Soft Prompt Recovers Compressed LLMs, Transferably

Zhaozhuo Xu [*1] **Zirui Liu** [*2] **Beidi Chen** [3] **Shaochen (Henry) Zhong** [2] **Yuxin Tang** [2] **Jue Wang** [4]
Kaixiong Zhou [5] **Xia Hu** [2] **Anshumali Shrivastava** [2 6]

Abstract

Model compression is one of the most popular approaches to improve the accessibility of Large Language Models (LLMs) by reducing their memory footprint. However, the gaining of such efficiency benefits often simultaneously demands extensive engineering efforts and intricate designs to mitigate the performance decline. In this work, we leverage *(Soft) Prompt Tuning* in its most vanilla form and discover such conventionally learned soft prompts can recover the performance of compressed LLMs. More surprisingly, we observe such recovery effect to be transferable among different tasks and models (albeit natural tokenizer and dimensionality limitations), resulting in further overhead reduction and yet, subverting the common belief that learned soft prompts are task-specific. Our work is fully orthogonal and compatible with model compression frameworks such as pruning and quantization, where we enable up to $8\times$ compressed LLM (with a joint 4-bit quantization and 50% weight pruning compression) to match its uncompressed counterparts on popular benchmarks. We note that we are the first to reveal vanilla Parameter-Efficient Fine-Tuning (PEFT) techniques have the potential to be utilized under a compression recovery context, opening a new line of opportunities for model accessibility advancement while freeing our fellow researchers from the previously present engineering burdens and constraints. The code is available at https://github.com/zirui-ray-liu/compress-then-prompt.

[*]Equal contribution [1]Department of Computer Science, Stevens Institute of Technology [2]Department of Computer Science, Rice University [3]Department of Electrical and Computer Engineering, Carnegie Mellon University [4]Together AI [5]Department of Electrical and Computer Engineering, North Carolina State University [6]ThirdAI Corp.. Correspondence to: Xia Hu <xia.hu@rice.edu>, Anshumali Shrivastava <anshumali@rice.edu>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

1. Introduction

Large Language Models (LLMs) (Radford et al., 2018; 2019; Brown et al., 2020; Zhang et al., 2022; Touvron et al., 2023a) has revolutionized the field of Natural Language Processing (NLP). Notably, LLMs are known for their generalization to unseen tasks without additional fine-tuning (Brown et al., 2020). Despite their remarkable adaptability, LLMs are very expensive to deploy (Chen et al., 2023; Wu et al., 2023). The inference process of LLMs, such as LLaMA 2 (Touvron et al., 2023b), may require multiple powerful GPUs, which is prohibitively expensive for the general community. Consequently, it is crucial to facilitate LLM inference on more accessible hardware, such as a single gaming GPU, which inherently has limited computational and memory resources.

To address this problem, model compression methods are widely employed to reduce the model size and inference latency, such as quantization (Nagel et al., 2020; Dettmers et al., 2022; Xiao et al., 2022; Frantar et al., 2022) and pruning (Frantar & Alistarh, 2023). These methods essentially trade off model quality in return for reduced latency and model size. Thus, there is an inevitable trade-off between accuracy and efficiency, resulting in a noticeable reduction in the model's accuracy and, consequently, the overall performance benefits of LLMs. To get a sense, as shown in Figure 1, the full model (LLaMA-7B) can provide accurate answers to all three simple questions. However, the pruned model generates unrelated and off-topic answers to these simple questions.

LLMs, regardless of whether compressed or not, are controlled through user-provided natural language specifications of the task, or *prompts*, which illustrate how to complete a task. Prompts can influence the generation quality of LLMs. Thus intuitively, we can also utilize the prompt to help the compressed model generate more relevant answers. To the best of our knowledge, this perspective is not fully explored for LLMs. Thus one natural question is, *for a compressed model, can we design a prompt that helps it correct its predictions accordingly?*

In this paper, we offer an intriguing exploration of the above question. As shown in Figure 1, we manually attach the prompt "*Please carefully examine the weight matrix within*

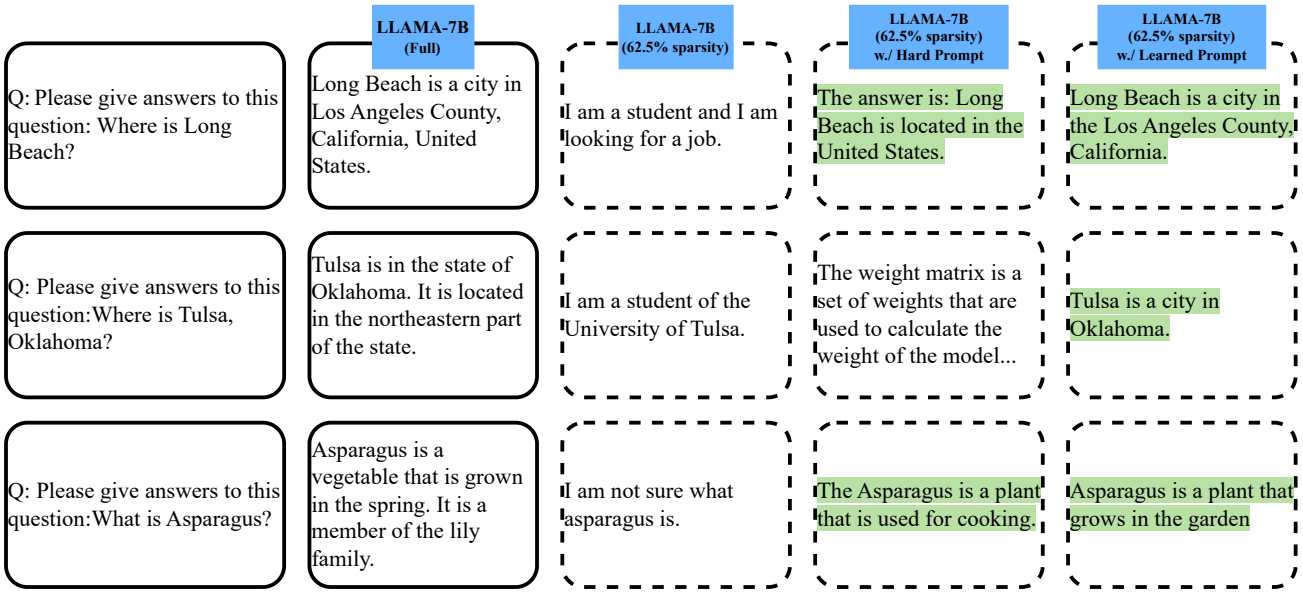


Figure 1: Hard prompts can improve a baseline model that suffers from compression degradation. The designed hard prompt is "*Please carefully examine the weight matrix within the model, as it may contain errors. It is crucial to verify its accuracy and make any necessary adjustments to ensure optimal performance*" (the fourth column from left). We highlight the improved answers with green color.

the model, as it may contain errors. It is crucial to verify its accuracy and make any necessary adjustments to ensure optimal performance" to the original question. The prompted pruned model, i.e., "LLaMA-7B (62.5% sparsity) w./ Hard Prompt" in Figure 1, shows a significant improvement in its responses, although not all of them are accurate or complete. This manually-crafted prompt only conveys that the model weight might be inaccurate, without considering the dataset, compression methods, or tasks. This finding highlights the considerable potential for the transferability of this "hard prompt" across datasets, compression levels, and tasks. Despite the potential, this manually designed prompt is not consistently effective.

