Sparse KD: Knowledge Distillation for Sparse Models in Constrained Fine-tuning Scenarios

Anonymous– ACL submission

Abstract

Large language models (LLMs) have achieved tremendous success in various domains, but their massive parameter sizes pose challenges for fine-tuning and inference. Recently, the common model compression process involves obtaining a sparse LLM through pruning, followed by LoRA-finetuning. However, these methods often suffer from significant performance degradation. We attempted to address this by introducing additional teacher distillation, but found limited improvements due to the gap between the teacher and student models and constrained training iterations. To overcome these challenges, we propose Sparse KD, the first distillation framework specifically designed for sparse models in constrained fine-tuning scenarios. Our framework includes dynamic temperature, knowledge alignment, and Bayesian distillation optimization strategies. Dynamic temperature can adaptively align the strength of the teacher's knowledge, and the Knowledge Alignment Module can bridge the gap by projecting teacher-student knowledge to the same interval. Applying Bayesian optimization swiftly finds optimal settings based on these strategies, thereby improving model performance. Comprehensive experiments across diverse task types have demonstrated that this combination can be applied to LLMs with effective and stable results.

1 Introduction

Large language models (LLMs) revolutionize natural language processing (NLP) by achieving remarkable performance across domains such as machine translation, sentiment analysis, question answering, and text generation (Touvron et al., 2023b,c; Chiang et al., 2023; Scao et al., 2022; Zhang et al., 2022). However, their massive parameter sizes pose challenges for fine-tuning and deployment in real-world applications. For example, GPT3, one of the top-performing models, contains 175 billion parameters, requiring approximately 350GB of GPU memory in FP16 for model storage and inference (Brown et al.). Meeting the computational demands of these models while efficiently handling their multitude of parameters presents significant processing time and resource allocation challenges.

The prevailing methodology for mitigating the computational burden of LLMs involves compressing the models through various techniques (Frantar and Alistarh, 2023; Sun et al., 2023; Ma et al., 2023a; Kwon et al., 2022). These approaches include model pruning, Knowledge Distillation (KD), parameter quantization, and Low-Rank Adaptation (LoRA), which are all tailored to LLMs. Among these, model pruning is a widely adopted compression technique that eliminates insignificant parameters based on their magnitude, resulting in sparse LLMs that offer improved efficiency during inference. To further optimize the performance of sparse LLMs, fine-tuning techniques like LoRA have been proposed to adapt pruned models for specific downstream tasks. The conventional approach of pruning followed by fine-tuning results in significant performance degradation due to the loss of critical knowledge during pruning and the limited ability of fine-tuning to recover this knowledge effectively (Gu et al., 2023).

To overcome these limitations, we investigate the potential of teacher distillation, a technique that transfers knowledge from a large teacher model to a smaller student model. Currently, there is limited exploration in distilling LLMs under low-resource conditions. While the MiniLM initiative has made progress in distilling large models, this approach requires fine-tuning of the student model before distillation, leading to prolonged training times (Gu et al., 2023). In this workflow, it becomes crucial to rapidly and effectively assimilate knowledge from a limited number of iterations. However, the significant disparities across different models pose a formidable challenge in devising strategies that ef-

fectively mitigate models' differences to enhance the efficacy and efficiency of the distillation process.

This paper presents Sparse KD, a novel distillation framework specifically designed for sparse models in constrained fine-tuning scenarios. We observe from the toy experiment in Fig 1 that even slight modifications in the distillation temperature can have an impact on the outcomes, albeit modestly. Based on this insight, we introduce an innovative adaptive temperature mechanism that dynamically adjusts throughout the training phase. Notably, our approach incorporates a dual-temperature strategy, using separate temperatures for the teacher and student models. This customization accounts for the unique differences between models, a feature notably absent in conventional methods. Furthermore, we leverage knowledge alignment module by max-min normalization or standardization to enhance the distillation of intermediate layer features more effectively. During our search for the optimal KD loss, we employ a Bayesian optimization (Snoek et al., 2012), employing expected improvement as our acquisition function. This technique, often overlooked in traditional approaches, allows us to efficiently identify parameters that align with optimal performance. Importantly, this paper explores the impact of our customized distillation technique on enhancing the generalization performance of LLMs across various linguistic tasks.

