SPARSITY MAY CRY: LET US FAIL (CURRENT) SPARSE NEURAL NETWORKS TOGETHER!

Shiwei Liu[1]*, **Tianlong Chen**[1]*, **Zhenyu Zhang**[1], **Xuxi Chen**[1], **Tianjin Huang**[2], **Ajay Jaiswal**[1], **Zhangyang Wang**[1]
[1]University of Texas at Austin [2]Eindhoven University of Technology
`shiwei.liu@austin.utexas.edu; t.huang@tue.nl`
`{tianlong.chen,zhenyu.zhang,xxchen,ajayjaiswal,atlaswang}@utexas.edu`

ABSTRACT

Sparse Neural Networks (SNNs) have received voluminous attention predominantly due to growing computational and memory footprints of consistently exploding parameter count in large-scale models. Similar to their dense counterparts, recent SNNs generalize just as well and are equipped with numerous favorable benefits (e.g., low complexity, high scalability, and robustness), sometimes even better than the original dense networks. As research effort is focused on developing increasingly sophisticated sparse algorithms, it is startling that a *comprehensive benchmark to evaluate the effectiveness* of these algorithms has been highly overlooked. In absence of a carefully crafted evaluation benchmark, most if not all, sparse algorithms are evaluated against fairly simple and naive tasks (eg. CIFAR-10/100, ImageNet, GLUE, etc.), which can potentially *camouflage* many advantages as well unexpected predicaments of SNNs. In pursuit of a more general evaluation and unveiling the true potential of sparse algorithms, we introduce "**Sparsity May Cry**" **Benchmark (SMC-Bench)**, a collection of carefully-curated 4 diverse tasks with 10 datasets, that accounts for capturing a wide range of domain-specific and sophisticated knowledge. Our systemic evaluation of the most representative sparse algorithms reveals an important obscured observation: *the state-of-the-art magnitude- and/or gradient-based sparse algorithms seemingly fail to perform on SMC-Bench* when applied out-of-the-box, sometimes at significantly trivial sparsity as low as 5%. The observations seek the immediate attention of the sparsity research community to reconsider the highly proclaimed benefits of SNNs. We further conduct a thorough investigation into the reasons for the failure of common SNNs. Our analysis points out that such failure is intimately related to the "lazy regime" of large model training, which hints us with stronger pruning recipes that alleviate the failure on SMC-Bench (though still more or less suffering). By incorporating these well-thought and diverse tasks, SMC-Bench is designed to favor and encourage the development of more scalable and generalizable sparse algorithms. We open-source SMC-Bench to assist researchers in building next-generation sparse algorithms that scale and generalize: `https://github.com/VITA-Group/SMC-Bench`.

1 INTRODUCTION

Sparse Neural Networks (SNNs) are no stranger to the deep learning community (Liu & Wang, 2023), but recently they have received stupendous attention in the era of transformers (eg. BERT, GPT, ViT, CLIP, etc.), when the parameter count is frequently measured in billions rather than millions. Due to the consistent efforts of sparsity researchers, SNNs have ushered enormous breakthroughs and can generalize just as well as original dense networks, and it is feasible to procure them after training (Frankle & Carbin, 2019; Sanh et al., 2020; Chen et al., 2020; Frankle et al., 2020), during training (Zhu & Gupta, 2017; Gale et al., 2019; Liu et al., 2021b), and even before training (Mocanu et al., 2018; Lee et al., 2019; Liu et al., 2022) their dense counterparts using pruning. Apart from well-known efficiency benefits, surprisingly, SNNs also enjoy auxiliary benefits such as adversarial robustness (Guo et al., 2018; Özdenizci & Legenstein, 2021; Chen et al., 2022), out-of-distribution

*These authors contributed equally to this work.

generalization (Zhang et al., 2021; Diffenderfer et al., 2021), and uncertainty estimation (Liu et al., 2021a), etc. Despite the multi-dimensional success of numerous sparse algorithms, startlingly, our extensive survey across over 100 recent SNN papers within `2015-2022` unveils multiple daunting issues regarding evaluation datasets and protocols blindly followed within the sparsity community, that may significantly impede future progress if left unacknowledged.

Issues with current evaluation paradigm: *Firstly*, the vast majority of current work on SNNs is *narrowly evaluated*, i.e., only targeting a single or a few tasks (usually on image classification and sometimes on language understanding) on which SNNs have already proven their proficiency (Gale et al., 2019; Frankle & Carbin, 2019). Surprisingly, 79 papers out of our carefully selected 100 papers on SNNs, evaluate sparse models *merely* on a single task, where 72 out of them evaluate image classification. *Secondly*, people are obsessed with evaluating SNNs on well-understood datasets, including but not limited to MNIST (LeCun, 1998) (26 papers), CIFAR-10/100 (Krizhevsky et al., 2009) (59 and 37 papers, respectively), ImageNet (Deng et al., 2009) (62 papers), and GLUE (Wang et al., 2018) (9 papers), where deep neural networks have already exceeded the human-equivalent performance (refer to Appendix D for more details). For instance, even though ImageNet has been considered a rather challenging task over years, very high accuracy (>90%) has been reported many times (Yu et al., 2022; Wortsman et al., 2022; Zhai et al., 2022). Such relatively restricted evaluations with "nearly saturated performance" limit the scope of sparse neural networks and are potentially ill-suited to identify new and unexpected capabilities of SNNs.

Addressing the aforementioned limitations of current SNN evaluation protocols is a pressing need for the community. To this end, we assemble a large-scale, fairly arduous, and diverse benchmark for sparse neural networks - "**Sparsity May Cry**" **Benchmark** (or briefly **SMC-Bench**). Specifically, we consider a broad set of tasks including *complex reasoning, multilingual translation, and protein prediction*, whose content spans multiple domains. Those tasks require a vast amount of commonsense knowledge, solid mathematical and scientific backgrounds to solve even for humans. Note that none of the datasets in SMC-Bench was created from scratch for the benchmark, we rely on pre-existing datasets as they have been agreed by researchers as challenging, interesting, and of high practical value. We rigorously measure the performance of state-of-the-art (SOTA) pruning and sparse training approaches (in their most common, basic settings) on SMC-Bench, to understand the potential of SNNs to scale and generalize. Our key observations and contributions can be unfolded as:

- We present "**Sparsity May Cry**" Benchmark, to **re-define** the evaluation protocols for sparse neural networks and facilitate a comprehensive assessment of SOTA sparse algorithms. The premise of SMC-bench is to develop a suite of large-scale, challenging, realistic, and diverse tasks and datasets that can empower the rigorous advancements in the community.
- SMC-Bench unveils a critical and startling observation - all of the SOTA sparse algorithms seem to fail on SMC-Bench "out-of-the-box", sometimes at significantly trivial sparsity *e.g.,* 5%. Note that the failure does not appear specific to one sparsification approach but unanimously across all approaches we evaluated. This observation *alarmingly* demands the attention of the sparsity community to reconsider the highly proclaimed benefits of SNNs.
- We conduct extensive experiments across representative SNNs produced by various SOTA pruning and sparse training approaches on SMC-Bench, and we summarize our findings: ❶ Model prunability is intimately related to task difficulty: models trained on difficult tasks suffer more from pruning compared to easier tasks; ❷ The success of before-training sparsification (sparse training or pruning at initialization) is hard to generalize in more complex scenarios; ❸ Iterative magnitude pruning (IMP) does not necessarily generalize better than one-shot pruning (OMP) or during-training pruning; ❹ Despite performance difference, different magnitude-based pruning approaches lead to extremely similar layerwise sprasities.
- We further carry out a comprehensive investigation into the potential causes of SNN failures on SMC-Bench. Our analysis suggests that the failure of the existing sparse algorithms might be due to the "lazy regime" dynamics emerging in sufficiently overparameterized models (Chizat et al., 2019; Malladi et al., 2022). Inspired by this finding, we hypothesize and confirm that the second-order pruning approaches, i.e., oBERT (Kurtic et al., 2022), are more reliable pruning approaches for SMC-Bench, which yield relatively more promising performance on SMC-Bench in Appendix C.

2 Related Work

2.1 Advances in Sparse Neural Networks

Post-Training. SNNs refer to a neural network where a certain portion of its components (e.g., weights, neurons, filters, and attention heads) have exactly zero values. The initial purpose of SNNs is retrospectively to accelerate model at inference time (a.k.a., post-training sparsification; Mozer & Smolensky (1989); LeCun et al. (1990)). Thanks to the over-parameterization property of deep neural networks, we can dramatically prune deep neural networks to smaller sizes with marginal loss of performance. Post-training sparsification has been well studied and results in various mature criteria that can be generally categorized into zero-order methods (magnitude-based; Han et al. (2015)), first-order methods (gradient-based; Molchanov et al. (2016); Sanh et al. (2020); Jiang et al. (2021)), and second-order methods (Hessian-based; LeCun et al. (1990); Hassibi & Stork (1992); Dong et al. (2017)). Second-order sparsification usually achieves higher performance than the other two but is also more expensive due to the full Hessian calculation. Fortunately, many approaches have been proposed to efficiently approximate Hessian (Zeng & Urtasun, 2018; Wang et al., 2019; Singh & Alistarh, 2020). The Lottery Ticket Hypothesis (LTH) adopts iterative magnitude pruning (IMP) on fully trained networks and finds a subnetwork at initialization that can be re-trained in isolation to match the original dense networks. Renda et al. (2020) further found that instead of re-training with the initial weights, re-training with the final weights achieves better performance. With the rise of large language models (LLMs), newer post-training pruning methods have emerged which aim to improve the affordability of these models (Sanh et al., 2020; Chen et al., 2020; Zafrir et al., 2021; Kurtic et al., 2022; Xu et al., 2021; Lagunas et al., 2021; Zhang et al., 2022; Frantar et al., 2021).