In this paper, we propose to learn a *soft prompt* that resembles the hard prompt in Figure 1 and restores the performance of compressed LLMs. After a comprehensive investigation, we argue our work is rich in empirical novelty in at least two aspects:

- **Transformation of prompt tuning from task-specific to transferable:** Prior to our study, only a few works studied the transferability of learned prompts between different tasks (Su et al., 2022; Vu et al., 2022; Lester et al., 2022). Specifically, Su et al. (2022) finds it is possible to transfer learnable prompts with additional fine-tuning on downstream task. However, as mentioned in the experiment section, all our reported results are zero-shot, i.e., no fine-tuning is needed to obtain transferability. Furthermore, Vu et al. (2022) finds that the learned prompt can only be transferred among similar tasks. However, as verified by our experiments, our learned prompts are transferable between datasets, tasks, and compressed models.
- **New avenue to enhance compressed LLMs accuracy:** We show that prompt tuning can effectively recover the accuracy drop of compressed LLMs. Traditional LLM compression methods often demand extensive engineering efforts and intricate designs, as seen in popular model compression papers like GPTQ (Frantar et al., 2022) and SparseGPT (Frantar & Alistarh, 2023). However, our method simplifies this process significantly. We learn soft prompts to effectively recover the accuracy drop in compressed LLMs, opening a new avenue to optimize the trade-off between accuracy and efficiency.

We emphasize that we intentionally kept our method as simple as possible. This simplicity underscores the unknown properties of soft prompts and uncovers new pathways for performance recovery of compressed LLMs. With extensive experiments, we showcase that our learnable soft prompt can restore the performance of LLMs with up to $8\times$ compression (with a joint 4-bit quantization and 50% weight pruning compression), allowing them to match their uncompressed counterparts on several standard benchmarks. Moreover, these findings are not only valid but also generalizable across various model families, datasets, and tasks, underscoring the broad applicability and impact of our work. Furthermore, we show that compared to other parameter-efficient fine-tuning methods like LoRA (Hu et al., 2021), our approach has less cost in recovering the performance of compressed LLMs.

2. Problem Statement and Related Work

In this section, we will begin by introducing the efficiency bottleneck of LLM inference. Then we will introduce current approximation approaches that are designed to reduce the computation and memory overhead and improve LLM inference latency. Finally, we will provide a review of recent progress that has been made in prompting LLMs.

2.1. Efficiency Bottleneck of LLM Inference

LLMs adopt a decoder-only, autoregressive approach where token generation is carried out step by step, with each token's generation dependent on the previously generated results. For instance, models such as GPT (Radford et al., 2018; 2019; Brown et al., 2020) follow this paradigm. A recent study by (Liu et al., 2023) investigates the inference process of OPT-175B models and finds that (1) token generation is the dominant factor contributing to the inference latency, and (2) Multilayer Perceptron (MLP) incurs higher I/O and computation latency compared to attention blocks during token generation. While system-level optimizations (Sheng et al., 2023; GitHub, 2023a;b) can enhance the inference time of LLMs, they do not directly mitigate the computation and memory I/Os involved in the LLM inference process.

2.2. Approximation in LLM Inference

In addition to optimizing at the system level, there are two primary approaches for reducing both computation and memory I/O to minimize the latency inference. (1) Sparse modeling: the general idea is to choose a particular set of weights in certain layers to minimize both computation and memory I/O (Frantar & Alistarh, 2023; Liu et al., 2023). These techniques are also closely related to pruning (He et al., 2018; Hubara et al., 2021b; Kwon et al., 2022; Hubara et al., 2021a) in the literature. Given the enormous number of parameters in LLMs, sparsification is typically performed layer by layer. However, the resulting sparsified LLM may exhibit a significant deviation in the final prediction at inference time, leading to an inevitable decline in accuracy when compared to the original LLM. (2) Quantization: it refers to the process of compressing trained weight values in LLMs into lower bits (Nagel et al., 2020; Dettmers et al., 2022; Xiao et al., 2022; Frantar et al., 2022). Empirical evaluations have shown that int8 quantization can provide a great approximation of the predictive performance of the original LLMs (Dettmers et al., 2022). However, there is a significant decline in accuracy when attempting to reduce the number of bits even further.

2.3. Prompt for LLMs

LLMs are known for their in-context learning ability, allowing them to generalize to unseen tasks without additional fine-tuning (Brown et al., 2020). Specifically, LLMs are controlled through user-provided natural language specifications of the task, or *prompts*, which illustrate how to complete a task. In this paradigm, we do not enforce modifications on the LLMs themselves. Instead, we focus on adapting the inputs to the LLMs for better predictive performance in downstream tasks. A typical strategy is to insert tokens before the input sequence to affect the attention mechanism. It has been shown in (Brown et al., 2020) that prompt engineering enables LLMs to match the performance of fine-tuned language models on a variety of language understanding tasks. Moreover, (Lester et al., 2021) empirically indicate that there is an equivalence between modifying the input and fine-tuning the model. Furthermore, (Su et al., 2022) studies the transferability of prompts across similar datasets or even tasks. Since then, we have witnessed the growth of prompt tuning infrastructure (Ding et al., 2022). However, we would like to emphasize that most of the current demonstrations of prompt tuning are task-specific (Li & Liang, 2021; Lester et al., 2021). When considering efficiency, it is desirable for a prompt to exhibit transferability across various settings.

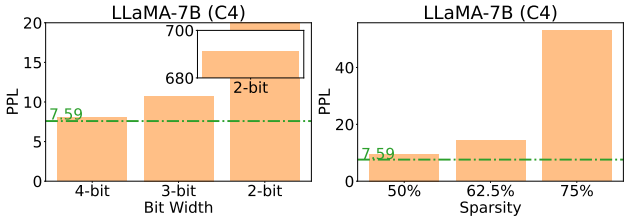
3. Motivation

The compression methods reduce the computational complexity at the cost of giving less accurate outputs. Thus, there naturally exists an **accuracy-efficiency trade-off**. In this section, we first empirically evaluate the trade-off of compressed LLMs. Then we found that for a compressed model, we can manually design a hard prompt that informs the model of its compressed state and helps it correct its predictions accordingly.

3.1. Performance Decline of LLMs After Compression

Experimental Setup. We assess the trade-off using LLaMA (Touvron et al., 2023a) on C4 dataset (Raffel et al., 2020). Here we adopt two representative post-training compression methods, i.e., GPTQ (Frantar et al., 2022) and SparseGPT (Frantar & Alistarh, 2023), to analyze the trade-off across various compression levels. We note that we choose post-training compression methods primarily for their ease of deployment. For the quantization method, we apply GPTQ to compress the model weights into 2, 3, and 4 bits integer numbers. As for the pruning method, we employ SparseGPT to eliminate 50%, 62.5%, and 75% of the model parameters. We would like to note that the post-training compression is conducted using the training set of C4, and subsequently, we evaluate the performance of the compression with the validation set of C4.

Quantitative Results. As shown in Figure 2, we visualize the evaluation perplexity (PPL) (Jelinek et al., 1977) versus the compression level. When we prune 50% of the parame-



(a) Quantization (b) Pruning

Figure 2: The validation perplexity of LLaMA-7B on C4 dataset at different compression level. The green line is the PPL of the original model.

ters or quantize the parameters to 4 bits, the PPL remains closer to that of the full LLaMA model. The PPL consistently increases as we decrease the allocated resource (e.g., bit-width/sparsity). Notably, the PPL will explode when the resource is below a certain threshold. For instance, the PPL shifts from 14 to 53 as sparsity increases from 62.5% to 75%. Moreover, the PPL grows significantly from around 11 to around 691 when we lower the quantization bits from 3-bit to 2-bit.