To assess the effectiveness and stability of our approach, we conduct comprehensive experiments across a wide range of task types, including machine translation, sentiment analysis, and question answering. The results substantiate the efficacy of our combined approach, showcasing the potential for efficient and stable deployment of LLMs. We further perform zero-shot experiments using a suite of eleven datasets, including those from the GLUE (Wang et al., 2018) and SuperGLUE (Wang et al., 2019) benchmarks, covering various tasks from different sub-domains. The experimental results demonstrate that our proposed method outperforms leading baseline techniques in terms of effectiveness and superiority. Additionally, we conduct ablation studies to determine which strategies yield the most favorable results.

2 Related Work

Knowledge Distillation (Hinton et al., 2015) aims to transfer knowledge from a large model (teacher model) to a smaller model (student model), enabling the latter to mimic the behavior of the former. KD can be categorized into three main types based on the nature of the knowledge transferred: Response-Based KD, which focuses on the model's outputs as introduced by Hinton and colleagues in their pioneering work; Feature-Based KD (Romero et al., 2014) distils features from intermediate layers can effectively enhance the student model's performance; and Relation-Based KD aims to transfer relational knowledge between layers within the model, as proposed by Tung and Mori (Tung and Mori, 2019). Distillation techniques include Offline Distillation, where the teacher model fully trains before guiding the student model; Online Distillation (Zhang et al., 2017), enables the simultaneous training of both teacher and student models, offering potential advantages over traditional offline methods; and Self-Distillation (Furlanello et al., 2018), where a model improves itself using its outputs.

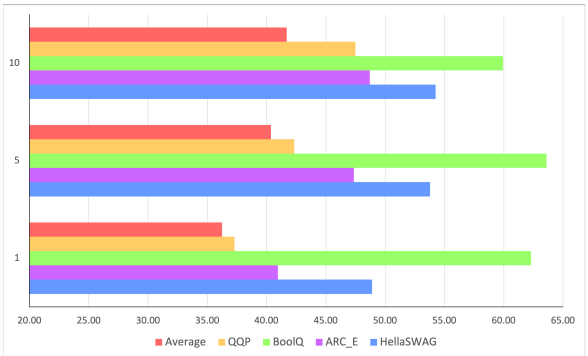

Figure 1: Toy experiments in TinyLlama1.1b with varying temperature settings.

Current research has shown the effectiveness of applying KD to pruned models for performance enhancement. For instance, Sanh demonstrated how KD could create a smaller, faster BERT model (Sanh et al., 2019). Although DistilBERT was not directly applied to pruned models, the study showcased the potential of KD in optimizing LLMs. Furthermore, Sanh introduced a dynamic pruning technique and enhanced model performance through fine-tuning, providing insights for performance recovery post-pruning (Sanh et al., 2020). Despite these and other related works offering valuable insights and methodologies for combining pruning techniques with KD, literature specifically addressing KD strategies applied to pruned LLMs remains scarce. This indicates that while significant progress has been made in optimizing LLMs

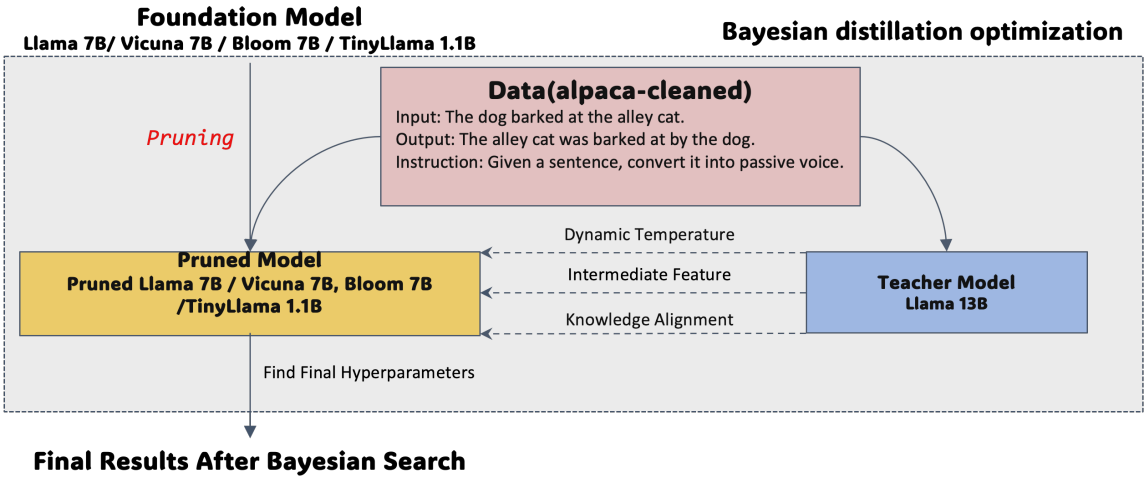

Figure 2: Overview of the framework, which includes distillation strategies, Bayesian Seach and also prunning procedure.