During-Training. During-training sparsification (Finnoff et al., 1993) is a cheaper option, compared to sparsify a fully converged model. Approaches of during-training sparsification usually train a dense network for some time and then gradually sparsify the network with some schedules and end up with a sparse model with target sparsities. Zhu & Gupta (2017); Gale et al. (2019); Lin et al. (2020); Liu et al. (2021b) are highlight approaches that gradually prune networks during training and meanwhile allow the pruned weights to be reactivated in case of inaccurate pruning. Another direction of during-training sparsification is adding sparsifying penalties such as (grouped) L_0 and L_1 norm to the loss function, which will punish the unimportant weights to zero, leading to sparse weights (Louizos et al., 2018; Luo & Wu, 2020; Savarese et al., 2020).

Prior-Training. Recently, foundation models (Brown et al., 2020; Chowdhery et al., 2022; Ramesh et al., 2022) have demonstrated promising quantitative improvement and new qualitative capabilities with increasing scale (Zhang et al., 2020b). Along with the scaling of model size and data size, the training resources of these foundation models also get outrageous. To accelerate training, we need to sparsify models before training. LTH unveils the possibility to find SNNs at initialization that can match their dense counterparts, even though it uses post-training pruning to find them. At the same time, sparse training (Mocanu et al., 2018; Mostafa & Wang, 2019; Dettmers & Zettlemoyer, 2019; Evci et al., 2020; Liu et al., 2021c; Schwarz et al., 2021) was proposed that can train a randomly-initialized sparse neural network from scratch while dynamically optimizing the sparse connectivity with promising performance. Instead of randomly initializing sparse networks, one iteration (Lee et al., 2019; Wang et al., 2020) or a few iterations (Tanaka et al., 2020; de Jorge et al., 2021) of training can be utilized to guide the search for sparse networks before training.

2.2 Benchmarking in Sparse Neural Networks

Gale et al. (2019) rigorously evaluated variational dropout (Molchanov et al., 2017), l_0 regularizaion (Louizos et al., 2018), and GMP (Zhu & Gupta, 2017) on two large-scale tasks. They demonstrated that the appealing performance advantages of variational dropout and l_0 regularization cannot generalize to large-scale tasks whereas simple magnitude pruning performs surprisingly well. Liu et al. (2018) examined two pipelines: training from scratch and fine-tuning, concluding that fine-tuning a pruned model only gives comparable or worse performance than training from scratch. Blalock et al. (2020) provided a comprehensive literature review on SNNs and found that pruning papers rarely make direct and controlled comparisons. Frankle et al. (2021) assessed the efficacy of various pruning at initialization approaches and attribute their inferior performance to their insensitivity to weight shuffling and re-initialization. Liu et al. (2022) re-evaluated the performance of various random pruning before training and found that sparsely training a randomly pruned network from scratch can surprisingly match the performance of its dense equivalent. These papers shed light on the behavior of SNNs and discover important research problems for future work.

3 SMC-BENCH

SMC-Bench is crafted for evaluating if all proclaimed benefits of SNNs can "scale and generalize". It consists of 4 diverse and difficult tasks, including commonsense reasoning, arithmetic reasoning, multilingual translation, and protein prediction, with 10 datasets collected from prior work and open-source GitHub repositories. To investigate if there is a strong correlation between model prunability and task difficulty, we choose multiple datasets with different degrees of difficulty.

3.1 COMMONSENSE REASONING

Commonsense reasoning task asks commonsense questions about the world involving complex semantics that often require rich common sense and background knowledge. We consider three commonly used datasets for commonsense reasoning. (1) **RACE** (Lai et al., 2017) contains near 28,000 passages and 100,000 questions collected from the English exams for Chinese students in middle (RACE-M) and high school (RACE-H). (2) **WinoGrande** (Sakaguchi et al., 2021) is a modified version of the Winograd Schema Challenge (WSC) benchmark (Levesque et al., 2012) with enhanced scale and hardness, containing 44k problems. (3) **Commonsense QA (CSQA)** (Talmor et al., 2018) is a challenging dataset containing 12,247 multiple-choice questions where one source concept and three target concepts are first extracted from ConceptNet (Speer et al., 2017) based on which the Crowd-works are asked to author multiple-choice questions with two additional distractors. In general, CSQA is harder than WinoGrande and RACE, with ceiling human performance of 89%, 94%, and 95%, respectively.

3.2 ARITHMETIC REASONING

Arithmetic reasoning poses a question of a math problem and the model is asked to generate a mathematical equation that can be evaluated to get the answer. We consider the following three math word problem datasets: (1) the widely used **MAWPS** benchmark (Koncel-Kedziorski et al., 2016) composed of 2,373 problems; (2) the arithmetic subset of ASDiv (Miao et al., 2021) - **ASDiv-A** with 1,218 math problems; (3) the more challenging **SVAMP** (Patel et al., 2021) dataset which is created by applying complex types of variations to the samples from ASDiv-A. The task difficulty monotonically increases from MAWPS to ASDiv-A, and to SVAMP.

3.3 PROTEIN THERMOSTABILITY PREDICTION

Maintaining a stable 3D structure is an essential pre-condition for protein to function correctly in biological phenomena. Numerous efforts are devoted to modeling and predicting protein's stability against pH, salinity, and temperature. We consider the tasks of protein thermostability prediction on two representative datasets: (1) **HotProtein** (Chen et al., 2023) is recently proposed as a large-scale, standardized protein benchmark with organism-level temperature annotations, which contains 182K protein sequences and 3K folded structure from 230 different species. Three dataset variants, **HP-S**, **HP-S^{2}C**5, and **HP-S^{2}C**2, are adopted to examine sequence- and structure-based methods, respectively. **HP-S** has $\{6,390,\ 3,4946,\ 30,333,\ 79,087,\ 31,549\}$ protein sequences from five categories of {*Cryophilic*, *Psychrophilic*, *Mesophilic*, *Thermophilic*, *Hyperthermophilic*}; **HP-S^{2}C**5 has both sequences and structures for $\{73,\ 387,\ 195,\ 196,\ 189\}$ proteins from the same five classes ordered from *Cryophilic* to *Hyperthermophilic*; **HP-S^{2}C**2 has both sequences and structures for $\{1,026,\ 939\}$ proteins from {"hot" ($\geq 45°\mathrm{C}$), "cold" ($< 45°\mathrm{C}$)} two classes. (2) **Meltome Atlas** (Jarzab et al., 2020) is another challenging test bed for protein's thermostability. It has $\{7,902,\ 15,833,\ 10,518\}$ protein sequences from three of the five aforementioned classes, from *Mesophilic* to *Hyperthermophilic*. All samples are annotated with their melting temperature.

3.4 MULTILINGUAL TRANSLATION

Multilingual translation processes multiple languages using a single language model and requires the model to have the ability to perform translation across languages. We follow Liu et al. (2020); Tang et al. (2020) and choose 10 English-centric language pairs (Fr, Cs, De, Gu, Ja, My, Ro, Ru, Vi, Zh $\leftrightarrow$ En) from an open source parallel corpus - OPUS (OPU, 2020). We follow Arivazhagan et al. (2019) and use pivot data through English to create 3 Many-to-Many multilingual translation fine-tuning settings including 2-to-2 (Fr, Cs), 5-to-5 (Fr, Cs, De, Gi, Ja), and 10-to-10.

4 EVALUATION ON SMC-BENCH

Models. Despite we are aware that performing few-shot prompting on large-scale pre-training language models with billions of parameters is able to solve these tasks (Wei et al., 2022; Srivastava et al., 2022), we choose to focus on fine-tuning or training with pre-trained mid-scale models with millions of parameters, to improve the accessibility of our Benchmark. Specifically, we choose to fine-tune the popular RoBERTa (Liu et al., 2019) for commonsense reasoning; to fine-tune mBART (Liu et al., 2020) for multilingual translation; to train GTS (Xie & Sun, 2019) and Graph2Tree (Zhang et al., 2020a) with RoBERTa's pre-trained embedding for arithmetic reasoning; to fine-tune Transformer-based (Vaswani et al., 2017) for protein property prediction. See Appendix A for full details.

Sparse Neural Networks. We select the most representative magnitude- and/or gradient-based approaches where the prune operation is performed before, during, or after training. Formally, given a dense network $\theta_l \in \mathbb{R}^{d_l}$ with a dimension of d_l in each layer $l \in \{1, \dots, L\}$, pruning generates binary masks $m_l \in \{0,1\}^{d_l}$ yielding sparse neural networks with sparse weights $\theta_l \odot m_l$. The sparsity level is the fraction of the weights that are zero-valued, calculated as $s = 1 - \frac{\sum_l m_l}{\sum_l d_l}$. Following a mainstream convention in many sparse training papers (Frankle & Carbin, 2019; Gale et al., 2019; Evci et al., 2020; Lee et al., 2019; Liu et al., 2021c), we sparsify most layers in the model including embedding layers and classifier heads, and we do not apply advanced techniques such as Iterative Learning Rate Rewinding (Renda et al., 2020) and Knowledge Distillation (Hinton et al., 2015) in our main evaluations, even if we observe that they help to alleviate accuracy drops as in Appendix C.

• *Lottery Ticket Hypothesis (LTH)* (Frankle & Carbin, 2019) is a strong post-training pruning baseline that iteratively adopts magnitude pruning after training to produce binary masks and re-train together with weights from step t. We set $t = 0$ in this paper, since rewinding to early training does not notably improve the performance of Transformer models (e.g., BERT) for downstream tasks (Chen et al., 2020). The pruning rate of each IMP is set as 20%.

• *Magnitude After Training* is a strong baseline for prune after training, which has demonstrated strong results in various regimes. After training or fine-tuning models on the specific task, we prune the model with one-shot magnitude pruning and re-train it with the full learning rate schedule from the beginning, dubbed "OMP (After)" in our experiments.