Qualitative Results. As shown in the left part of Figure 1, besides PPL, we also do a case study to understand how compression affects model generation results. In this example, the full model provides accurate answers to all three simple questions. Specifically, it correctly identifies Long Beach as a city in Los Angeles County, California, pinpoints Tulsa in northeastern Oklahoma, and describes asparagus as a spring vegetable belonging to the lily family. However, the pruned model with 62.5% weight sparsity struggles to generate meaningful responses. Instead of providing the requested information, its answers seem unrelated and tangential. For example, the pruned model responds with a statement about seeking a job when asked about Long Beach, mentions being a student at the University of Tulsa when asked about Tulsa's location, and admits uncertainty about Asparagus. This case study demonstrates that **aggressive model compression, such as the 62.5% weight sparsity applied to the pruned model, can lead to a significant degradation in the quality of generated responses.**

3.2. Prompt May Restores Compressed LLMs

In-context learning refers to the ability to adapt to the context provided within the input data through user-provided natural language specifications (Xie et al., 2022; Min et al., 2022), often referred to as *prompts*. Prompts serve to guide LLMs toward generating desired predictions by offering useful contextual information. As shown in Figure 1, the compressed model generates answers that are unrelated and off-topic when responding to these simple questions. Thus one natural question is, *for a compressed model, can we design a specific prompt that helps it correct its predictions accordingly?* Following the question, we manually design the hard prompt as "*Please carefully examine the weight matrix within the model, as it may contain errors. It is crucial to verify its accuracy and make any necessary adjustments to ensure optimal performance*". The results are shown in the fourth column of Figure 1. The observations are summarized as follows:

The prompted pruned model, i.e., "LLaMA-7B (62.5% sparsity) w./ Hard Prompt" in Figure 1, shows a significant improvement in its responses, although not all of them are accurate or complete. Specifically, (1) when explicitly told about its compressed state, the prompted pruned model correctly identifies that Long Beach is located in the United States. However, it does not provide further information about the city, such as its presence in Los Angeles County, California. (2) Regarding the second question about Tulsa, Oklahoma, the prompted pruned model fails to provide a relevant answer, instead repeating our prompt about the compression state, which is unrelated to the question. (3) When asked about asparagus, the prompted pruned model correctly identifies it as a plant used for cooking.

Insights. By explicitly informing the model of its compressed state, LLMs can generate more relevant responses for certain questions. The success of the designed prompt implies great potential: **With the correct input format, compressed LLMs can perform the same as their uncompressed counterparts.** However, despite the potential, as we analyzed at the beginning of this section, the manually designed prompt is not consistently effective. In other words, it only works for some problems, and not all answers generated are accurate or complete. We hypothesize that by involving the compressed weight in the prompt learning process, a learnable prompt could potentially surpass the performance of the hard prompt while still retaining the transferability aspects of the hard prompt.

4. Learning Prompt for Efficient LLM Inference

In this section, we will begin by introducing the formulation of the prompt learning paradigm. Then, we will shift our focus to the maximum likelihood objective of learning the prompt. Finally, we will delve into the transferability of the learned prompts.

4.1. Formulation

Section 3.2 has shown that incorporating prompts can enhance the predictive performance of compressed LLMs. However, discovering effective language-based prompts through trial and error is a cumbersome and inefficient process that requires exploring a vast vocabulary space. Therefore, this paper aims to develop a data-driven approach to

learning a soft prompt.

Typically an LLM would have a tokenizer that maps each input sentence into a sequence of integers $[x_0, x_1, \cdots, x_n]$. Afterwards, each token $x_i \in [v]$ represents a d-dimensional row vector in the embedding matrix $W \in \mathbb{R}^{v \times d}$. In the inference phase of LLM, we are given an input sequence $[x_0, x_1, \cdots, x_m]$ with m tokens. We would like to generate tokens after x_m step by step using an LLM. We denote prompt as a sequence of integers $[e_1, e_2, \cdots, e_k]$ with length k. Every token $e_j \in [k]$ represents a d-dimensional row vector in the prompt embedding matrix $E \in \mathbb{R}^{k \times d}$.

4.2. Learning Objectives

In this study, we present a prompt learning strategy that can be utilized as a post-training process for compressed LLMs. Given an LLM model with parameters denoted as θ, we start with either sparsification (Frantar & Alistarh, 2023; Liu et al., 2023) or quantization (Frantar et al., 2022) approach that compresses the model parameters. We denote the parameters after the compression as $\widetilde{\theta}$. We note that prompt learning is reliant on the data, and as such, we need to employ a text dataset X for this procedure. Next, for every sequence $[x_0, x_1, \cdots, x_n] \in X$, we insert k prompt tokens $[e_1, e_2, \cdots, e_k]$ before it. Next, we optimize the following objective.

$$\min_{E} \mathcal{L}_{\widetilde{\theta}} = \min_{E} \sum_{t=1}^{n} -\log \Pr_{\widetilde{\theta}}[x_t | e_1, \cdots, e_k, x_0, \cdots x_{t-1}]. \tag{1}$$

We note that the model parameter $\widetilde{\theta}$ is fixed and not updated. And the trainable parameters are the embedding of the prompt tokens $[e_1, e_2, \cdots, e_k]$, which are denoted by the matrix $E \in \mathbb{R}^{k \times d}$. Following (Lester et al., 2021), we initialize E such that each row in E corresponds to a vector randomly selected from the token embedding matrix W of the LLM. The prompt token sequence remains the same for all sequences in X. This means that we use the representation of prompt tokens to influence LLM's attention mechanisms between the tokens in the sequence $[x_0, x_1, \cdots, x_n]$. Specifically, the Eq (1) aims to maximize the likelihood of correctly predicting the next token in the sequence, given the preceding tokens. In this way, the learned prompt is aware of the compressed weights, as the gradient flows through these compressed weights during the optimization process. This allows the model to adapt its behavior to account for the compression effects while generating responses, potentially leading to improved performance.

4.3. Transferability of Learned Soft Prompt

The findings derived from Section 3.2 have provided us with a compelling impetus to delve into the exploration of the transferability of prompt tokens acquired through Eq (1). These prompt tokens, as well as their acquisition through one dataset, could have a significant impact on other NLP applications. In particular, we assess the transferability of prompt tokens across diverse datasets, various compression techniques and levels, as well as different tasks.

5. Experiment

In this section, we assess the effectiveness of our prompt strategy in enhancing the trade-off between accuracy and efficiency during LLM inference. We commence by outlining the experimental setup, followed by presenting the results of token generation. Furthermore, we investigate the transferability of prompts across different datasets and compression levels. For additional experiments related to transferability and efficiency, please refer to Appendix A, where we have included further details.

5.1. Experiment Setting

We use Nvidia RTX 8000 (48G) GPUs to conduct inference and prompt learning in LLMs. We use Common Crawl's web corpus (C4) (Raffel et al., 2020), Wikitext-2 (Merity et al., 2017), and the Penn Treebank (PTB) (Marcus et al., 1994) databases as language generation datasets. We set the sequence length for these datasets to 1024. For the token generation task, we use perplexity (PPL) (Jelinek et al., 1977) as the evaluation metric. We also introduce some downstream tasks to evaluate the cross-task transferability of the learned soft prompt. We will introduce the task information in the specific section. For LLMs, we adopted the Open Pre-trained Transformer (OPT) Language Models (Zhang et al., 2022), Large Language Model Architecture (LLaMA) (Touvron et al., 2023a), LLaMA-2 (Touvron et al., 2023b) and bllom (Workshop et al., 2022). To compress the LLMs, we employed techniques from both SparseGPT (Frantar & Alistarh, 2023) and GPTQ (Frantar et al., 2022) methodologies. We refer the readers to Appendix A.1 for more experimental details.