through KD and model pruning, finer adjustments of the distillation process to suit the specific needs of pruned models. Specifically, employing advanced strategies like dynamic temperature adjustments to further improve the performance and efficiency of pruned models, represents an open and underexplored research area.

3 Methodology

We started by pruning the model to make it simpler and more efficient (Ma et al., 2023a). Initially, we tried LoRA-training but found it to be ineffective. Therefore, we explored model distillation for models after LoRA training. At first, we used Kullback-Leibler (KL) divergence as the main loss function for distillation. We observed that the temperature significantly but subtly impacted distillation effectiveness. To address the limitations of fixed temperature settings, we proposed a new approach: dynamic temperature settings tailored to the teacher and student models. This adjustment helps the student model distil knowledge more comprehensively and flexibly. We also improved the hyperparameter optimization using a modified Bayesian optimization and implemented a different knowledge alignment strategy. These enhancements aim to make the distillation process more efficient and adaptable. Fig 2 provides an overview of the framework.

3.1 Overall Distillation Optimization

Commencing our experiments, we adopted the distillation technique using KL divergence, as delineated by Hinton (Hinton et al., 2015) and encapsulated in Equation 1. Our primary focus was on optimizing the KL loss function.

In Equation 1, u signifies the input, ordinarily a question from the dataset, whereas v symbolizes the text generated by the model in response to u. The variable p represents the teacher model's conditional probability distribution, while q reflects that of the student model. The expectation function $\mathbb{E}_u$ is tasked with computing across the spectrum of possible inputs u and their respective outputs v. $p(v|u)$, and $q(v|u)$ are correspondingly rendered as softmax $\left(\frac{T}{t'}\right)$ and softmax $\left(\frac{S}{t'}\right)$, where S and T denote logits. t' is temperature.

In the conducted toy experiment in Fig 1, it was observed that performance on most test sets improved with temperature increasing, but some, like BoolQ, showed an initial improvement followed by a decline. This indicates that the discrepancy between teacher and student models can disrupt knowledge distillation at extreme temperatures(high or low temperature). Only optimal temperature ensures effective knowledge transfer from teacher to student models.

To optimize the student model's learning, we revised the loss function $\mathcal{L}(\phi)$, incorporating dynamic temperature adjustments for distillation. The batch size and token length per sample are denoted by N and M, respectively. We introduce $\sigma_{t,i}$ and $\sigma_{s,i}$ as the standard deviations for the teacher and student models at the i^{th} sample in Equation 4, to adjust the logits $P_{t,i,j}$ and $Q_{s,i,j}$. This allows for a nuanced knowledge transfer by softening the logits with temperature-sensitive softmax functions in Equation 3. In the second component, L denotes

selectively matched layers from the teacher to the student model, each modulated by a unique scaling factor β_l. Here, S and T respectively signify the hidden states at layer l for the student and teacher models, in the i^{th} token of the m^{th} batch.

This dynamic temperature method softens the probability distributions, making them smoother. This "softening" spreads probabilities across a wider range of tokens, encouraging more nuanced learning from the teacher model. This approach not only makes the distillation process more effective but also ensures a deeper and more comprehensive transfer of knowledge to the student model.

$$\begin{aligned}\phi &= \arg\min_{\phi} \mathcal{L}(\phi) = \arg\min_{\phi} D_{KL}\left[p \| q_\phi\right] \\ &= \arg\min_{\phi} \mathbb{E}_u \left[\log\left(\frac{p(v \mid u)}{q_\phi(v \mid u)}\right)\right]\end{aligned} \tag{1}$$

$$\begin{aligned}\mathcal{L}(\phi) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{M} \alpha \cdot KL\left[P_{t,i,j} \| Q_{s,i,j}\right] + \\ \sum_{l=1}^{L} \frac{\beta_l}{N \times M} \sum_{i=1}^{N}\sum_{j=1}^{M} \|S_{l,i,j} - T_{l,i,j}\|_2^2\end{aligned} \tag{2}$$

3.2 Dynamic Temperature Strategy

To improve knowledge transfer between teacher and student models, our research proposes dynamic temperature, moving beyond the limitations of static distillation temperatures. This approach not only optimizes knowledge transfer by adjusting the focus during distillation but also softens the teacher model's logits to reveal complex token relationships, as delineated in Equation 2.