• *Random After Training* (Mittal et al., 2019) is the most naive baseline for post-training pruning. It uniformly samples a random score $s_l \in \text{Uniform}(0, 1)$ for each weight and prunes the weights with the lowest scores. After pruning, we also re-train with the entire training schedule.

• *Gradual Magnitude Pruning (GMP)* (Zhu & Gupta, 2017; Gale et al., 2019) gradually sparsifies networks during training according to a pre-defined sparsification schedule with sorting-based weight thresholding. The starting and the ending iteration of the gradual sparsification process are set as 10% and 80% of the entire training iterations. The frequency of sparsification steps is tuned among 500, 1000, and 4000, depending on the specific tasks. While we are aware of the advanced gradual pruning methods - movement pruning (Sanh et al., 2020), it usually exceeds GMP only at high sparsities (e.g., $>$90%), which is interesting but not within the scope of this paper.

• *Magnitude Before Training* (Frankle et al., 2021) simply removes weights with the lowest magnitude at initialization. Since we inherit weights from pre-trained models, the initial weights actually refer to the weights that are learned on the pre-trained tasks. We abbreviate this approach to "OMP (Before)" as we use one-shot magnitude pruning.

• *Random Before Training* (Liu et al., 2022) is the most naive baseline for prior-training pruning. We randomly sample scores for each weight and removes the weights with the lowest scores. Different from Random After Training, the pruning operation is performed before fine-tuning.

• *SNIP* (Lee et al., 2019) is a prior-training pruning technique that globally removes weights with the lowest connection sensitivity score $|g \odot w|$. SNIP is a strong baseline that consistently performs well among various prior-training approaches (Frankle et al., 2021).

• *Rigging the Lottery (RigL)* (Evci et al., 2020) is a leading sparse training method that updates the topology of sparse neural networks during training via a prune-and-grow scheme. To evaluate its effectiveness on downstream fine-tuning, we combine RigL with the other three prior-training methods. The update interval of RigL is set the same as the ones used for updating sparsity in GMP, following Liu et al. (2021b).

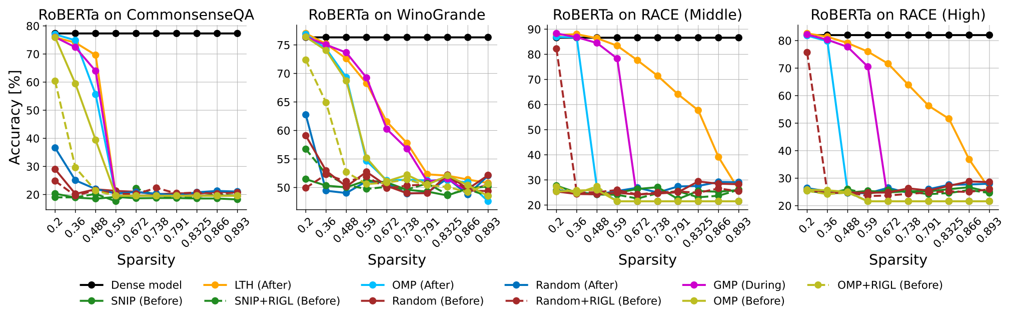

Figure 1: Commonsense reasoning performance of various sparse RoBERTa.

4.1 Commonsense Reasoning

Implementation Details. We follow the training settings of sequence modeling toolkit Fairseq (Ott et al., 2019) and fine-tune the pre-trained RoBERTa on our datasets with a standard cross-entropy loss. Specifically for each question, we also construct five inputs, one for each of the five candidate answer choices. Each input is constructed by concatenating the question and candidate answer together. We then encode each input and pass the resulting "[CLS]" representations through a classifier to predict the correct answer. All models are trained with the Adam (Kingma & Ba, 2014) optimizer with a learning rate of 1×10^{-5} using an A100 GPU. For CSQA, we train the model for 3000 steps with a linear warmup of 150 steps and a batch size of 16. The dropout rate is set as 0.1. This gives us a test accuracy of 77.3% with dense RoBERTa. For RACE, we train each model for 3 epochs with a batch size of 16. This gives us 86.6% and 82.0% dense accuracy on RACE (H) and RACE (M), matching the ones reported in Fairseq (86.5% and 81.3%). Models on WinoGrande are trained for $23,750$ steps with $2,735$ warmup steps and 32 batch size, reaching a 76.3% accuracy.

Results and Analyses. The results of various sparse neural networks are demonstrated in Figure 1. We summarize our main observations below:

① *All sparse algorithms seemingly fail to find matching SNNs, even at trivial sparsities such as 36%.* While several methods maintain the dense performance at 20% sparsity, their performance starts to drop significantly after that, and will undergo a catastrophic failure as the sparsity continues increasing. It is difficult even for the top-performing LTH to maintain the matching performance after the 2^{rd} IMP iteration. This is in stark contrast with the behavior of SNNs on the image classification task, where LTH can gracefully preserve the matching performance even at very extreme sparsities (>95% on CIFAR-10/100 (Yin et al., 2022) and >80% on ImageNet (Renda et al., 2020)).

② *The quality of SNNs on harder tasks suffers more from sparsity.* Models trained on the hardest task, CSQA, undergo a larger accuracy loss at the same sparsity than the other two datasets. For instance, all the SNNs on CSQA suffer from a catastrophic accuracy drop (up to 74%) and become no better than the random prediction at 59% sparsity. Meanwhile, when trained on WinoGrande and RACE at 59% sparsity, two sparse algorithms (LTH and GMP) can maintain relatively good performance with a smaller performance loss (i.e., 3% $\sim$ 10%).

③ *Post-training pruning consistently outperforms prior-training pruning.* LTH achieves the best performance across datasets, GMP performs well, and OMP (After) follows behind. However, prior-training pruning achieves worse performance. OMP (Before) performs closely behind OMP (After), whereas SNIP performs no better than the naive random pruning. After digging deeper into the case of SNIP, we find SNIP aggressively prunes the embedding layers to more than 99% sparsity even with a mild overall sparsity of 20%. Surprisingly, the leading dynamic sparsity approach RigL does not bring significant gains to these prior-training approaches, and sometimes even hurts the performance.

4.2 Arithmetic Reasoning

Implementation Details. We follow SVAMP (Patel et al., 2021) and choose the two top-performing tree-based models for arithmetic reasoning: GTS (Xie & Sun, 2019) and Graph2Tree (Zhang et al., 2020a). Graph2Tree in general performs slightly better than GTS. GTS adopts LSTM to encode input sequences and a tree-based Decoder to generate questions. Graph2Tree uses a graph transformer to learn the latent quantity representations from data, and a tree structure decoder to generate a solution expression tree. We follow exactly the training settings of Patel et al. (2021). The embedding weights are inherited from THE pre-trained RoBERTa. All models are trained with Adam for 50 epochs.

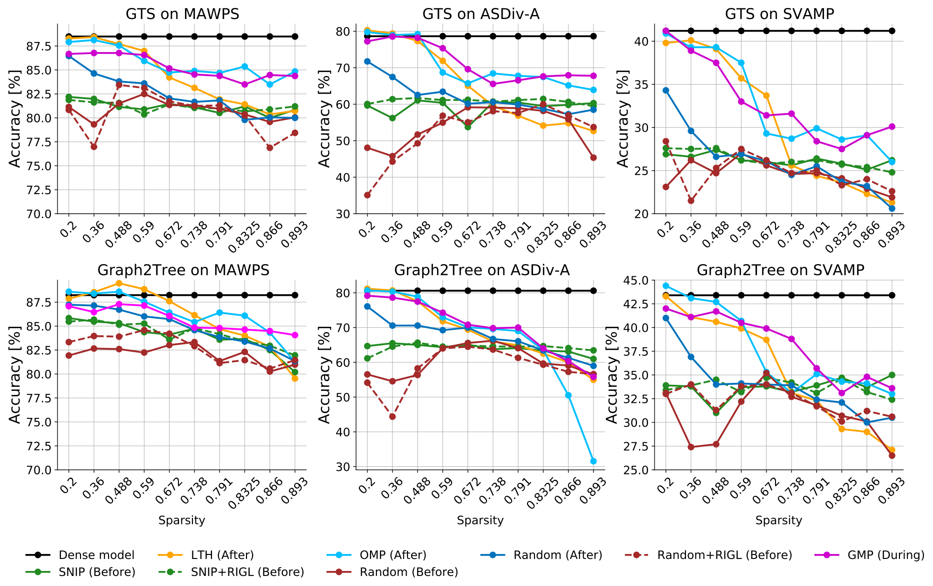

Figure 2: Arithmetic reasoning performance of various sparse GTS and Graph2Tree.

On MAWPS and ASDiv-A, models are trained with the training data and then evaluated on 5-fold cross-validation based on pre-assigned splits. For SVAMP, we train the models on a combination of MAWPS and ASDiv-A and test them on SVAMP, following Patel et al. (2021).

Results and Analyses. The performance on arithmetic reasoning is reported in Figure 2. The overall performance trend is very similar to the commonsense reasoning: SNNs can only match the dense performance when the sparsity level is lower than 48.8% with the exception of Graph2Tree on the relatively simple MAWPS dataset whose failing sparsity is 59%; SNNs are prone to sacrifice more performance on harder datasets (i.e., ASDiv-A and SVAMP) than the easier MAWPS dataset; prior-training methods perform no differently than random pruning. Moreover, we want to highlight that LTH surprisingly reaches lower accuracy than OMP and GMP at high sparsity levels, indicating that iterative magnitude pruning may not necessarily generalize better than on more complex tasks. Moreover, Magnitude Before Training (OMG (Before)) consistently causes severe layer collapse in the non-embedding layers, leading to zero accuracies. Since including the results of OMG (Before) will significantly dilute the performance difference of different sparsification methods, we choose to report it in Appendix B.