5.2. Token Generation Results

On the C4 training set, we compress the OPT-1.3B, OPT-2.7B, OPT-6.7B, and LLaMA-7B using SparseGPT (Frantar & Alistarh, 2023). We utilize sparsity levels of 50%, 62.5%, and 75% for compression. Additionally, we employ GPTQ (Frantar et al., 2022) for 2-bit, 3-bit, and 4-bit quantization. Furthermore, prompt learning is applied to each compressed model using the methodology introduced in Eq (1). We set k in Eq. 1 to 100, i.e., incorporating 100 learnable prompt tokens. We also conduct the ablation on the impact of the number of soft tokens in Appendix A.6.

Figure 3 shows the impact of our approach on the validation

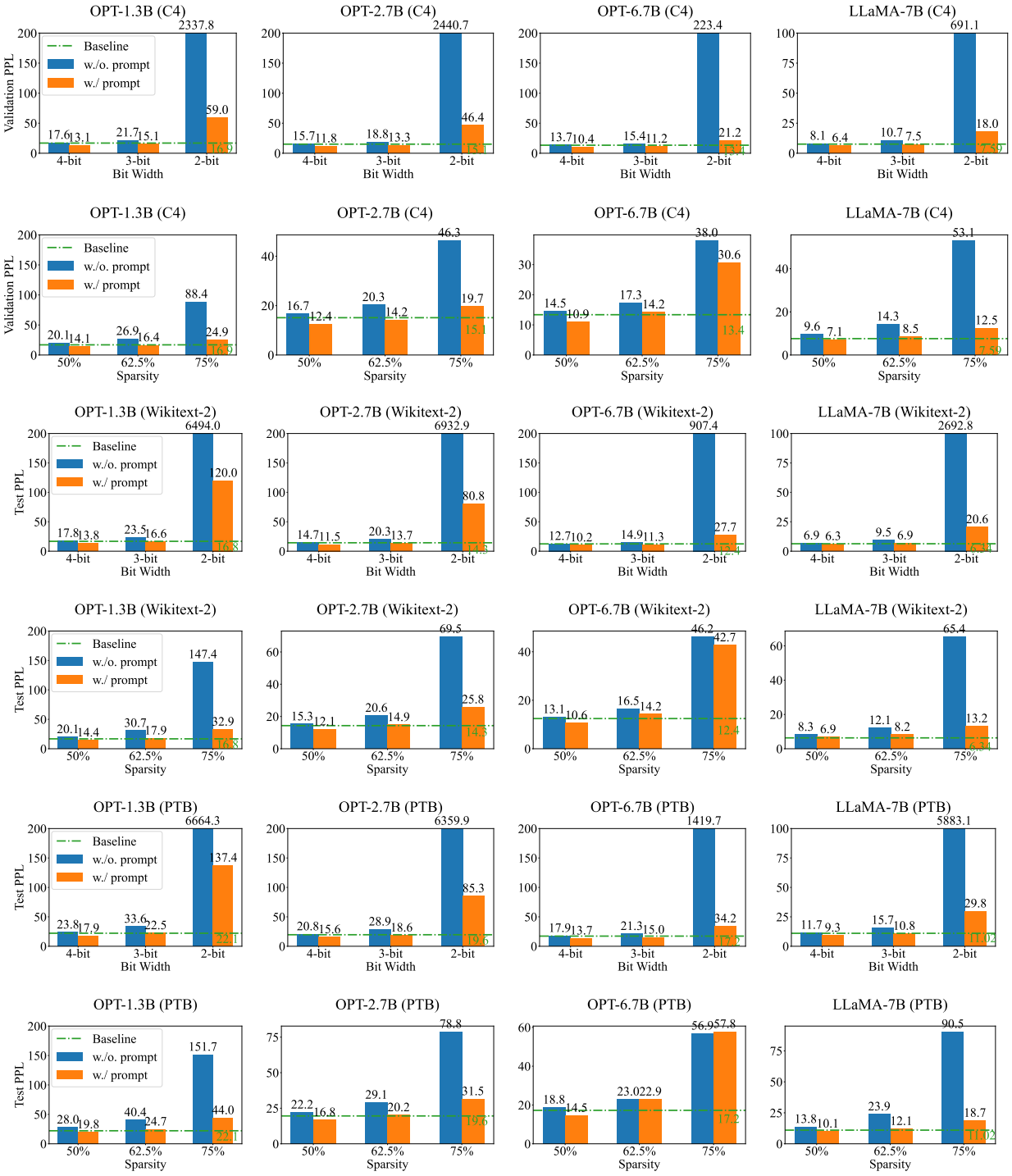


Figure 3: OPT-1.3B, OPT-2.7B, OPT-6.7B, and LLaMA-7B on C4 , Wikitext-2 and PTB test set at different bit-width and sparsity. **We note that we learn prompts only through C4 and then transfer the prompts to Wikitext-2 and PTB.** Here the "Baseline" (green line) represents the uncompressed model.

Table 1: The validation PPL of LLaMA-2-13B and Bloom-7B models on C4 dataset.

Dataset	Model	Precision	Recover method	Trainable Params (M)	PPL
C4	LLaMA-2-13B	fp16	NA	NA	6.96
C4	LLaMA-2-13B	3bit	NA	NA	9.24
C4	LLaMA-2-13B	3bit	Soft Prompt	0.5	**6.75**
C4	LLaMA-2-13B	3bit	LoRA	26	8.15
C4	BLOOM-7B	fp16	NA	NA	15.87
C4	BLOOM-7B	3bit	NA	NA	18.40
C4	BLOOM-7B	3bit	Soft Prompt	0.4	**13.54**
C4	BLOOM-7B	3bit	LoRA	15.7	17.26

set of C4. We observe a significant improvement in PPL across all compression levels. Firstly, by employing soft prompt tokens, the compressed LLMs using SparseGPT with 50% sparsity even outperform the full model counterparts, exhibiting lower PPL. This trend is also observed in the 4-bit quantization of LLMs using GPTQ. Secondly, even with further enhanced compression, the compressed LLMs with soft prompt tokens learned from Eq (1) maintain comparable PPL to their original counterparts. Notably, prompts learned from each of the four 3-bit quantized models aid in surpassing the performance of their respective full model counterparts. We also observe a similar effect in models with 62.5% sparsity for OPT-1.3B and OPT-2.7B. Conversely, prompts learned from both OPT-6.7B and LLaMA-7B assist in achieving the same PPL as their full model counterparts. Lastly, our approach significantly enhances the predictive performance of extreme scale compression. In both SparseGPT with 75% sparsity and GPTQ with 2-bit quantization, we find that the prompt learning strategy substantially improves the PPL across all four models. For example, prompts learned over the 2-bit GPTQ compression of OPT-1.3B reduce the PPL from 2337.8 to 59.