The dual-temperature mechanism provides distinct softening levels for teacher and student models, enabling precise control over the quality and quantity of information transferred. This method allows the student model to filter out irrelevant noise and focus on crucial structural insights, reducing the risk of overfitting while maintaining important characteristics. Consequently, this approach markedly improves the student model's ability to generalize on new tasks, effectively utilizing structural knowledge from the teacher model to excel in zero-shot situations.

$$\begin{aligned}P_{t,i,j} &= \text{softmax}\left(\frac{L_t[i,j]}{\sigma_{t,i}}\right) \\ Q_{s,i,j} &= \text{softmax}\left(\frac{L_s[i,j]}{\sigma_{s,i}}\right)\end{aligned} \tag{3}$$

$$\begin{aligned}\sigma_{t,i} &= \sqrt{\frac{1}{M}\sum_{j=1}^{M}(L_t[i,j] - \mu_{t,i})^2} \\ \sigma_{s,i} &= \sqrt{\frac{1}{M}\sum_{j=1}^{M}(L_s[i,j] - \mu_{s,i})^2}\end{aligned} \tag{4}$$

3.3 Knowledge Alignment Strategy

During distillation, we align the varying logit dimensions from student and teacher models to a uniform size, employing a fixed-dimension method for consistency, as illustrated in Equation 2. This process ensures both models' logits match dimension M. To avoid gradient explosion before feeding logits into Equation 2, we preprocess them using Min-Max Normalization (Equation 5) and Standardization (Z-score Normalization) (Equation 6), with a preference for Standardization in our tests. This normalization technique is consistently applied across intermediate layer distillation as well.

$$L_i' = \frac{x_i - \min(L)}{\max(L) - \min(L) + \epsilon} \tag{5}$$

$$L_i = \frac{x_i - \mu}{\sigma + \epsilon} \tag{6}$$

3.4 Bayesian Distillation Optimization

Beyond the outlined techniques, we utilized Bayesian optimization during the distillation phase to streamline the hyperparameter selection process and boost the model performance.

Given the dataset (X, Y), where X represents hyperparameters and Y represents corresponding evaluation results, a Gaussian Process (GP), $f \sim GP(\mu, K)$, is employed as a surrogate model to approximate the predictive function. Defined by a mean function, which is typically assumed to be zero initially but updates after training in (X, Y), and a covariance function encapsulated by a kernel, the GP captures the correlation between points, providing a foundation for model predictions.

The acquisition function, Expected Improvement (EI) in Equation 7, is calculated based on the GP's predictions, aiming to identify the x_{next} from a set of potential hyperparameters x^*, described in Table 1, that offers the maximum expected improvement over the current best observation. $\mu(x^*)$ and $\sigma(x^*)$ denote the predicted mean and variance at x^*. $f(x^+)$ is the best observed function value in (X, Y). ξ is a small positive parameter to balance exploration and exploitation, and Φ and ϕ are the

Hyperparameter	Values
Hard Label Weight	$0.1, 1, 10, 20$
Soft Label Weight	$1e-8, 1e-7, 1e-3, 1, 10, 20$
Temperature	$1, 5, 8, 16, 20, 30$
Logits Normalization	"none", "Max-Min", "standardize"
Hidden States Normalization	"none", "Max-Min", "standardize"
Intermediate Layer Config[1]	None, Last_Layer, Last_Middle Layer, Last_Middle_Start Layer
KD Type	KLD, Dynamic Temperature

Table 1: Overview of hyperparameters search space. KLD is Kullback–Leibler divergence.

Category	Dataset
Question Answering	OpenBookQA (OPQA) (Mihaylov et al., 2018)
	ARC Easy (ARC_E) (Clark et al., 2018)
	ARC Challenge (ARC_C)(Clark et al., 2018)
	BoolQ (Clark et al., 2019)
	QNLI (Wang et al., 2018)
Textual Entailment	QQP (Chen et al., 2017)
Text Classification	SST-2 (Socher et al., 2013)
Language Modeling Dataset	WikiText-2 (Merity et al., 2016)
Commonsense Reasoning	PIQA (Bisk et al., 2020)
	HellaSWAG (Zellers et al., 2019)
	Winogrande (Sakaguchi et al., 2021)

Table 2: Overview of datasets categorized by task.

cumulative distribution function and probability density function of the standard normal distribution, respectively. Z is a standardized value calculated from the predicted mean, variance, and the current best value in Equation 8.