4.3 Protein Thermal Stability Prediction

4.3.1 Sequence-Based Models

Implementation Details. We examine two classic sequence-based approaches in protein property prediction, *i.e.*, TAPE (Rao et al., 2019) and ESM-1B (Rives et al., 2021). For TAPE, we fine-tune it from the official pre-training (Rao et al., 2019) for 4 epochs with an initial learning rate of 1×10^{-4} and a linear decay scheduler together with 100 warm-up steps. As for ESM-1B (Rives et al., 2021), starting from the official pre-trained checkpoint, we fine-tune the backbone with a learning rate of 1×10^{-6} and the linear classification head on top of the backbone with a learning rate of 2×10^{-2}. The learning rate schedulers used for both backbone and linear head are OneCycle (Smith & Topin, 2019) decay schedulers. The training batch size is 2 for Meltome Atlas and 3 for HotProtein (HP-S). Classification accuracy on test sets is collected to measure the model performance.

Results and Analyses. In this section, we examine diverse sparse neural networks of sequence-based models (*i.e.*, transformers) on protein thermostability prediction benchmarks. ESM-1B (Rives et al., 2021), a SOTA approach in protein property modeling, is evaluated on both HotProtein (HP-S) and Meltome Atlas datasets. TAPE (Rao et al., 2019) is a classic transformer-based model adopted on HotProtein (HP-S). Extensive results of both static and dynamic sparsifications are collected in Figure 3. We observe that: ❶ For ESM-1B, all extracted sparse neural networks incur significant performance degradation whenever the sparsity level is larger than 20%. Note that here we only sparsify the fully connected layers in multi-head self-attentions & feed-forward networks of each

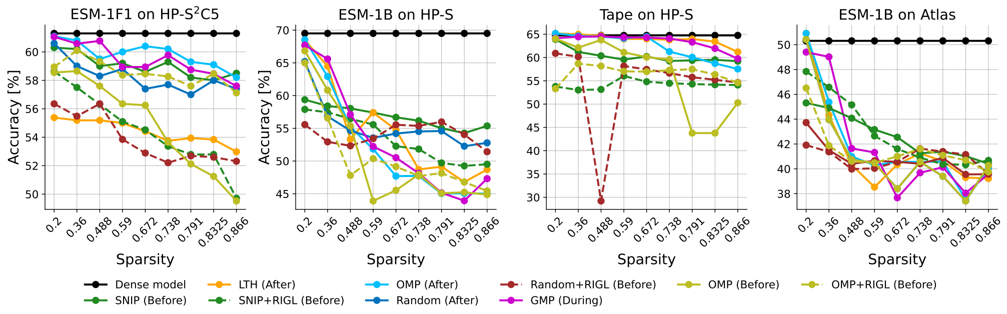

Figure 3: Protein prediction performance of various sparse models.

transformer layer, and leave all other modules unpruned. Even under this loose condition, ESM-1B still fails after pruning on both HP-S and Meltome Atlas, which indicates the low parameter redundancy in ESM-1B for modeling protein thermal stability. ❷ In general, static and dynamic pruning algorithms achieve similar performance with ESM-1B. SNIP (Before) and SNIP + RIGL (Before) deliver relatively better accuracies, especially for the high sparsity ($\geq 48.80\%$) subnetworks on HP-S. ❸ As for the worse backbone TAPE compared with ESM-1B, magnitude-based prunings like LTH (After), OMP (After), and GMP (During) show satisfactory results before 59% sparsity.

Furthermore, we conduct a more fine-grained pruning schedule to investigate the tolerance of ESM-1B against sparsification. In detail, we prune 5% weights with OMP (after) on different modules in ESM-1B and present the obtained accuracies in Table 1. Q, K, V, O, and FFN represent the fully connected layer in the query, key, value, & output of the self-attention module and feed-forward networks, respectively. The results show that whatever modules we select, 5% **sparsity** damages the ESM-1B performance of protein thermostability prediction on HP-S^{2}C2.

Table 1: OMP (after) pruning 5% weights of ESM-1B on different modules with HP-S^{2}C2.

Pruned Modules	Accuracy (↑)
None (Dense)	94.68
Q, K, V, O, FFN	92.19
Q, K, V	92.55
Q, K	93.62
Q, V	93.62
K, V	92.55

4.3.2 Structure-Based Models

Implementation Details. We further consider a representative structure-based algorithm for thermostability prediction, *i.e.*, ESM-IF1 (Hsu et al., 2022). Specifically, for ESM-IF1, we train the backbone and its linear head with learning rates of 1×10^{-4} and 2×10^{-2}, respectively. A batch size of 4 is used for both models on HotProtein (HP-S^{2}C5). Classification testing accuracy is reported to reflect the model performance.

Results and Analyses. In this section, we study structure-based models and their sparse variants on HotProtein (HP-S^{2}C5). ESM-IF1 (Hsu et al., 2022), a recent SOTA approach, is chosen for benchmarking. It takes the 3D structure of proteins as input and predicts its thermal stable temperature. As shown in Figure 3, ESM-IF1 produces inferior sparse neural networks with all pruning mechanisms of both static and dynamic, where OMP (after) and GMP (During) present relatively better accuracies.

4.4 Multilingual Translation

Implementation Details. We choose the official multilingual model mBART[1] (Liu et al., 2020), which was originally pre-trained on 25 languages using masked language modeling (MLM), following the fine-tuning setting of Tang et al. (2020). We first choose 10 languages from the language pools used for MLM pre-training; create three sub-groups containing 2, 5, 10 languages; and fine-tune mBART on each sub-group, referring to 2-to-2, 5-to-5, and 10-to-10 multilingual translation fine-tuning, respectively. During inference, we report the averaged BLEU (Tang et al., 2020; Liu et al., 2020) scores of bi-directional translation across 10 languages to measure the translation performance. Hence, the task difficulty monotonically decreases from 2-to-2 to 5-to-5, and to 10-to-10 fine-tuning as more languages are involved during training. The default fine-tuning configurations in Tang et al. (2020) are adopted for 40K iterations with an Adam optimizer and a learning rate of 1×10^{-6}.

Results and Analyses. Intuitively, fewer languages involved during fine-tuning leads to a more difficult translation for all languages. As demonstrated in Figure 4, several consistent conclusions can be drawn: ❶ Besides OMP (After) and LTH (After), all other produced sparse subnetworks perform

[1] https://github.com/facebookresearch/fairseq

worse than the dense baseline when the sparsity is larger than 20%. The BLEU scores of OMP (After) and LTH (After) also decline and fail to match at $\geq 20\%$, $\geq 48.8\%$, $\geq 59\%$ sparsity levels for 2-to-2, 5-to-5, and 10-to-10 fine-tuning, respectively. ❷ Magnitude-based sparsifications like OMP, LTH, and GMP are comparably robust across all three translation settings, while other pruning methods have negligible advantages compared to random pruning. ❸ While the overall tendency of SNNs is quite consistent across different tasks, the prunability of mBART increases as more languages are involved during fine-tuning. It seems that multilingual translation has already been a challenging task for pruning, and involving more languages in inference causes extra obstacles. This is why in the hardest scenario of fine-tuning on 2-to-2 and evaluating with 10 languages, all sparse subnetworks suffer from substantial performance degradation.

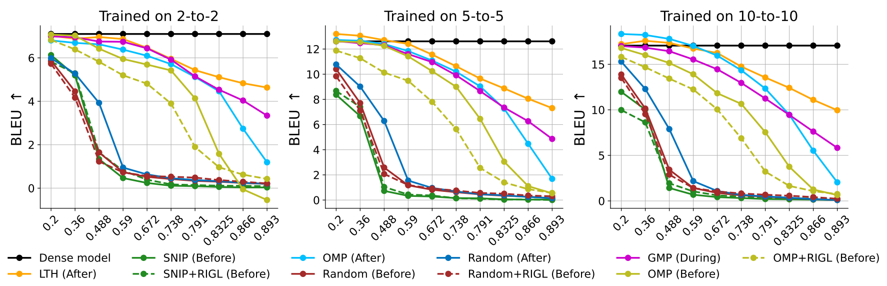

Figure 4: Multilingual performance of various sparse mBART. All models are tested on 10-to-10 multilingual translation and the averaged BLEU are reported.

4.5 Why SNNs Fail on SMC-Bench

We conduct a thorough investigation into the reasons why most SNNs struggle to perform on SMC-Bench. Our analysis identifies two possible causes for the failure: (1) the "lazy regime" in LLMs, and (2) the specific model components that are pruned. Based on these findings, we discover a set of stronger pruning recipes that alleviates (though still more or less suffering from) the failure on SMC-Bench, by breaking down the performance of the state-of-the-art BERT-pruning framework - oBERT (Kurtic et al., 2022) on SMC-Bench (note that most models evaluated in this paper are also BERT-based). Due to the limited space, we present our full investigation in Appendix C, and briefly present our sanity check of layer collapse below.

Does layer collapse occur unexpectedly on SMC-Bench? Layer collapse is the most common cause that blocks the information flow (signal propagation) of sparse neural networks, leading to a catastrophic performance drop (Tanaka et al., 2020). We plot the layerwise sparsity ratios of various sparse models in Appendix C.1. We do not observe severe layer collapse across methods except for SNIP which specifically removes nearly entire embedding layers. However, we do observe an unexpected phenomenon: *layerwise sparsities of different magnitude-based pruning approaches (i.e., IMP, OMP, and GMP) are extremely similar,* all overlapped on one line, despite the significant performance gap among them (up to 42.3%); small differences only start to appear until reaching very deep layers (e.g., classification heads) (see Appendix C.1 for more details). This abnormal behavior is highly correlated with the "lazy regime" (Neyshabur et al., 2020; Malladi et al., 2022) where the model stays in the same basin during fine-tuning with fairly small weight changes, and hence all magnitude-based pruning approaches, before, during, and after fine-tuning, tend to collapse to the same solution.