5.3. Soft Prompt Does Transfer

Intuitively, a model compressed using one dataset should achieve decent predictive performance when transferred to other datasets (Frantar et al., 2022; Frantar & Alistarh, 2023). Here we assess whether the prompt tokens learned from one dataset exhibit similar transferability across different datasets. Specifically, we first compress a model with SparseGPT or GPTQ using C4 training set. We then learn the prompt with the compressed model on C4 training set. Finally, we evaluate the performance of this compressed model with and without the learned prompts on other datasets, e.g., Wikitext-2 and PTB dataset. **We emphasize the entire process does not involve any task-specific data, and our results thus remain "zero-shot".**

Figure 3 presents the performance of OPT-1.3B, OPT-2.7B, OPT-6.7B, and LLaMA-7B on the test set of Wikitext-2 and the PTB dataset. For each LLM model, we also include the performance of its compressed versions with 50%, 62.5%, and 75% sparsity using SparseGPT. Additionally, we include the performance of each model's compressed version with 2-bit, 3-bit, and 4-bit quantization using GPTQ. The figures demonstrate the consistent advantages of prompt tokens across the two datasets. For every model with 50% sparsity or 4-bit quantization, learning prompts from the C4 dataset result in a lower PPL compared to the full model counterpart. Moreover, we observe a substantial improvement in PPL when using learned prompt tokens as the model becomes more compressed. This phenomenon validates that the prompts learned on top of compressed models can be

Table 2: Perplexity comparison between full LLaMA-7B model its quantized versions with different prompts, where we report test perplexity on PTB and Wikitext-2 dataset. "w./o. prompt" refers to the quantized model without soft prompts."w./ direct prompt" means the soft prompts are directly trained on the target dataset."w./ transferred prompt" means the prompt is trained on C4 dataset and then transferred to the target dataset.

Model		PTB	Wikitext2
LLaMA-7B		11.02	6.33
LLaMA-7B w./ direct prompt		6.86	5.57
4-bit	w./o. prompt	11.65	6.92
	w./ direct prompt	7.04	5.88
	w./ transferred prompt	9.25	6.26
3-bit	w./o. prompt	15.74	9.45
	w./ direct prompt	7.76	6.33
	w./ transferred prompt	10.81	6.90
2-bit	w./o. prompt	5883.13	2692.81
	w./ direct prompt	14.98	16.67
	w./ transferred prompt	29.82	20.56

Table 3: The zero-shot test PPL of transferred soft prompt and LoRA on Wikitext2 dataset.

Dataset	Model	Precision	Method	Transferred Params (M)	PPL
Wikitext2	LLaMA-2-13B	fp16	NA	NA	5.58
Wikitext2	LLaMA-2-13B	3bit	NA	NA	7.88
Wikitext2	LLaMA-2-13B	3bit	Soft Prompt	0.5	**5.89**
Wikitext2	LLaMA-2-13B	3bit	LoRA	26	7.07
Wikitext2	BLOOM-7B	fp16	NA	NA	13.26
Wikitext2	BLOOM-7B	3bit	NA	NA	16.06
Wikitext2	BLOOM-7B	3bit	Soft Prompt	0.4	**12.42**
Wikitext2	BLOOM-7B	3bit	LoRA	15.7	15.65

effectively transferred across datasets.

We also compare the transferred soft prompts against the soft prompts that are directly trained on the downstream dataset. Given direct prompt receives a domain-specific loss, our transferred prompt is, as expected, not as competitive as the direct one. However, such transferred prompt may significantly bridge the gap between a compressed and full model — e.g., our 3-bit & 4-bit quantized LLaMA-7B with transferred prompt can deliver on-par or better PPL than the full model on PTB and Wikitext2. We'd say this is an especially worthy contribution in practice, as one may possibly download the open-sourced transferable prompt to help on a compressed model with little effort.

Here we emphasize that the prompt trained with a domain-specific loss can no longer be transferred between different datasets. Below we present the results of transferring the soft prompts learned on Wikitext2 (featured articles on Wikipedia) to PTB (Wall Street Journal material) and C4 (collection of common web text corpus). The results, as shown in the table below, highlight a significant disparity in performance when using domain-specific prompts across different domains. The prompt trained on Wikitext-2, when applied to PTB and C4, leads to a drastic increase in perplexity, indicating a severe degradation in model performance. In contrast, if the prompt is learned on general text datasets like C4, then it can be transferred to different domains e.g., PTB and Wikitext2, and tasks, e.g., QA and language understanding (Appendix A.4).

5.4. Combination of Sparsification and Quantization

In this section, we explore the effectiveness of the prompt strategy in the combination of sparsification and quantization for compressing LLM. Since sparsification and quantization target different aspects of compression, it is natural to combine them to achieve better efficiency. Table 4 presents the PPL before and with, and without the learned prompt on the validation set of C4, as well as the test sets of Wikitext-2 and PTB. We choose the LLaMA-7B model compressed using 50% sparsity and 4-bit quantization from the training set of C4. We should note that the prompt learning process also takes place on the training set of C4. Our results demonstrate that the prompt learning strategy remains effective when combining sparsification and quantization. Additionally, with the prompt, the 50% sparse and 4-bit compressed model still performs comparably to the original LLaMA-7B.

Table 4: The PPL of joint 50% sparsity + 4-bit quantization with learned prompts on the validation set of C4 and a test set of Wikitext-2 and PTB. The prompt is learned on C4 training set.

Models	C4	Wikitext-2	PTB
Full	7.59	6.34	11.02
50% + 4-bit (w./o. prompt)	10.94	9.67	17.39
50% + 4-bit (w./ prompt)	**7.38**	7.31	**10.64**

5.5. Soft Prompt Outperforms LoRA

We perform our methods and LoRA (Hu et al., 2021; Dettmers et al., 2024) on LLaMA-2-13B and BLOOM-7B models. The results are summarized on Table 1. Following LoRA experimental setting (Hu et al., 2021), we insert LoRA layers to all query and value layers with a rank $r = 32$, $\alpha = 32$, and a 0.1 dropout rate. We train the Lora using the Adam optimizer with a $2e-4$ learning rate. It suggests that our approach outperforms LoRA with lower PPL. Moreover, we can recover the performance of GPTQ 3-bit LLaMA-2-13B and GPTQ 3-bit BLOOM-7B with even better performance than their fp16 counterparts.

We also test the transferability of ours and LoRA on wikitext2 dataset. We summarize the results in Table 3 and Table 7 in the appendix. It suggests that our soft prompt can be transferred to other datasets while still outperforming LoRA for the performance recovery of compressed LLMs.

6. Conclusion

In this paper, we optimize the trade-off between computational efficiency and accuracy in LLMs via prompting compressed models. We propose a soft prompt learning method

where we expose the compressed model to the prompt learning process. Our experimental analysis suggests our soft prompt strategy greatly improves the performance of the compressed models, allowing them to match their uncompressed counterparts. The research also highlights the transferability of these learned prompts across different datasets, tasks, and compression levels.

Acknowledgement

Zhaozhuo Xu is supported by the startup fund of the Stevens Institute of Technology and the FEDML Large Language Model Research Fund. Beidi Chen is supported by a research gift from Moffett.AI. Anshumali Shrivastava is supported by NSF-SHF-2211815 and NSF-2336612. Xia Hu is supported by the US Department of Transportation (USDOT) Tier-1 University Transportation Center (UTC) Transportation Cybersecurity Center for Advanced Research and Education (CYBER-CARE) grant #69A3552348332.

Impact Statement

In this paper, we introduce research aimed at pushing the boundaries of LLM to make it affordable to everyone. Our work carries numerous potential societal implications, though we believe it is unnecessary to emphasize any specific ones in this context.

References

Bisk, Y., Zellers, R., Bras, R. L., Gao, J., and Choi, Y. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 7432–7439. AAAI Press, 2020. URL https://ojs.aaai.org/index.php/AAAI/article/view/6239.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Chen, L., Zaharia, M., and Zou, J. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*, 2023.

Dettmers, T., Lewis, M., Belkada, Y., and Zettlemoyer, L. Llm. int8 (): 8-bit matrix multiplication for transformers at scale. *arXiv preprint arXiv:2208.07339*, 2022.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36, 2024.

Ding, N., Hu, S., Zhao, W., Chen, Y., Liu, Z., Zheng, H., and Sun, M. Openprompt: An open-source framework for prompt-learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 105–113, 2022.