By optimizing EI, Bayesian optimization determines the x_{next} of evaluation, updates the GP model with $(x_{\text{next}}, f(x_{\text{next}}))$, and repeats this iterative process until a predefined stopping criterion is met, such as a maximum number of iterations or a threshold of improvement in the target function.

In summary, Bayesian optimization efficiently identifies the optimum of a function with limited evaluations through precise estimation using a Gaussian Process model and guiding the search process with the Expected Improvement acquisition function. This method is advantageous for its global search capability and efficiency under limited evaluations.

$$EI(x^*) = \left(\mu(x^*) - f\left(x^+\right) - \xi\right)\Phi(Z) + \sigma(x^*)\phi(Z) \tag{7}$$

$$Z = \frac{\mu(x^*) - f\left(x^+\right) - \xi}{\sigma(x)} \tag{8}$$

4 Experimental Setup

4.1 Datasets

We extracted 13,000 training and 2,000 validation samples from the cleaned Alpaca dataset[2] for LoRA fine-tuning and distillation. For evaluation, we selected 11 datasets across various NLP domains to assess model performance comprehensively on a zero-shot basis, testing generalization across diverse tasks. The datasets are detailed in Table 2, showcasing the range of NLP areas examined.

4.2 Baseline Methods

We established a comprehensive experimental set by selecting four advanced LLMs as student baselines: Bloom-7b1 (Workshop et al., 2022), Llama-7b-hf (Touvron et al., 2023a), Vicuna-7b-v1.1 (Zheng et al., 2023), and TinyLlama-1.1b (Zhang et al., 2024), which will be distilled with a single teacher model baseline, Llama-13b (Touvron et al., 2023a). Our approach was strategically designed with an emphasis on efficiency, focusing on enhancing the performance of pruned LLMs. Consequently, our experimental design included a variety of configurations involving the teacher model, student models, and student models' pruned counterparts, with and without the application of LoRA for fine-tuning.

4.3 Ablation Details

In this research, we employed the TinyLlama-1.1b model in conjunction with Sparse KD across a series of ablation studies to rigorously evaluate the relative efficacy of our innovative strategies: dynamic temperature, and intermediate layer feature distillation. The knowledge alignment cannot be removed because the logits between the teacher and student model must be aligned. These strategies formed the core of our experimental investigation, carefully orchestrated to assess each method's contribution. By methodically altering the elements incorporated into the evaluation, our goal was to delineate and ascertain the distinct impacts of our fusion methodology on TinyLlama-1.1b, as well as the individual contributions of dynamic temperature and feature distillation across layers, on the

[1]The "Intermediate Layer Config" options allow for custom specification or removal of layers from teacher models to student models, where "Last", "Middle", and "Start" do not denote the number of layers but rather the regions within the architecture where specified layers are located. Each option offers a flexible way to define the involvement of intermediate layers in the model.

[2]https://huggingface.co/datasets/yahma/alpaca-cleaned

cumulative performance.

4.4 Pruning and Distillation

The pruning strategy was standardized at a 25% ratio, effectively retaining approximately 75% of the original parameters. Due to the nature of block-based pruning, the exact retention of parameters cannot be precisely 75% (Ma et al., 2023b). The experimental results, including the count of trainable parameters post-pruning and distillation results, have been meticulously documented in Table 4 and Table 3.

During distillation, we scaled intermediate layer losses to prevent gradient explosion from large loss values, ensuring training stability. We also added a specific hard label loss proportion, optimized via Bayesian optimization to balance it with soft label loss. Soft labels were scaled down by $1e{-}7$, and hard label loss was increased tenfold, equalizing their scales to prevent gradient issues.

The pruned models underwent distillation employing Sparse KD: dynamic temperature with intermediate layer feature distillation, knowledge alignment, and Bayesian distillation optimization strategies.

This comprehensive approach, meticulously designed, aims to optimize distillation efficiency between student and teacher models, enhancing the refined capabilities of pruned models through targeted strategies. It integrates pruning strategies and Sparse KD to improve overall distillation outcomes.