5 Conclusion

Given the enormous breakthroughs and the fruitful results that sparse neural networks have achieved in numerous fields, it is necessary to rethink the sufficiency of current evaluation protocols and introduce more difficult and diverse benchmarks to explore the limitation of sparse neural networks. In this paper, we assemble a large-scale, challenging, and more diverse benchmark, SMC-Bench. Through our thorough evaluation across various leading sparsifications, we confirm that SMC-Bench notably challenges the capability of most magnitude- or/and gradient-based sparse algorithms. We further dig deeper into the behavior of SNNs, and observe several surprising phenomena that are absent in the current evaluation. Our analysis points out that such failure is intimately related to the "lazy regime", which leads us to a suite of strong pruning recipes that alleviates (yet still more or less suffering from) the failure on SMC-Bench. Our subsequent effort will focus on exploring stronger sparse training algorithms that can scale and generalize on SMC-Bench, and meanwhile will consider the training costs of different sparse algorithms for a more holistic picture of their efficiency benefits.

Acknowledgement

We thank Dan Alistarh and Eldar Kurtic for the extremely helpful discussions about the implementation details of oBERT as well as our benchmark's claims; and Zhangheng Li for helping run extra experiments with oBERT. S. Liu and Z. Wang are in part supported by the NSF AI Institute for Foundations of Machine Learning (IFML). Part of this work used the Dutch national e-infrastructure with the support of the SURF Cooperative using grant no. NWO2021.060, EINF-2694 and EINF-2943/L1.

References

Open source parallel corpus of opus. https://opus.nlpl.eu/opus-100.php, 2020.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*, 2019.

Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. What is the state of neural network pruning? *Proceedings of machine learning and systems*, 2:129–146, 2020.

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. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Zhangyang Wang, and Michael Carbin. The lottery ticket hypothesis for pre-trained bert networks. *Advances in neural information processing systems*, 33:15834–15846, 2020.

Tianlong Chen, Zhenyu Zhang, Santosh Balachandra, Haoyu Ma, Zehao Wang, Zhangyang Wang, et al. Sparsity winning twice: Better robust generalization from more efficient training. In *International Conference on Learning Representations*, 2022.

Tianlong Chen, Chengyue Gong, Daniel Jesus Diaz, Xuxi Chen, Jordan Tyler Wells, qiang liu, Zhangyang Wang, Andrew Ellington, Alex Dimakis, and Adam Klivans. Hotprotein: A novel framework for protein thermostability prediction and editing. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=YDJRFWBMNby.

Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *Advances in neural information processing systems*, 32, 2019.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

Pau de Jorge, Amartya Sanyal, Harkirat Behl, Philip Torr, Grégory Rogez, and Puneet K. Dokania. Progressive skeletonization: Trimming more fat from a network at initialization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=9GsFOUyUPi.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Tim Dettmers and Luke Zettlemoyer. Sparse networks from scratch: Faster training without losing performance. *arXiv preprint arXiv:1907.04840*, 2019.

James Diffenderfer, Brian Bartoldson, Shreya Chaganti, Jize Zhang, and Bhavya Kailkhura. A winning hand: Compressing deep networks can improve out-of-distribution robustness. *Advances in Neural Information Processing Systems*, 34:664–676, 2021.

Xin Dong, Shangyu Chen, and Sinno Pan. Learning to prune deep neural networks via layer-wise optimal brain surgeon. *Advances in Neural Information Processing Systems*, 30, 2017.

Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. Rigging the lottery: Making all tickets winners. In *International Conference on Machine Learning*, pp. 2943–2952. PMLR, 2020.

William Finnoff, Ferdinand Hergert, and Hans Georg Zimmermann. Improving model selection by nonconvergent methods. *Neural Networks*, 6(6):771–783, 1993.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJl-b3RcF7.

Jonathan Frankle, David J. Schwab, and Ari S. Morcos. The early phase of neural network training. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Hkl1iRNFwS.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Pruning neural networks at initialization: Why are we missing the mark? In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Ig-VyQc-MLK.

Elias Frantar, Eldar Kurtic, and Dan Alistarh. M-fac: Efficient matrix-free approximations of second-order information. *Advances in Neural Information Processing Systems*, 34:14873–14886, 2021.

Trevor Gale, Erich Elsen, and Sara Hooker. The state of sparsity in deep neural networks. *arXiv preprint arXiv:1902.09574*, 2019.

Yiwen Guo, Chao Zhang, Changshui Zhang, and Yurong Chen. Sparse dnns with improved adversarial robustness. *Advances in neural information processing systems*, 31, 2018.

Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *International Conference on Learning Representations*, 2015.

Babak Hassibi and David Stork. Second order derivatives for network pruning: Optimal brain surgeon. *Advances in neural information processing systems*, 5, 1992.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Chloe Hsu, Robert Verkuil, Jason Liu, Zeming Lin, Brian Hie, Tom Sercu, Adam Lerer, and Alexander Rives. Learning inverse folding from millions of predicted structures. *bioRxiv*, 2022. doi: 10.1101/2022.04.10.487779. URL https://www.biorxiv.org/content/early/2022/04/10/2022.04.10.487779.

Anna Jarzab, Nils Kurzawa, Thomas Hopf, Matthias Moerch, Jana Zecha, Niels Leijten, Yangyang Bian, Eva Musiol, Melanie Maschberger, Gabriele Stoehr, et al. Meltome atlas—thermal proteome stability across the tree of life. *Nature methods*, 17(5):495–503, 2020.

Hao Jiang, Ke Zhan, Jianwei Qu, Yongkang Wu, Zhaoye Fei, Xinyu Zhang, Lei Chen, Zhicheng Dou, Xipeng Qiu, Zikai Guo, et al. Towards more effective and economic sparsely-activated model. *arXiv preprint arXiv:2110.07431*, 2021.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. Mawps: A math word problem repository. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1152–1157, 2016.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Eldar Kurtic, Daniel Campos, Tuan Nguyen, Elias Frantar, Mark Kurtz, Benjamin Fineran, Michael Goin, and Dan Alistarh. The optimal bert surgeon: Scalable and accurate second-order pruning for large language models. *arXiv preprint arXiv:2203.07259*, 2022.

Aditya Kusupati, Vivek Ramanujan, Raghav Somani, Mitchell Wortsman, Prateek Jain, Sham Kakade, and Ali Farhadi. Soft threshold weight reparameterization for learnable sparsity. In *International Conference on Machine Learning*, pp. 5544–5555. PMLR, 2020.

François Lagunas, Ella Charlaix, Victor Sanh, and Alexander M Rush. Block pruning for faster transformers. *arXiv preprint arXiv:2109.04838*, 2021.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*, 2017.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Yann LeCun, John S Denker, and Sara A Solla. Optimal brain damage. In *Advances in neural information processing systems*, pp. 598–605, 1990.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip Torr. SNIP: SINGLE-SHOT NETWORK PRUNING BASED ON CONNECTION SENSITIVITY. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=B1VZqjAcYX.

Hector Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*, 2012.

Tao Lin, Sebastian U. Stich, Luis Barba, Daniil Dmitriev, and Martin Jaggi. Dynamic model pruning with feedback. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJem8lSFwB.

Shiwei Liu and Zhangyang Wang. Ten lessons we have learned in the new" sparseland": A short handbook for sparse neural network researchers. *arXiv preprint arXiv:2302.02596*, 2023.

Shiwei Liu, Tianlong Chen, Zahra Atashgahi, Xiaohan Chen, Ghada Sokar, Elena Mocanu, Mykola Pechenizkiy, Zhangyang Wang, and Decebal Constantin Mocanu. Deep ensembling with no overhead for either training or testing: The all-round blessings of dynamic sparsity. *arXiv preprint arXiv:2106.14568*, 2021a.

Shiwei Liu, Tianlong Chen, Xiaohan Chen, Zahra Atashgahi, Lu Yin, Huanyu Kou, Li Shen, Mykola Pechenizkiy, Zhangyang Wang, and Decebal Constantin Mocanu. Sparse training via boosting pruning plasticity with neuroregeneration. *Advances in Neural Information Processing Systems (NeurIPs).*, 2021b.

Shiwei Liu, Lu Yin, Decebal Constantin Mocanu, and Mykola Pechenizkiy. Do we actually need dense over-parameterization? in-time over-parameterization in sparse training. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 6989–7000. PMLR, 2021c.

Shiwei Liu, Tianlong Chen, Xiaohan Chen, Li Shen, Decebal Constantin Mocanu, Zhangyang Wang, and Mykola Pechenizkiy. The unreasonable effectiveness of random pruning: Return of the most naive baseline for sparse training. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=VBZJ_3tz-t.

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

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742, 2020.

Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. Rethinking the value of network pruning. *arXiv preprint arXiv:1810.05270*, 2018.

Christos Louizos, Max Welling, and Diederik P Kingma. Learning sparse neural networks through l_0 regularization. *International Conference on Learning Representations*, 2018.

Jian-Hao Luo and Jianxin Wu. Autopruner: An end-to-end trainable filter pruning method for efficient deep model inference. *Pattern Recognition*, 107:107461, 2020.

Sadhika Malladi, Alexander Wettig, Dingli Yu, Danqi Chen, and Sanjeev Arora. A kernel-based view of language model fine-tuning. *arXiv preprint arXiv:2210.05643*, 2022.

Shen-Yun Miao, Chao-Chun Liang, and Keh-Yih Su. A diverse corpus for evaluating and developing english math word problem solvers. *arXiv preprint arXiv:2106.15772*, 2021.