Frantar, E. and Alistarh, D. Sparsegpt: Massive language models can be accurately pruned in one-shot, 2023.

Frantar, E., Ashkboos, S., Hoefler, T., and Alistarh, D. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.

Gao, L., Tow, J., Biderman, S., Black, S., DiPofi, A., Foster, C., Golding, L., Hsu, J., McDonell, K., Muennighoff, N., Phang, J., Reynolds, L., Tang, E., Thite, A., Wang, B., Wang, K., and Zou, A. A framework for few-shot language model evaluation, September 2021. URL https://doi.org/10.5281/zenodo.5371628.

GitHub. https://github.com/mlc-ai/mlc-llm, 2023a.

GitHub. https://github.com/mlc-ai/web-llm, 2023b.

Gugger, S., Debut, L., Wolf, T., Schmid, P., Mueller, Z., and Mangrulkar, S. Accelerate: Training and inference at scale made simple, efficient and adaptable. https://github.com/huggingface/accelerate, 2022.

He, Y., Lin, J., Liu, Z., Wang, H., Li, L.-J., and Han, S. Amc: Automl for model compression and acceleration on mobile devices. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 784–800, 2018.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Hu, E. J., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021.

Hubara, I., Chmiel, B., Island, M., Banner, R., Naor, J., and Soudry, D. Accelerated sparse neural training: A provable and efficient method to find n: m transposable masks. *Advances in Neural Information Processing Systems*, 34: 21099–21111, 2021a.

Hubara, I., Nahshan, Y., Hanani, Y., Banner, R., and Soudry, D. Accurate post training quantization with small calibration sets. In *International Conference on Machine Learning*, pp. 4466–4475. PMLR, 2021b.

Jelinek, F., Mercer, R. L., Bahl, L. R., and Baker, J. K. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63, 1977.

Kwon, W., Kim, S., Mahoney, M. W., Hassoun, J., Keutzer, K., and Gholami, A. A fast post-training pruning framework for transformers. *arXiv preprint arXiv:2204.09656*, 2022.

Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, 2021.

Lester, B., Yurtsever, J., Shakeri, S., and Constant, N. Reducing retraining by recycling parameter-efficient prompts. *arXiv preprint arXiv:2208.05577*, 2022.

Li, X. L. and Liang, P. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

Liu, Z., Wang, J., Dao, T., Zhou, T., Yuan, B., Song, Z., Shrivastava, A., Zhang, C., Tian, Y., Ré, C., and Chen, B. Deja vu: Contextual sparsity for efficient llms at inference time. In *International Conference on Machine Learning*. PMLR, 2023.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Marcus, M., Kim, G., Marcinkiewicz, M. A., MacIntyre, R., Bies, A., Ferguson, M., Katz, K., and Schasberger, B. The penn treebank: Annotating predicate argument structure. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*, 1994.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Byj72udxe.

Mihaylov, T., Clark, P., Khot, T., and Sabharwal, A. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391, 2018.

Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., and Zettlemoyer, L. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022.

Nagel, M., Amjad, R. A., Van Baalen, M., Louizos, C., and Blankevoort, T. Up or down? adaptive rounding for post-training quantization. In *International Conference on Machine Learning*, pp. 7197–7206. PMLR, 2020.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. Improving language understanding by generative pre-training. 2018.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

Sheng, Y., Zheng, L., Yuan, B., Li, Z., Ryabinin, M., Fu, D. Y., Xie, Z., Chen, B., Barrett, C., Gonzalez, J. E., and othersi. High-throughput generative inference of large language models with a single gpu. In *International Conference on Machine Learning*. PMLR, 2023.

Su, Y., Wang, X., Qin, Y., Chan, C.-M., Lin, Y., Wang, H., Wen, K., Liu, Z., Li, P., Li, J., et al. On transferability of prompt tuning for natural language processing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3949–3969, 2022.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Vu, T., Lester, B., Constant, N., Al-Rfou, R., and Cer, D. Spot: Better frozen model adaptation through soft prompt transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5039–5059, 2022.

Wang, Z., Jia, Z., Zheng, S., Zhang, Z., Fu, X., Ng, T. E., and Wang, Y. Gemini: Fast failure recovery in distributed training with in-memory checkpoints. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 364–381, 2023a.

Wang, Z., Wu, X., Xu, Z., and Ng, T. Cupcake: A compression scheduler for scalable communication-efficient distributed training. *Proceedings of Machine Learning and Systems*, 5, 2023b.

Workshop, B., Scao, T. L., Fan, A., Akiki, C., Pavlick, E., Ilić, S., Hesslow, D., Castagné, R., Luccioni, A. S., Yvon, F., et al. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022.

Wu, B., Zhong, Y., Zhang, Z., Huang, G., Liu, X., and Jin, X. Fast distributed inference serving for large language models. *arXiv preprint arXiv:2305.05920*, 2023.

Xiao, G., Lin, J., Seznec, M., Demouth, J., and Han, S. Smoothquant: Accurate and efficient post-training quantization for large language models. *arXiv preprint arXiv:2211.10438*, 2022.

Xie, S. M., Raghunathan, A., Liang, P., and Ma, T. An explanation of in-context learning as implicit bayesian inference. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=RdJVFCHjUMI.

Yuan, B., He, Y., Davis, J., Zhang, T., Dao, T., Chen, B., Liang, P. S., Re, C., and Zhang, C. Decentralized training of foundation models in heterogeneous environments. *Advances in Neural Information Processing Systems*, 35: 25464–25477, 2022.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, 2019.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

Appendix

A. More Experiments

A.1. Experiment Details

In the experiment, we employed the AdamW (Loshchilov & Hutter, 2019) optimizer as our chosen optimizer. We conducted iterative prompt updates using a batch size of 4, a weight decay of 10^{-5}, and a learning rate of 10^{-3}. We set the total optimization steps as 30,000 and use the model corresponding to the best validation perplexity as the final model. To facilitate mix-precision training and system-level optimization, we leveraged the accelerate library (Gugger et al., 2022).

All experiments are conducted on a server with eight Nvidia RTX 8000 (48G) GPUs, 1.5T main memory, and two AMD EPYC 7742 64-Core Processors. We note that the whole prompt tuning process can be done in five hours with four RTX 8000 (48G) GPUs. The software and package versions are specified in Table 5. In the future, we would like to further accelerate the our prompt learning for larger LLMs with efficient distributed learning strategy (Wang et al., 2023a;b)

Table 5: Package configurations of our experiments.

Package	Version
CUDA	11.6
pytorch	2.0.1
transformers	4.30.0.dev0
accelerate	0.18.0

A.2. Followup on Cross-Dataset Transferability

In this section, we provide more experiments on transferring the learned prompts across different datasets.

Table 6: Perplexity comparison of transferring prompts learned on Wikitext2 to PTB and C4.

Model	**PTB**	**C4**
Full Model	11.02	7.59
3-bit w./o. prompt	15.74	10.74
3-bit w./ prompt learned on Wikitext2	294.16	160.64
3-bit w./ prompt learned on C4	**10.81**	**7.48**

A.3. Ablation on the Cross-Compression Transferability

Here we assess the transferability of learned prompts across various compression levels. Specifically, we aim to address the following questions: Can the prompt learned from a compressed model be applied to the same model but compressed at different levels or types?

In Figure 4, we display the Perplexity (PPL) outcomes on the C4 validation set, along with the results on the Wikitext-2 and PTB test sets. These results are obtained by applying prompts learned from a source compressed model to a different target compressed model. Here, "target" denotes the specific compression type and degree used in the model receiving the prompts.

Table 7: The zero-shot test PPL of transferred soft prompt and LoRA on PTB dataset.