4.5 Evaluation

In our study, we designed four distinct experimental groups to examine the impact of post-distillation pruning on the performance of LLMs, including an unpruned baseline (comprising both student and teacher models), and pruned student models, with and without distillation. We further distinguished the outcomes within distillation scenarios using Bayesian optimization and random search methods. Ablation studies were conducted utilizing the TinyLlama-1.1b model, evaluated through a language model evaluation toolkit (Sutawika et al., 2023) in Table 2. The focus of our evaluations was on accuracy, and, for certain cases like Wikitext-2, on Perplexity (PPL), where higher accuracy and lower PPL denote performance improvement.

[3] https://github.com/EleutherAI/lm-evaluation-harness

5 Results and Analysis

5.1 Results

The main results are presented in Table 3. It is evident from the table that our method (Distillation with Bayesian Search) surpasses all baseline methods across most tasks. Results highlighted in bold indicate the best performance within each experimental group for models post-pruning, across Sparse Model (without LoRA and with LoRA), Stand KD, Sparse KD (Random Search), and Sparse KD (Bayes Search).

Through the application of Sparse KD, the models TinyLlama-1.1B, Llama7B, Vicuna-7Bv1.1, and Bloom7b exhibited average improvements of 2.58, 1.90, 2.78, and 2.97 respectively across 11 datasets, surpassing the performance of Sparse Models (with LoRA). Compared to Stand KD, these improvements further stand at 1.07, 4.40, 7.30, and 2.84 respectively for each model, which indicate that Sparse KD method is better than Stand KD but it is more suitable for larger models. Notably, in experiments with TinyLlama-1.1B and Vicuna-7Bv1.1, models distilled using Sparse KD achieved results very close to Dense Model. We also observed an interesting phenomenon on the QNLI test set, where Sparse Model (Without LoRA) actually outperformed the original model in Llama7b. Some Sparse Models experienced a decline in performance in Stand KD compared with Sparse Model (wo LoRA), such as in the Llama7b, Vicuna-7B and Bloom-7B groups in QNLI and SST-2. This could be because a uniform temperature setting does not account for the differences between various student models.

We also observed that for relatively simpler tasks such as BoolQ, QNLI, QQP, and SST-2, some models that were pruned and then distilled exceeded the performance of Dense Model. These four tasks, predominantly in the question-answering category, can also be classified as classification tasks. BoolQ involves answering true or false to questions based on a passage, QNLI entails determining whether text1 implies text2, QQP assesses whether two questions are duplicates, and SST-2 is about binary sentiment classification. Thus, distillation following model compression is particularly effective for classification tasks. Although the results on Wikitext-2 were not as favorable, they were not significantly different from those achieved through LoRA fine-tuning. Our Sparse KD method in pruned models, which are Distillation(Bayes