Deepak Mittal, Shweta Bhardwaj, Mitesh M Khapra, and Balaraman Ravindran. Studying the plasticity in deep convolutional neural networks using random pruning. *Machine Vision and Applications*, 30(2):203–216, 2019.

Decebal Constantin Mocanu, Elena Mocanu, Peter Stone, Phuong H Nguyen, Madeleine Gibescu, and Antonio Liotta. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature communications*, 9(1):2383, 2018.

Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Variational dropout sparsifies deep neural networks. In *International Conference on Machine Learning*, pp. 2498–2507. PMLR, 2017.

Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. *International Conference on Learning Representations*, 2016.

Hesham Mostafa and Xin Wang. Parameter efficient training of deep convolutional neural networks by dynamic sparse reparameterization. In *International Conference on Machine Learning*, pp. 4646–4655. PMLR, 2019.

Michael C Mozer and Paul Smolensky. Using relevance to reduce network size automatically. *Connection Science*, 1(1):3–16, 1989.

Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning? *Advances in neural information processing systems*, 33:512–523, 2020.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

Ozan Özdenizci and Robert Legenstein. Training adversarially robust sparse networks via bayesian connectivity sampling. In *International Conference on Machine Learning*, pp. 8314–8324. PMLR, 2021.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are nlp models really able to solve simple math word problems? *arXiv preprint arXiv:2103.07191*, 2021.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Peter Chen, John Canny, Pieter Abbeel, and Yun Song. Evaluating protein transfer learning with tape. *Advances in neural information processing systems*, 32, 2019.

Alex Renda, Jonathan Frankle, and Michael Carbin. Comparing rewinding and fine-tuning in neural network pruning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1gSj0NKvB.

Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15), 2021.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Victor Sanh, Thomas Wolf, and Alexander Rush. Movement pruning: Adaptive sparsity by fine-tuning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 20378–20389. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/eae15aabaa768ae4a5993a8a4f4fa6e4-Paper.pdf.

Pedro Savarese, Hugo Silva, and Michael Maire. Winning the lottery with continuous sparsification. *Advances in Neural Information Processing Systems*, 33:11380–11390, 2020.

Jonathan Schwarz, Siddhant Jayakumar, Razvan Pascanu, Peter E Latham, and Yee Teh. Powerpropagation: A sparsity inducing weight reparameterisation. *Advances in neural information processing systems*, 34:28889–28903, 2021.

Sidak Pal Singh and Dan Alistarh. Woodfisher: Efficient second-order approximation for neural network compression. *Advances in Neural Information Processing Systems*, 33:18098–18109, 2020.

Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pp. 1100612. International Society for Optics and Photonics, 2019.

Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*, 2018.

Hidenori Tanaka, Daniel Kunin, Daniel LK Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. *Advances in Neural Information Processing Systems. arXiv:2006.05467*, 2020.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. Multilingual translation with extensible multilingual pretraining and finetuning. *arXiv preprint arXiv:2008.00401*, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

Chaoqi Wang, Roger Grosse, Sanja Fidler, and Guodong Zhang. Eigendamage: Structured pruning in the kronecker-factored eigenbasis. In *International Conference on Machine Learning*, pp. 6566–6575. PMLR, 2019.

Chaoqi Wang, Guodong Zhang, and Roger Grosse. Picking winning tickets before training by preserving gradient flow. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkgsACVKPH.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, pp. 23965–23998. PMLR, 2022.

Zhipeng Xie and Shichao Sun. A goal-driven tree-structured neural model for math word problems. In *IJCAI*, pp. 5299–5305, 2019.

Dongkuan Xu, Ian EH Yen, Jinxi Zhao, and Zhibin Xiao. Rethinking network pruning–under the pre-train and fine-tune paradigm. *arXiv preprint arXiv:2104.08682*, 2021.

Lu Yin, Shiwei Liu, Fang Meng, Tianjin Huang, Vlado Menkovski, and Mykola Pechenizkiy. Lottery pools: Winning more by interpolating tickets without increasing training or inference cost. *arXiv preprint arXiv:2208.10842*, 2022.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022.

Ofir Zafrir, Ariel Larey, Guy Boudoukh, Haihao Shen, and Moshe Wasserblat. Prune once for all: Sparse pre-trained language models. *arXiv preprint arXiv:2111.05754*, 2021.

Wenyuan Zeng and Raquel Urtasun. Mlprune: Multi-layer pruning for automated neural network compression. 2018.

Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12104–12113, 2022.

Dinghuai Zhang, Kartik Ahuja, Yilun Xu, Yisen Wang, and Aaron Courville. Can subnetwork structure be the key to out-of-distribution generalization? In *International Conference on Machine Learning*, pp. 12356–12367. PMLR, 2021.

Jipeng Zhang, Lei Wang, Roy Ka-Wei Lee, Yi Bin, Yan Wang, Jie Shao, and Ee-Peng Lim. Graph-to-tree learning for solving math word problems. Association for Computational Linguistics, 2020a.

Qingru Zhang, Simiao Zuo, Chen Liang, Alexander Bukharin, Pengcheng He, Weizhu Chen, and Tuo Zhao. Platon: Pruning large transformer models with upper confidence bound of weight importance. In *International Conference on Machine Learning*, pp. 26809–26823. PMLR, 2022.

Yian Zhang, Alex Warstadt, Haau-Sing Li, and Samuel R Bowman. When do you need billions of words of pretraining data? *arXiv preprint arXiv:2011.04946*, 2020b.

Michael Zhu and Suyog Gupta. To prune, or not to prune: exploring the efficacy of pruning for model compression. *arXiv preprint arXiv:1710.01878*, 2017.

A Summary of Tasks, Models, Datasets, and Training

We summarize the combinations of models and configurations that we used to evaluate SNNs on SMC-Bench.

Table 2: Summary of models and datasets that we used to evaluate on SMC-Bench.

Task	Datasets	Models	Source
Commonsense Reasoning	CSQA	RoBERTa Large	Facebook AI Research Sequence-to-Sequence Toolkit (Ott et al., 2019)
	WinoGrande	RoBERTa Large	Facebook AI Research Sequence-to-Sequence Toolkit (Ott et al., 2019)
	RACE	RoBERTa Large	Facebook AI Research Sequence-to-Sequence Toolkit (Ott et al., 2019)
Arithmetic Reasoning	MAVPS	GTS	GitHub Repository (Patel et al., 2021)
		Graph2Tree	GitHub Repository (Patel et al., 2021)
	ASDiv-A	GTS	GitHub Repository (Patel et al., 2021)
		Graph2Tree	GitHub Repository (Patel et al., 2021)
	SVAMP	GTS	GitHub Repository (Patel et al., 2021)
		Graph2Tree	GitHub Repository (Patel et al., 2021)
Protein Thermostability Prediction	HotProtein (HP-S)	TAPE	GitHub Repository (Rao et al., 2019)
		ESM-1B	GitHub Repository (Rives et al., 2021)
	HotProtein (HP-S^{2}C5)	ESM-IF1	GitHub Repository (Rives et al., 2021)
	HotProtein (HP-S^{2}C2)	ESM-1B	GitHub Repository (Rives et al., 2021)
	Meltome Atlas	ESM-1B	GitHub Repository (Rives et al., 2021)
Multilingual Translation	2-to-2	mBART	Facebook AI Research Sequence-to-Sequence Toolkit (Ott et al., 2019)
	5-to-5	mBART	Facebook AI Research Sequence-to-Sequence Toolkit (Ott et al., 2019)
	10-to-10	mBART	Facebook AI Research Sequence-to-Sequence Toolkit (Ott et al., 2019)

We strictly follow the training configurations reported in the original source to replicate the results of each task. The hyperparameters and configurations used for each model in this paper are shared below.

A.1 Commonsense Reasoning

Table 3: Hyperparameters and training configurations used for models on Commonsense Reasoning.

Models	**RoBERTa**	**RoBERTa**	**RoBERTa**
Dataset	CSQA	WinoGrande	RACE
Pre-trained Models	RoBERTa	RoBERTa	RoBERTa
Hidden Size	[1024]	[1024]	[1024]
FFN Inner Hidden Size	[4096]	[4096]	[4096]
Number of Layers	[24]	[24]	[24]
Learning Rate	[1e-5]	[1e-5]	[1e-5]
Weight Decay	[0.01]	[0.01]	[0.01]
Batch Size	[16]	[32]	[16]
Dropout	[0.1]	[0.1]	[0.1]
Attention Dropout	[0.1]	[0.1]	[0.1]
Clip Norm	[0.0]	[0.0]	[0.0]
Adam ϵ	[1e-06]	[1e-06]	[1e-06]
Adam β_1	[0.9]	[0.9]	[0.9]
Adam β_1	[0.98]	[0.98]	[0.98]
# Parameters	355M	355M	355M
Training Time	3000 steps	23750 steps	3 epochs
Wramup Time	150 steps	2375 steps	500 steps

A.2 Arithmetic Reasoning

Table 4: Hyperparameters and training configurations used for models on Arithmetic Reasoning.

Models	**GTS**	**Graph2Tree**
Dataset	MAVPS, ASDiv-A, SVAMP	MAVPS, ASDiv-A, SVAMP
Pre-trained Embedding	RoBERTa	RoBERTa
Embedding Size	[768]	[768]
Hidden Size	[512]	[384]
Number of Layers	[2]	[2]
Learning Rate	[1e-3]	[8e-4]
Weight Decay	[1e-5]	[1e-5]
Embedding LR	[8e-6]	[1e-5]
Batch Size	[4 (MAVPS, ASDiv-A), 8 (SVAMP)]	[4 (MAVPS, ASDiv-A), 8 (SVAMP)]
Dropout	[0.5]	[0.5]
Adam ϵ	[1e-08]	[1e-08]
Adam β_1	[0.9]	[0.9]
Adam β_1	[0.999]	[0.999]
# Parameters	140M	143M
Training Time	50 epochs	50 epochs

A.3 Protein Thermostability Prediction

Table 5: Hyperparameters and training configurations used for models on Protein Thermostability Prediction.