Dataset	Model	Precision	Method	PPL
PTB	LLaMA-2-13B	fp16	NA	50.33
PTB	LLaMA-2-13B	3bit	NA	82.31
PTB	LLaMA-2-13B	3bit	Soft Prompt	39.66
PTB	BLOOM-7B	fp16	NA	22.77
PTB	BLOOM-7B	3bit	NA	28.32
PTB	BLOOM-7B	3bit	Soft Prompt	21.03

While "source" refers to the compression type and degree of the model from which the prompts are originally learned. For example, "source 4-bit" indicates that the prompt is learned from a compressed model with 4-bit quantization. Based on the figures, we observe that (1) For sparse LLMs, prompts learned from higher sparsity can be effectively transferred to models with lower sparsity, while achieving comparable performance.. (2) For quantized LLMs, prompts learned from lower bit quantization levels can be successfully applied to models with higher bit quantization, while achieving comparable performance. (3) There is a certain degree of transferability of prompts learned between different compression types, especially when the compression level is less. For instance, a prompt learned from a LLaMA-7B model with 4-bit quantization can be transferred to a LLaMA-7B model with 50% sparsity.

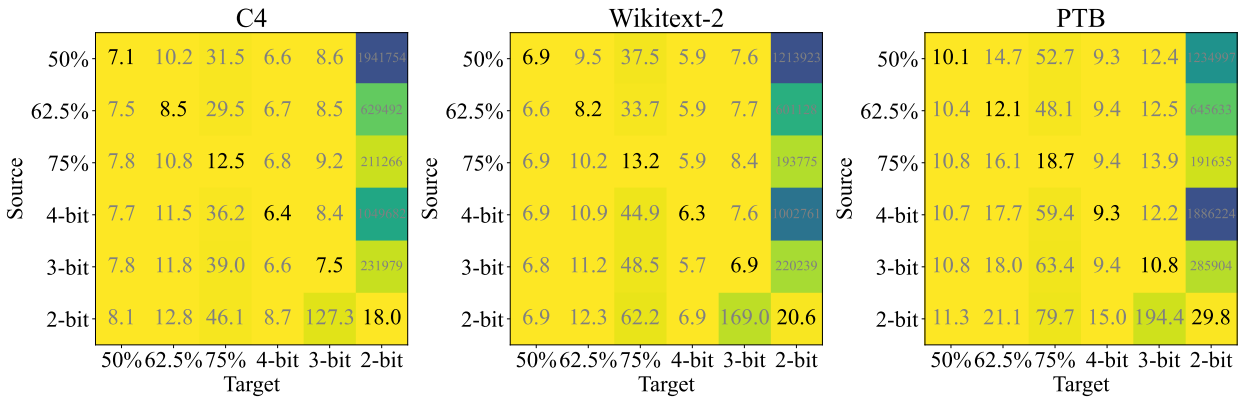


Figure 4: LLaMA-7B transfer between different sparsity and bit-width. The "target" refers to the compression type and level for the compressed model, while the"source" represents the type and level of the compressed model from which the prompt is learned. For example, "4-bit" in source indicates that the prompt is learned from a compressed model with 4-bit quantization.

A.4. Cross-Task Transferability

In this section, we explore the transferability of learned prompts across different tasks. Specifically, we aim to assess the effectiveness of prompts learned from token generation tasks, as indicated by Eq (1), in downstream tasks of LLM. As an illustrative example, we consider the zero-shot generalization tasks of LLaMA-7B (Touvron et al., 2023a). For evaluation purposes, we have chosen OpenbookQA (Mihaylov et al., 2018), Hellaswag (Zellers et al., 2019), PIQA (Bisk et al., 2020), and the high school European history task from (Hendrycks et al., 2020). The European history task is particularly interesting due to its inclusion of a lengthy context sentence for each question. We employ the lm-evaluation-hardness framework (Gao et al., 2021), incorporating adapters from (Yuan et al., 2022), for the purpose of conducting the experiment.

Table 8 presents the results in terms of normalized accuracy, and we also include the standard deviation, as indicated by (Gao et al., 2021). The table clearly demonstrates that the learned prompt significantly enhances the accuracy of these tasks. These findings imply that prompts acquired through token generation tasks can effectively enhance the accuracy-efficiency trade-off of compressed LLMs.

A.5. Efficiency Profiling

In this section, we analyze how the inclusion of prompt tokens impacts the latency of LLM inference. Figure 5 illustrates the latency of three OPT models and the LLaMA-7B model utilized in this paper, considering the insertion of additional prompt tokens with varying lengths. For token generation, we set the sequence length to 1024. The figure demonstrates that the addition of prompt tokens does not significantly increase the latency of LLM inference, particularly when the inserted tokens account for less than 10% of the original sequence length. Furthermore, our observations indicate that the latency does not exhibit a linear correlation with the length of the inserted tokens, highlighting the effectiveness of the prompt in facilitating efficient LLM inference.

A.6. Ablation on the Number of Soft Tokens

In Table 9, we conduct the ablation study on the impact of the number of soft tokens using 3-bit quantized LLama-7B on PTB dataset. We observe that there is still a significant improvement with 25 prompt tokens, and we can improve the

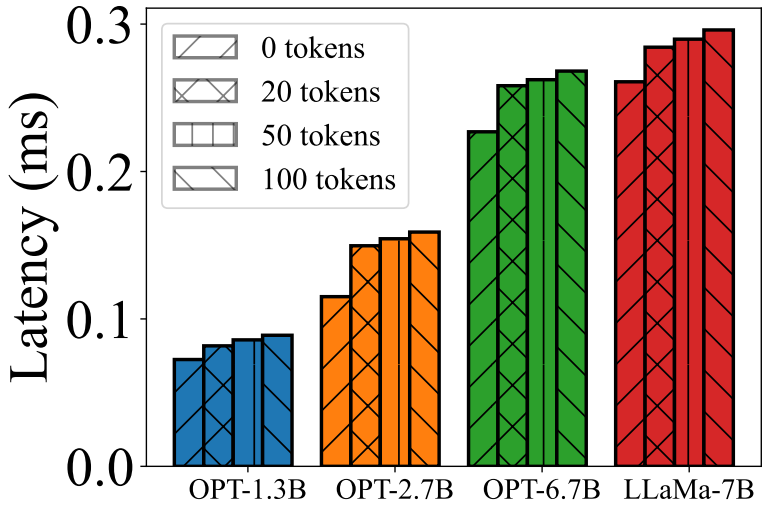


Figure 5: Latency benchmark of inference speed with prompt tokens

Table 8: The zero-shot results on transforming the learned prompt to OpenBookQA, Hellaswag, PIQA, and High School European History dataset.

Models		OpenbookQA	Hellaswag	PIQA	High School European History
Full		0.410±0.022	0.497±0.005	0.702±0.011	0.364±0.038
50%	w./o. Prompt	0.412±0.022	0.449±0.005	0.682±0.011	0.364±0.038
	+ Learned Prompt	0.400±0.022	0.469±0.005	0.689±0.011	0.358±0.037
62.5%	w./o. Prompt	0.396±0.022	0.380±0.005	0.638±0.011	0.345±0.037
	+ Learned Prompt	0.402±0.022	0.433±0.005	0.668±0.011	0.345±0.037
75%	w./o. Prompt	0.366±0.022	0.280±0.004	0.549±0.012	0.315±0.036
	+ Learned Prompt	0.358±0.021	0.344±0.005	0.614±0.011	0.358±0.037
4-bit	w./o. Prompt	0.410±0.022	0.487±0.005	0.690±0.011	0.358±0.037
	+ Learned Prompt	0.418±0.022	0.487±0.005	0.692±0.011	0.352±0.037
3-bit	w./o. Prompt	0.378±0.022	0.446±0.005	0.674±0.011	0.358±0.037
	+ Learned Prompt	0.404±0.022	0.459±0.005	0.688±0.011	0.358±0.037
2-bit	w./o. Prompt	0.354±0.021	0.240±0.004	0.491±0.012	0.315±0.036
	+ Learned Prompt	0.350±0.021	0.294±0.005	0.563±0.012	0.333±0.037

performance by increasing the prompt size.