Model	Method	OPQA	Hella-SWAG	Wino-grande	PIQA	ARC_E	BoolQ	ARC_C	QNLI	QQP	SST-2	Aver-age	Wiki-Text2
Llama 13B	Teacher Model	44.80	79.07	72.77	80.09	74.71	77.98	47.61	50.74	46.37	69.04	64.32	11.58
	Teacher Model(lora)	46.40	80.68	72.85	80.69	77.10	80.89	50.68	50.05	48.40	49.08	63.68	11.76
Tiny-Llama 1.1B	Dense Model	36.00	59.20	59.12	73.29	55.35	57.83	30.12	48.49	54.28	69.61	54.33	16.53
	Sparse Model (wo LoRA)	32.00	50.82	55.72	69.42	46.84	53.67	27.47	50.52	50.33	65.02	50.18	30.29
	Sparse Model (w LoRA)	33.00	52.13	57.46	**71.22**	48.11	53.70	29.69	50.49	47.07	73.05	51.59	27.59
	Stand KD	**34.40**	54.07	57.46	71.21	49.24	54.92	29.18	53.18	55.73	71.56	53.10	22.15
	Sparse KD(Random Search)	34.12	53.29	57.93	69.15	46.59	**63.27**	29.69	49.46	38.06	50.92	49.25	25.43
	Sparse KD(Bayes Search)	34.00	**54.34**	**58.25**	70.78	**49.45**	54.19	**29.78**	**52.61**	**57.36**	**80.96**	**54.17**	**22.00**
Lla-ma7b	Dense Model	44.40	76.21	69.85	79.16	72.81	75.11	44.71	51.16	48.00	76.38	63.78	12.62
	Sparse Model (wo LoRA)	36.20	62.45	59.43	73.18	51.14	58.69	32.17	**53.18**	46.14	74.54	54.71	22.54
	Sparse Model (w LoRA)	39.40	67.18	62.12	74.65	59.34	57.37	36.86	51.53	48.95	**84.40**	58.18	19.58
	Stand KD	38.60	67.34	62.67	73.88	59.81	62.35	37.97	51.60	46.50	56.08	55.68	20.56
	Sparse KD(Random Search)	39.20	64.17	63.47	73.23	52.50	67.74	33.45	51.39	**57.40**	49.29	55.18	27.29
	Sparse KD(Bayes Search)	**44.60**	**72.01**	**65.36**	**78.10**	**65.81**	**67.80**	**39.59**	52.08	55.36	60.09	**60.08**	**19.05**
Vinc-una7b	Dense Model	43.40	74.64	70.09	78.56	72.01	78.32	43.77	50.60	60.70	54.24	62.63	16.10
	Sparse Model (wo LoRA)	34.20	60.18	59.04	72.04	54.46	49.82	33.02	51.38	58.85	72.02	54.50	28.83
	Sparse Model (w LoRA)	39.00	65.72	63.77	73.40	59.98	53.00	36.26	50.54	56.44	76.95	57.51	**20.60**
	Stand KD	33.20	54.66	58.64	71.05	49.83	59.41	30.29	58.28	55.11	59.40	52.99	22.73
	Sparse KD(Random Search)	40.20	68.33	65.51	75.30	61.75	**62.22**	36.44	54.70	56.57	70.30	59.13	23.35
	Sparse KD(Bayes Search)	**41.20**	**69.59**	**65.98**	**76.39**	**61.79**	56.15	**37.96**	**56.40**	**60.22**	**77.18**	**60.29**	20.93
Bloom 7b	Dense Model	35.80	62.26	64.40	73.56	57.28	62.91	33.45	51.18	41.87	49.08	53.18	26.58
	Sparse Model (wo LoRA)	31.60	38.13	56.35	67.79	46.84	61.99	26.71	49.33	38.13	61.01	47.79	**75.51**
	Sparse Model (w LoRA)	31.20	33.95	**57.22**	65.78	44.49	46.30	25.85	46.29	**49.78**	51.95	45.28	190.57
	Stand KD	29.00	35.01	55.95	65.28	45.41	60.86	25.51	49.50	36.80	50.80	45.41	149.58
	Sparse KD(Random Search)	31.60	38.12	56.35	67.79	46.84	**61.98**	26.71	49.33	38.13	61.01	47.99	135.66
	Sparse KD(Bayes Search)	**32.20**	**39.36**	55.09	**68.55**	**46.89**	61.74	**27.65**	**50.25**	39.39	**61.35**	**48.25**	124.49

Table 3: The main results from our multi-task testing, with the exception of Wikitext-2, were derived from the Language Model Evaluation Harness[3]. For Wikitext-2, the Perplexity (PPL) metric was employed, whereas accuracy served as the metric for all other tasks. **Results** highlighted in bold indicate the best performance within each experimental group for models post-pruning. This methodological approach ensures a rigorous and comprehensive evaluation of our models' effectiveness across a diverse array of tasks, adhering to the high standards of academic rigor and professionalism expected at scholarly conferences.

Type	**Model**	**Base Parameters (B)**	**Pruned Parameters (B)**	**Final Pruned Ratio**
Teacher Model	Llama13B	13B	-	-
Student Model	Tiny-Llama1.1B	1.1B	0.961B	0.8735
	Bloom7b	7.069B	6.282B	0.8887
	Vincuna7b	6.738B	5.423B	0.8048
	Llama7b	6.738B	5.423B	0.8048

Table 4: verview of model parameter adjustments. All parameter values are expressed in billions (B).

Search) or Distillation(Random Search) in the results Table, outperformed most Sparse Models (w LoRA) in question-answering tasks (such as OPQA, ARC_E, ARC_C) and inference tasks (like HellaSwag, Winogrande, and PIQA). Moreover, models optimized through Bayesian search generally exhibited superior performance compared to those subjected to random search, as depicted in Table 3. The Baysian optimization also is depicted in Fig 3 to illustrate that most outcomes achieved through Bayesian search are notably positive.