Models	**TAPE**	**ESM-1B**	**ESM-IF1**
Dataset	HP-S	HP-S^{2}C2, Meltome Atlas, HP-S	HP-S^{2}C5
Hidden Size	[768]	[1280]	[512]
Number of Layers	[12]	[33]	[20]
Learning Rate	[1e-4]	[2e-2 (head), 1e-6 (backbone)]	[2e-2 (head), 1e-4 (backbone)]
Weight Decay	[1e-2]	[1e-2]	[5e-2]
Batch Size	[16]	[3,2,3]	[4]
Attention Dropout	[0.1]	[0.0]	[0.1]
Dropout	[0.1]	[0.0]	[0.1]
Adam ϵ	[1e-08]	[1e-08]	[1e-08]
Adam β_1	[0.9]	[0.9]	[0.9]
Adam β_1	[0.999]	[0.999]	[0.999]
# Parameters	92M	650M	124M
Training Time	4 epochs	4 epochs	8 epochs

A.4 Multilingual Translation

Table 6: Hyperparameters and training configurations used for models on Multilingual Translation.

Models	**mBART**	**mBART**	**mBART**
Dataset	2-to-2	5-to-5	10-to-10
Pre-trained Models	mBART	mBART	mBART
Hidden Size	[1024]	[1024]	[1024]
Number of Layers	[24]	[24]	[24]
Learning Rate	[3e-5]	[3e-5]	[3e-5]
Weight Decay	[0.0]	[0.0]	[0.0]
Batch Size	[16]	[32]	[16]
Dropout	[0.3]	[0.3]	[0.3]
Attention Dropout	[0.1]	[0.1]	[0.1]
Clip Norm	[0.0]	[0.0]	[0.0]
Adam ϵ	[1e-06]	[1e-06]	[1e-06]
Adam β_1	[0.9]	[0.9]	[0.9]
Adam β_1	[0.98]	[0.98]	[0.98]
# Parameters	680M	680M	680M
Training Time	40,000 steps	40,000 steps	40,000 steps
Wramup Time	2,500 steps	2,500 steps	2,500 steps

B Results of Arithmetic Reasoning with Magnitude before Training

In this appendix, we report the performance of SNNs on arithmetic reasoning including Magnitude before Training (OMP (Before)). We can clearly observe that the accuracy of magnitude pruning before training dramatically falls from 80% to nearly 0% when sparsity is larger than 36%. After checking the corresponding layerwise sparsity, we find that OMP (Before) completely removes all the weights from non-embedding and non-encoder layers, leading to severe layer collapse.

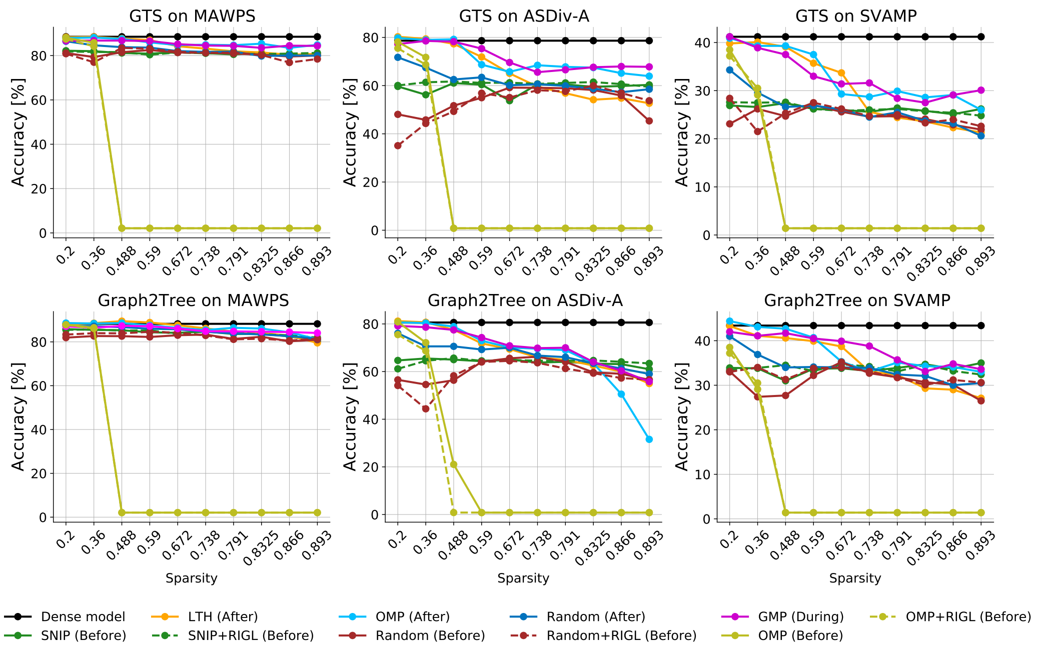

Figure 5: Arithmetic reasoning performance of various sparse GTS and Graph2Tree on MAWPS, ASDiv-A, and SVAMP.

C An Investigation of Why SNNs Fail on SMC-Bench

In this section, we conduct a full investigation, attempting to open the box for the potential causes of SNN failures on SMC-Bench. Our analysis reveals two possible causes: (1) the "lazy regime" in fine-tuning LLMs, and (2) the model components to prune. Due to the "lazy regime" phenomenon (Chizat et al., 2019; Malladi et al., 2022), commonly used pruning techniques that rely on magnitude and gradient can be very uninformative. Therefore, we turn to the latest strong second-order pruning framework - oBERT (Kurtic et al., 2022), which utilizes inverse-Hessian approximations to guide pruning decisions. We choose RoBERTa.large on CSQA to conduct this investigation. The roadmap of our full investigation is presented below.

- **Layerwise sparsity ratios** of various magnitude-based pruning methods are strikingly similar, suggesting that "lazy regime" may occur during fine-tuning. In this regime, the most common pruning criteria such as magnitude and gradient can be rather unreliable.
- **Second-order pruning** approaches like oBERT provide more faithful signals than magnitudes and gradients for LLM pruning, and hence achieve significantly higher accuracy at high sparsities.

C.1 Layerwise Sparsity Ratios on SMC-Bench

To check if severe layer collapse occurs on SMC-Bench, we plot the per-layer sparsity ratios discovered by various sparsification approaches at three sparsity levels: 36%, 64%, and 83%. Layers are ordered from input to output on the X-axis. We respectively report the layerwise sparsity of commonsense reasoning with RoBERTa on CSQA and RACE in Figure 6 and 7, and arithmetic reasoning with GTS on SVAMP in Figure 8. We summarize our main findings here.

❶ **Layerwise sparsities of magnitude-based pruning approaches are extremely similar.** IMP, OMP, and GMP that rely on weight magnitude for pruning share an extremely similar set of layerwise sparsities. Especially, sparsity values of magnitude pruning on commonsense reasoning are completely identical, all overlapped on blue lines, except for the tiny difference in classification heads. This phenomenon indicates that weights of RoBERTa excluding classifiers remain rather stable during commonsense reasoning fine-tuning so that all the magnitude pruning variants (both before and after) discover the same sparsity pattern. The sparsity difference of arithmetic reasoning is more distinguishable than commonsense reasoning. Still, sparsities in the encoder (pre-trained RoBERTa) of IMP, GMP, and OMP (After) largely overlap. Until reaching the later layers, the sparsity ratios of different approaches start to be distinct. ❷ **SNIP tends to prune all the weights in embedding layers aggressively.** Even at the mild 36% sparsity, SNIP prunes weights of embedding layers to 99.7% sparsity, which may explain why SNIP struggles on SMC-Bench. ❸ **OMP (Before) suffers from layer collapse on arithmetic reasoning.** We empirically find that OMP (Before) leads to completely empty deep layers when sparsity is larger than 36%, indicating the limitation of only considering pruning with the magnitude before fine-tuning or re-training.

The near-identical layerwise sparsity ratios across various magnitude-based methods remind us of the "lazy training" regime (Chizat et al., 2019; Malladi et al., 2022) which was revealed to occur during the fine-tuning of LLMs. Under this regime, weight changes during fine-tuning are negligible, hence non-informative and more "noisy". Consequently, various magnitude-based pruning approaches, regardless of their timing, all tend to converge to the same solution.

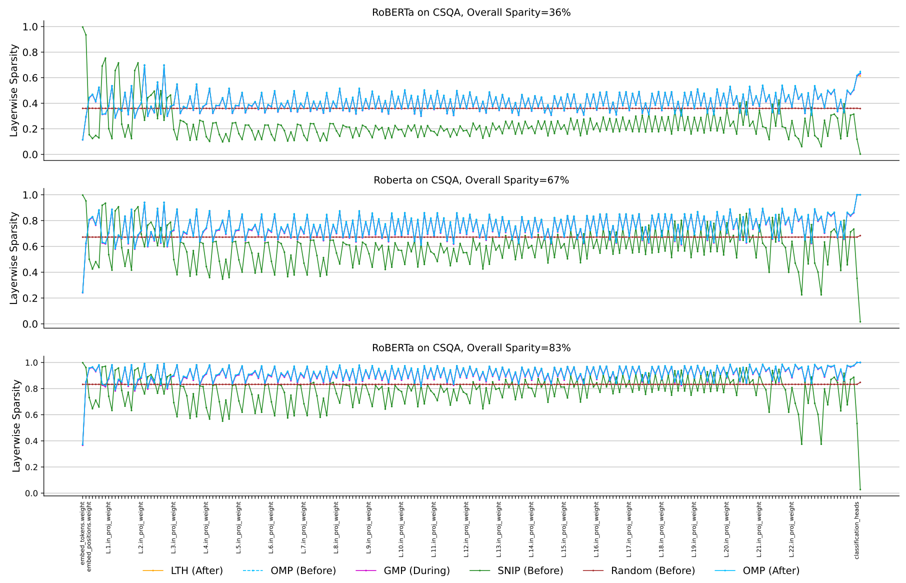

Figure 6: Layerwise sparsity of RoBERTa on CSQA at sparsity levels ∈ [36%, 67%, 83%].