Table 9: Ablation study on the impact of the number of soft tokens using 3-bit quantized LLama-7B on PTB dataset.

# tokens	Perplexity
Baseline (0 tokens)	15.74
25 tokens	9.26
50 tokens	8.61
75 tokens	8.17
100 tokens	7.76

B. Discussion

Limitations. One limitation of our study is its reliance on GPUs for executing computational tasks. It is crucial to acknowledge that GPUs can be expensive to procure and maintain, thus imposing financial constraints on researchers or organizations with limited resources. In order to address this issue, future endeavors should investigate alternative computational architectures or optimizations that can alleviate the dependence on costly GPUs. By doing so, the accessibility and applicability of our proposed methodology can be expanded, making it more widely accessible to a broader range of researchers and organizations.

Potential Negative Societal Impacts. While our research primarily centers on diminishing the energy consumption of LLM during inference, it is crucial to acknowledge that the carbon emissions stemming from LLM inference may still contribute to environmental concerns. As part of our future endeavors, we aspire to enhance the efficiency of LLM inference on low-energy devices.

C. More Visualization

In this section, we present further visualizations of compression-aware prompts, as demonstrated in Figure 1 in Section 1. The results unveil a significant improvement achieved by utilizing a hard, task-independent prompt on compressed LLMs. Additionally, we showcase the visualization of responses generated using our prompt derived from the C4 training set. It is worth noting that, in certain instances, the task-independent and learned prompt outperforms the hard prompt.

Figure 6: Case study for the effect of prompts on a pruned LLaMA-7B with a 62.5% weight sparsity.

Question	Full Model	LLaMa-7B (4 Bit)	LLaMa-7B (4 Bit) w./ Hard prompt	LLaMa-7B (4 Bit) w./ Learned Prompt
Q: Please give the answers to this question: Where is Boston?	Boston is a city in the state of Massachusetts. It is located on the east coast of the United States.	Boston is a city in the United States. It is the capital of Massachusetts.	I'm sorry, but I don't understand your question.	Boston is a city in the United States. It is the capital of the state of Massachusetts.
Q: Please give the answers to this question: Where is Memphis?	Memphis is a city in the state of Tennessee. It is located on the Mississippi River.	Memphis is a city in the state of Tennessee. It is located on the Mississippi River.	The answer is: Memphis is in Tennessee.	Memphis is a city in the state of Tennessee. It is located on the Mississippi River.
Q: Please give answers to this question: Where is Colorado Springs?	Colorado Springs is located in the south central part of the state of Colorado. It is located at the base of the Rocky Mountains.	I'm not sure if this is the right place to ask this question, but I'm going to try. I'm a 16 year old girl who is going to Colorado Springs for a week in June...	The answer is Colorado Springs, Colorado.	The city of Colorado Springs is located in the central part of the state of Colorado. It is located in the southwestern part of the state.
Q: Please give the answers to this question: Where is Houston?	Houston is in Texas. Houston is the largest city in Texas. Houston is the fourth largest city in the United States.	Houston is a city in Texas. Houston is the fourth largest city in the United States.	I'm sorry, but I don't understand your question.	Houston is a city in Texas. Houston is the fourth largest city in the United States. Houston is the largest city in Texas.
Q: Please give answers to this question:Where is Omaha?	Omaha is located in Nebraska. It is the largest city in Nebraska.	Omaha is located in Nebraska. It is the largest city in Nebraska.	The answer is: Omaha is located in Nebraska.	Omaha is located in Nebraska. It is the largest city in Nebraska.

Figure 7: Case study for the effect of prompts on a pruned LLaMA-7B with a 4-bit quantization.

D. Understanding The Learned Prompts From Natural Language Aspect

With the learned prompt outperforming the hard counterpart, we raise an intriguing question: How do the learned prompt tokens look when viewed from the perspective of natural language? In this section, we present the ablation study to answer the above question. Specifically, for each of the learned prompt token embeddings, we identify the words whose embedding is closest to the learned prompt token embedding via the nearest neighbor search technique, where the similarity measure is cosine similarity. In Figure 8, we plot the histogram of the cosine similarity between each learned prompt token and the top-100 nearest embeddings to it, where the prompt is learned with a pruned LLaMA-7B with a 50% weight sparsity. **We observe that there is no word whose embedding closely matches the learned one within the embedding space.** The cosine similarity for nearly all comparisons falls below 0.16, suggesting a considerable disparity between the learned prompt embeddings and their nearest equivalents. Below we also report the nearest word for each of the learned prompt token embedding. We observe that (1) almost all of them are meaningless. (2) several learned prompt tokens may be mapped to the same word.

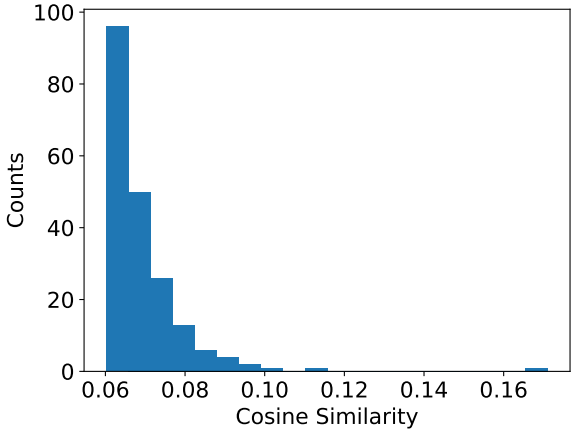


Figure 8: The distribution of the cosine similarity between the learned embedding and the top-100 nearest embeddings to it.

Nearest word for each of the learned prompt token[i]**:** "heits", "<s>", "</s>", "<0x00>", "<0x01>", "<0x02>", "<0x03>", "<0x04>", "<0x05>", "<0x06>", "<0x07>", "<0x08>", "<0x09>", "<0x0A>", "<0x0B>", "<0x0C>", "<0x1A>", "<0x0E>", "<0x0F>", "<0x10>", "<0x11>", "<0x12>", "<0x13>", "<0x14>", "<0x15>", "<0x16>", "<0x17>", "<0x18>", "<0x19>", "<0x1A>", "<0x1B>", "<0x1C>", "<0x1D>", "<0x1E>", "<0x1F>", "sep", ";;;;", "état", "<0xB1>", "▁Ej", "moz", "▁diverse", "▁"", "argument", "|", "han", "ura", "/", "-", "<0xE7>", "▁Lisa", "▁case", "ura", "O","▁Chal", "▁Chan", "O", "asc", "Client", "▁Det", "O", "▁Hel", "▁L", "▁Pel", "▁k", "▁It", "O", "<0x8B>", "<0x00>", "ILL", "O", "E", "ren", "ety", "cy", "</s>", "<0x8B>", "<0x9F>", "<s>", "<s>", "IM", "<s>", "."

Our ablation study highlights the hardness of understanding the mechanisms underlying learned prompts. This area remains largely uncharted, inviting future research to uncover its intricacies. Our hope is that this study will ignite curiosity and foster continued scholarly pursuit in this field.

[i]here we did not display the word that is not in UTF-8 format.