparts, and any synergies that can be exploited while learning. Second, there are situations where linguistics can help numerical reasoning (math word problems, data-to-text generation) and multi-task formulations that capture the underlying semantics can be developed. Finally, there are other forms of non-linguistic reasoning (diagrammatic reasoning) that can potentially be studied using the multi-task framework that we have described here.

Limitations

In our study, we address the issue of linguistic forgetting via the injection of the strict non-linguistic skill of quantitative reasoning. Although quantitative reasoning with LLMs is an active research area, as discussed above, further fine-grained studies are required to extrapolate this behavior to tasks that leverage synergies between aspects of both linguistics and non-linguistics - such as math word problems or data-to-text generation. Further, investigations into the linguistic forgetting tendencies of different languages would lend an insight into the role of linguistic morphology in this behavior. The restrictions from our in-house GPU resources does not allow scaling this study to more recent models that exceed 100 Billion parameters, although, due to the sharing of similar architectures, we forecast our findings to hold despite of model scaling.

Ethics Statement

Although the ethical waters of the development and deployment of LLMs are difficult to nagivate, we can ascertain that our study does not bring forth further complications. The datasets we use in this study are established benchmark datasets from publicly accessible websites and do not contain any personally identifiable information. Our analyses does not constitute human subjects and thus is not within the purview of the IRB. Further, in the landscape of increasing emission costs from large-scale computation, our study offers avenues for severely restricting the size of the training data - both linguistic and non-linguistic.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*.

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)*.

Stanislas Dehaene. 2003. The neural basis of the weber–fechner law: a logarithmic mental number line. *Trends in cognitive sciences*.

Stanislas Dehaene, Ghislaine Dehaene-Lambertz, and Laurent Cohen. 1998. Abstract representations of numbers in the animal and human brain. *Trends in neurosciences*.

Stanislas Dehaene, Emmanuel Dupoux, and Jacques Mehler. 1990. Is numerical comparison digital? analogical and symbolic effects in two-digit number comparison. *Journal of experimental Psychology: Human Perception and performance*.

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)*.

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

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. 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)*.

Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting numerical reasoning skills into language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Chengyue Jiang, Zhonglin Nian, Kaihao Guo, Shanbo Chu, Yinggong Zhao, Libin Shen, and Kewei Tu. 2020. Learning numeral embedding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Elana Joram, Lauren B Resnick, and Anthony J Gabriele. 1995. Numeracy as cultural practice: An examination of numbers in magazines for children, teenagers, and adults. *Journal for Research in Mathematics Education*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*.

Dharshan Kumaran, Demis Hassabis, and James L McClelland. 2016. What learning systems do intelligent agents need? complementary learning systems theory updated. *Trends in Cognitive Sciences*.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*.

Aitor Lewkowycz, Anders Johan Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Venkatesh Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. 2022. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems*.

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2022. On the advance of making language models better reasoners.

Aman Madaan, Ashish Mittal, Ganesh Ramakrishnan, Sunita Sarawagi, et al. 2016. Numerical relation extraction with minimal supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*.

OpenAI. 2022. Chat-gpt: Optimizing language models for dialogue.

Theodore M Porter. 1996. Trust in numbers. In *Trust in Numbers*. Princeton University Press.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

Yasaman Razeghi, Robert L Logan IV, Matt Gardner, and Sameer Singh. 2022. Impact of pretraining term frequencies on few-shot reasoning. *arXiv preprint arXiv:2202.07206*.

Jorma J Rissanen. 1996. Fisher information and stochastic complexity. *IEEE transactions on information theory*.

Mandar Sharma, John S Brownstein, and Naren Ramakrishnan. 2021. T 3: Domain-agnostic neural time-series narration. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 1324–1329. IEEE.

Mandar Sharma, Ajay Gogineni, and Naren Ramakrishnan. 2022a. Innovations in neural data-to-text generation. *arXiv preprint arXiv:2207.12571*.

Mandar Sharma, Nikhil Muralidhar, and Naren Ramakrishnan. 2022b. Overcoming barriers to skill injection in language modeling: Case study in arithmetic. *36th Conference on Neural Information Processing Systems (NeurIPS 2022) Workshop on Math-AI*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*.

Georgios Spithourakis and Sebastian Riedel. 2018. Numeracy for language models: Evaluating and improving their ability to predict numbers. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Avijit Thawani, Jay Pujara, Filip Ilievski, and Pedro Szekely. 2021. Representing numbers in nlp: a survey and a vision. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do nlp models know numbers? probing numeracy in embeddings. 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)*.