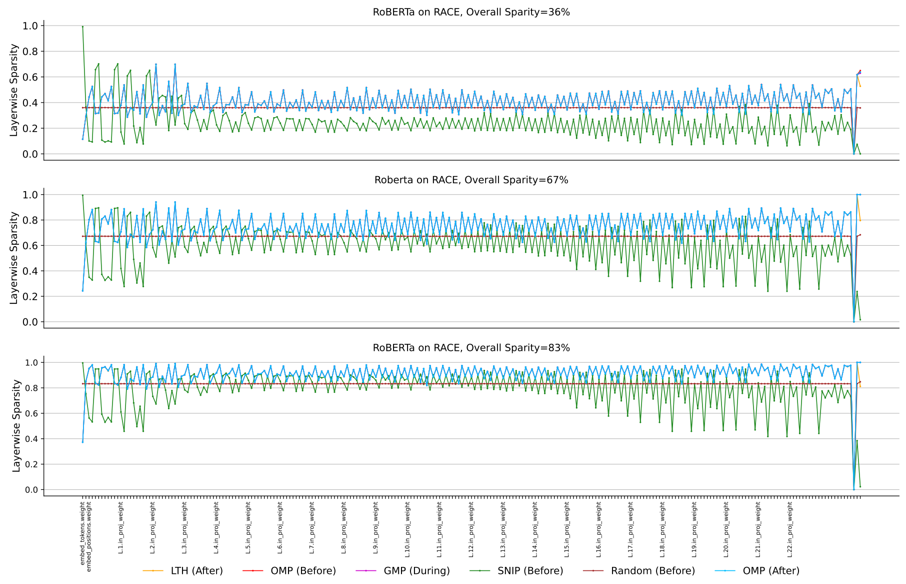

Figure 7: Layerwise sparsity of RoBERTa on RACE at sparsity levels ∈ [36%, 67%, 83%].

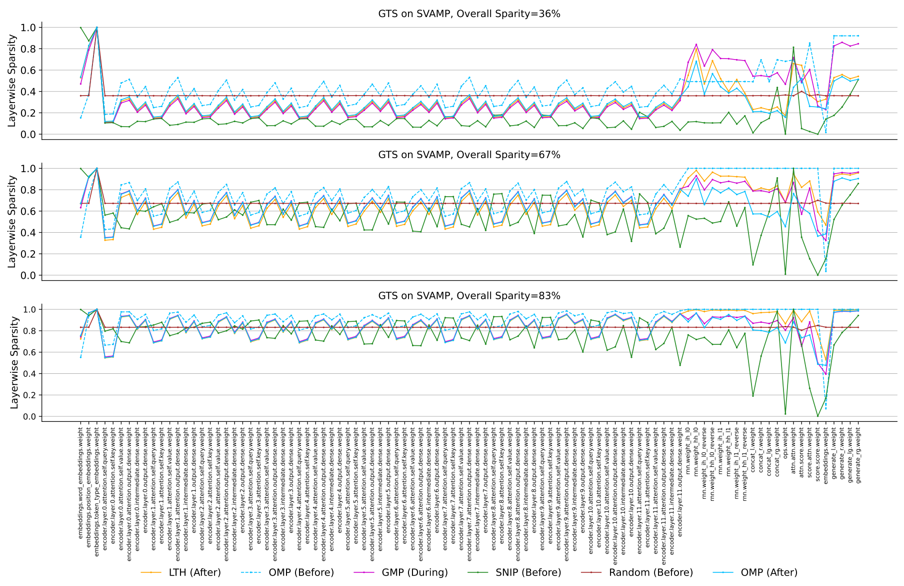

Figure 8: Layerwise sparsity of GTS on SVAMP at sparsity levels ∈ [36%, 67%, 83%].

C.2 Evaluation of oBERT on SMC-Bench

Based on our conjecture that magtitudes/gradients become unreliable for pruning LLMs, we hypothesize that the second-order pruning approaches with approximated Hessian matrix would be more accurate options. To verify our conjecture, we turn to the latest stronger second-order pruning framework, oBERT (Kurtic et al., 2022). Specifically, we follow Kurtic et al. (2022) and replace the magnitude pruning criterion of the LTH framework with the second-order oBERT criterion; adopt Learning Rate Rewinding (LRR) (Renda et al., 2020) and Knowledge Distillation (KD) (Hinton et al., 2015) during each pruning iteration; and keep the embeddings and classification heads dense.

Figure 9 support our hypothesis and demonstrates that oBERT notably outperforms the zero-order and first-order sparsification approaches, and substantially improves the accuracy to a competitive level. More importantly, oBERT produces a completely different layerwise sparsity pattern from magnitude-based pruning approaches, which is consistent with the patterns that are commonly observed in sparse computer vision models: deeper layers tend to have higher sparsities than lower layers (Evci et al., 2020; Kusupati et al., 2020; Tanaka et al., 2020; Liu et al., 2021b).

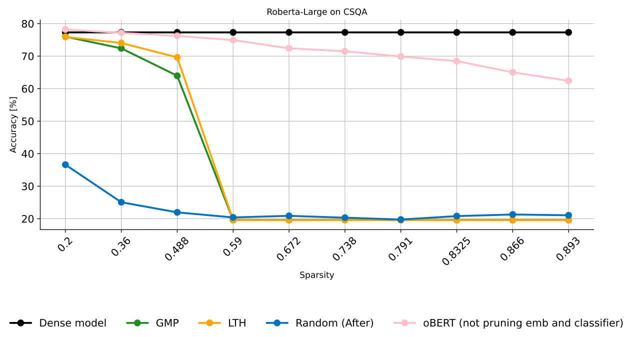

Figure 9: A roadmap of accuracy recovering via a suite of stronger pruning recipes with RoBERTa-Large on CSQA.

So far, our investigation has discovered that the pruning recipe used in (Kurtic et al., 2022) can remarkably improve the LLM pruning performance on SMC-Bench. Nevertheless, such a well-tuned pruning recipe is both time- and resource-intensive (9× more fine-tuning time, besides Hessian matrix

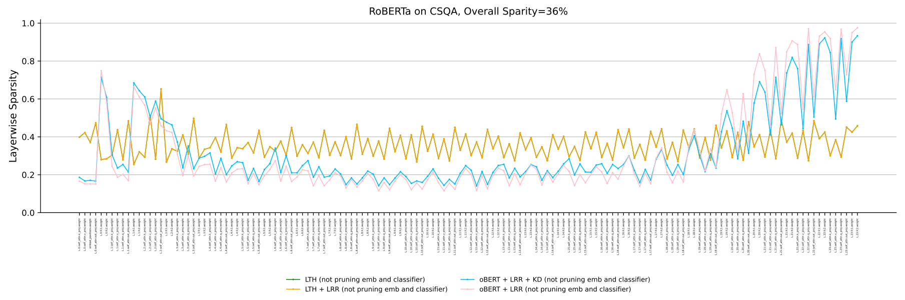

Figure 10: Layerwise sparsity comparison among LTH, LRR, oBERT, and KD with RoBERTa-Large on CSQA.

approximation); and even so, the strongest SNNs still fall short of their dense counterpart by around 10% accuracy at sparsities between $60\% - 80\%$, in contrast to "normal" SNNs that easily match their dense models on CIFAR, ImageNet, or GLUE. Therefore, the main claim of our paper still holds, that is, SMC-Bench indeed provides a new benchmark that is way more challenging for SOTA sparse algorithms, than existing testbeds.

D Summary of Evaluation Tasks and Datasets in 100 Papers

Table 7: Summary of Evaluation Tasks and Datasets Used in 100 Recent SNN Papers.

Task	Total #Paper	Datasets	#Paper
Image Classification	82	ImageNet	62
		CIFAR-10	59
		CIFAR-100	37
		MNIST	26
		Fashion MNIST	10
		SVHN	4
		Birds-200	1
		Flowers-102	1
		EMNIST	1
NLP Task	16	GLUE	9
		SQUAD	4
		WikiText-103	3
		WMT	5
		IMDB	1
		AAN	1
		LO	1
		OpenWeb text	1
		One Billion Word Benchmark	1
Face Recognition	3	LFW	3
		Youtube Faces	2
		CASIA-WebFace	1
Object Detection	3	COCO dataset	2
		PASCAL-VOL-2007	1
Speech Recognition	2	Google-12	1
		TIMIT	1
High-Resolution Reconstruction	2	Set5	2
		Set14	2
		B100	2
		Urban100	2
		Manga109	2
Image Generation	2	CIFAR-10	2
		ImageNet	1
		STL-10	1
Human Activity Recognition	1	HAR-2	1
Microarray Classification	1	Leukemia	1
		CLL-SUB-111	1
		SMK-CAN-18	1
		GLI-85	1
Hand Gesture Reconstruction	1	nvGesture	1
Regression Task	1	NYU Depth	1
3D Object Part Segmentation	1	ShapeNet	1
RL task	1	CartPole	1
		Acrobot	1
		MountainCar	1
		Atari suite	1
Vedio Deblurring	1	DVD	1
		GOPRO	1
		Real Blurry Videos	1
Vocabulary Speech Recognition	1	VS	1
		SWB	1