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*.

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

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)*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing.

Xikun Zhang, Deepak Ramachandran, Ian Tenney, Yanai Elazar, and Dan Roth. 2020. Do language embeddings capture scales? In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

A Appendix

A.1 Substantial Forgetting on the Basis of Parameter Sharing

A.1.1 Fisher Information Computation

The Fisher information score, as depicted in (1) is the expected value of the square of the gradient for a sample $y \in Y$. Thus, to compute the Fisher sensitivity of a model θ to a task A, we compute the sum of the squared gradients averaged by the number of parameters in θ. In our case, where θ is a pretrained transformer-based LLM, the model cross-entropy loss $dlogf(y;\theta)$ (4) for each sample y is computed, through which the gradient $\frac{dlogf(y;\theta)}{d\theta}$ can then be computed. The sum of these squared gradients gives us the Fisher information score for each parameter θ_i in a model θ with respect to a task A.

A.1.2 Parameter Sensitivities for the Self-Attention Encoder Layers

In §1.2, we substantiated the linguistic forgetting of LLMs through parameter sharing tendencies of the model with illustrations of the parameter sensitivities across different tasks for the first (1st) and last (12th) self-attention encoder layer of the transformer. Here, through Figure 6, we show that the findings hold across all self-attention encoder layers of model. Further, it is interesting to observe that the task CoLA shares more parameters with the Arithmetic task in the earlier layers compared to the latter layers.

A.2 Designing Skill-LM

A.2.1 Hyperparameterization for $\mathcal{L}_{REG}$

The intuition for the selection of hyperparameter λ_1 within the range $\{1e^{-3}, 1e^{-4}\}$ was to scale-match the exceedingly large values of regression loss $\mathcal{L}_{REG}$ to the cross-entropy loss $\mathcal{L}_{CE}$ during the intial phases of training where incorrect predictions of target numerals are frequent. In addition to evaluating the model convergence with λ_1 set to these constants, we also evaluate the following update schedule configurations for λ_1:

Algorithm 1 Update Schedule 1

```
$\lambda_{prev} \leftarrow 1e^{-4}$
for i in epochs do
    $\lambda_{current} \leftarrow \frac{\mathcal{L}_{REG}}{\mathcal{L}_{CE}+\mathcal{L}_{REG}}$
    $\lambda_1 \leftarrow 0.99 * \lambda_{prev} + 0.01 * \lambda_{current}$
    $\lambda_{prev} \leftarrow \lambda_1$
end for
```

Algorithm 2 Update Schedule 2

```
$\lambda_{prev} \leftarrow 1e^{-4}$
for i in epochs do
    $\lambda_{current} \leftarrow \frac{\mathcal{L}_{REG}}{\mathcal{L}_{CE}+\mathcal{L}_{REG}}$
    $\lambda_1 \leftarrow 0.01 * \lambda_{prev} + 0.99 * \lambda_{current}$
    $\lambda_{prev} \leftarrow \lambda_1$
end for
```

A.2.2 Model Training Configurations

The models BERT$_{Arith}$, GenBERT, and Skill-LM all share the base BERT architecture. The baseline GenBERT has been employed as-is with the model that the authors provide used for comparative evaluation. For models BERT$_{Arith}$ and Skill-LM, these are initialized as pre-trained base BERT models with 160M parameters and further trained on randomly sampled $\frac{n}{4}$th of the arithmetic portion of GenBERT's training data. The pre-trained base BERT model is loaded from the HuggingFace library (Wolf et al., 2019).

The scheme for training follows BERT's standard training protocol of using masked-language modeling. However, instead of randomly masking 15% of the tokens as done in BERT, we mask the result of the each sample quantitative prompt. For instance, from Figure 5, for the sample *61176.23 - 46741.95 = 14434.28*, the models BERT$_{Arith}$ and Skill-LM are trained to predict 14434.28 for the masked prompt *61176.23 - 46741.95 = [MASK]*. With the standard sequence size of 512 for BERT, the models were trained for 60 epochs in a cluster of 4 Tesla P100 GPUs.

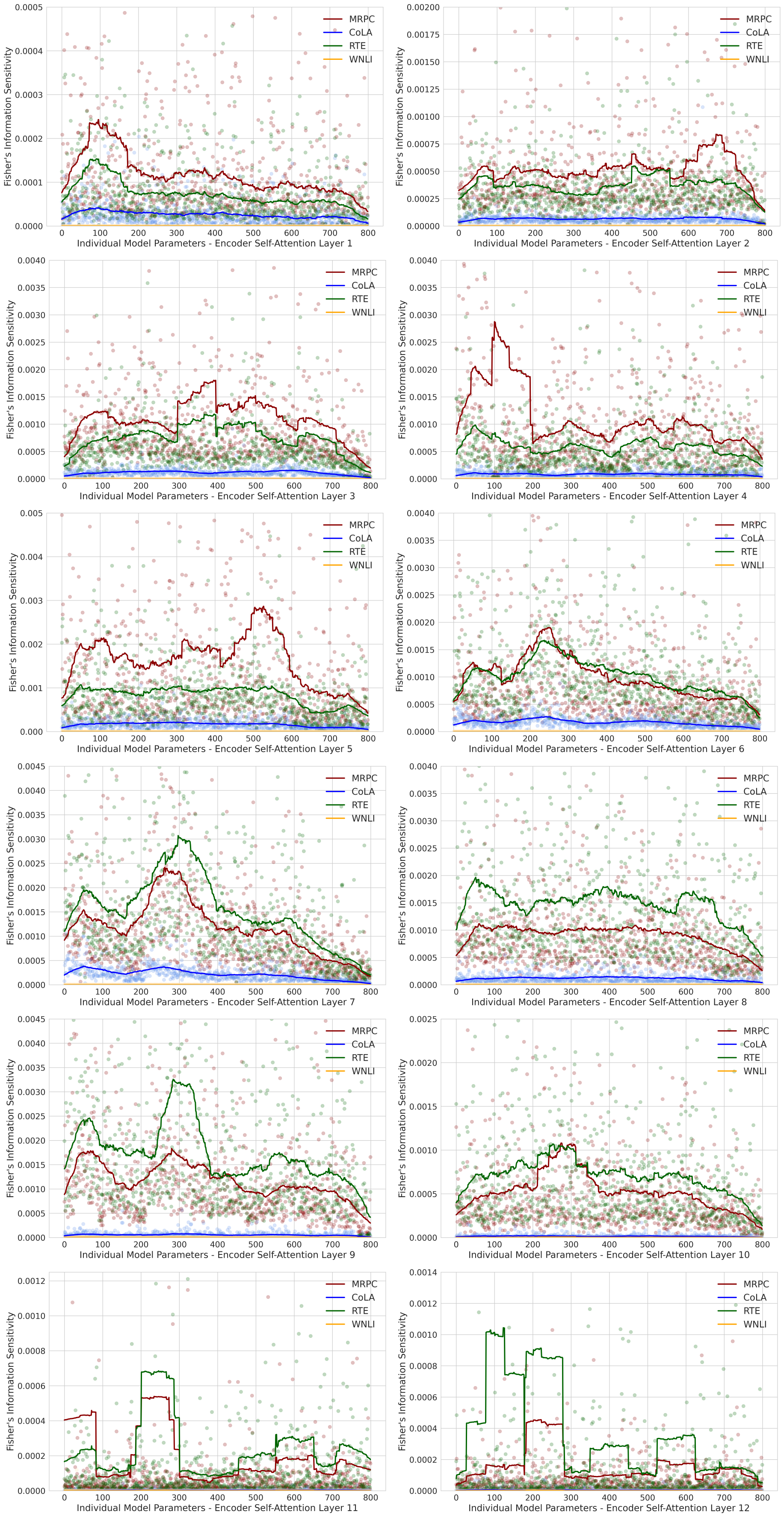


Figure 6: Given Fisher parameter sensitivities $I(\theta)$ the self-attention encoder layers for four different models based on continued training of the base BERT model on four datasets: $I_{arith}(\theta)$ on an arithmetic reasoning and $I_{CoLA}(\theta)$, $I_{MRPC}(\theta)$, $I_{RTE}(\theta)$ on GLUE tasks CoLA, MRPC, and RTE respectively, this plot takes the $n = 800$ most crucial parameters based on $I_{arith}(\theta)$ and showcases how sensitive those *same* parameters are to the GLUE tasks based on $I_{CoLA}(\theta)$, $I_{MRPC}(\theta)$, and $I_{RTE}(\theta)$.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitations

□ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract, Introduction (Section 1)

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Introduction (Section 1) , Experiments (Section 3)

☑ B1. Did you cite the creators of artifacts you used?
Appendix

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

□ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Ethics Statement

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Experiments (Section 3) and Appendix

C ☑ Did you run computational experiments?

Introduction (Section 1), Experiments (Section 3)

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Designing Skill-LM (Section 2)

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Experimentation (Section 3)

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Appendix

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?
Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.