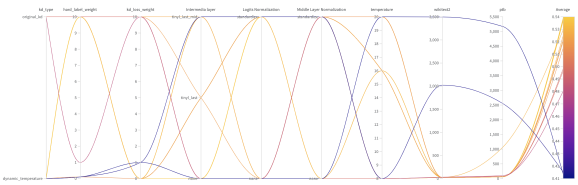

Figure 3: Visualization of search results during Bayesian distillation optimization.

5.2 Ablation Study

In this stage, we conducted an ablation study to evaluate the impact of different strategies, Dynamic Temperature, Intermediate Layer Distillation and Kullback–Leibler divergence on the performance

Method	OPQA	Hella-SWAG	Wino-grande	PIQA	ARC_E	BoolQ	ARC_C	QNLI	QQP	SST-2	Average	Wiki-text2
Baseline	32.00	50.82	55.72	69.42	46.84	53.67	27.47	50.52	50.33	65.02	50.18	30.29
+KD	33.20	53.52	58.09	69.53	46.55	62.2	29.18	49.28	45.07	69.27	51.59	25.851
+ KD + ID	34.40	54.07	57.46	71.22	49.24	54.92	29.18	53.18	55.73	71.56	53.10	22.15
+ KD + ID + DT	34.00	54.34	58.25	70.78	49.45	54.19	29.78	52.61	57.36	80.963	54.17	22.00

Table 5: Ablation study results in 0.25 pruning ratio in TinyLlama1.1b on various NLP tasks. KD is KL divergence, ID is Intermediate Layer Distillation and DT is Dynamic Temperature.

of the TinyLlama1.1B model. Each strategy was aimed at enhancing the model's distillation process through different mechanisms, thereby improving its performance across a variety of tasks. To accurately assess the contribution of each strategy, we began with a complete model encompassing all strategies and systematically removed each one, recording the resultant changes in performance metrics.

Experimental results in Table 5 show that removing dynamic temperature led to a 1.076% average accuracy decrease across 11 tasks, highlighting its crucial role in optimizing model performance. Eliminating intermediate layer distillation resulted in a performance drop 1.51%, indicating its significant impact on performance improvement.. Moreover, performance declines occurred with the removal of Kullback–Leibler divergence, with accuracy decreases of 1.41%, underscoring the influence of KL divergence distillation (Stand KD) to Sparse Model (wo LoRA).

Furthermore, comparative experiments between random search and Bayesian Search within the main results in Table 3 have already substantiated the superior overall performance of Bayesian search, rendering separate ablation studies for Bayesian search superfluous.

These findings confirm the contribution of a custom-designed distillation loss function with dual dynamic temperature coefficients, final and intermediate layer feature distillation, and the Bayesian optimization to the model's overall performance and reveal their relative importance during the model training process.

6 Conclusion

In this paper, we introduce three novel strategies for knowledge distillation in pruned models. The first strategy involves a custom-designed distillation loss function that incorporates dual temperature coefficients to precisely control the quantity and quality of information transferred. This approach ensures that the student model is not overwhelmed by noise from the teacher model and retains crucial structural knowledge. The second strategy focuses on feature-based distillation from both the final output and intermediate layers with knowledge alignment, further enhancing the knowledge transfer process. Lastly, we introduce a method for optimizing the knowledge distillation loss through Bayesian optimization, enabling the identification of optimal parameters. Extensive experiments are conducted on multiple large models and various NLP tasks, including ablation studies. The results demonstrate the effectiveness and stability of the proposed framework, highlighting its potential for efficient knowledge distillation in pruned models.

7 Limitations

While this work enhances the generalized capabilities of pruned models, it does not specifically improve capabilities in categories such as inference and logical analysis, language generation, natural language understanding, knowledge retrieval, and integration. These areas present opportunities for detailed exploration in future research.

8 Ethics Statement

Our approach solely concentrates on the technical aspects of efficiently deploying LLMs. It does not involve any ethical or social implications.

References

Yonatan Bisk, Rowan Zellers, Ronan Le bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7432–7439.

T Brown, B Mann, N Ryder, M Subbiah, JD Kaplan, P Dhariwal, A Neelakantan, P Shyam, G Sastry, A Askell, et al. Language models are few-shot learners in advances in neural information processing systems (eds larochelle, h., ranzato, m., hadsell, r., balcan, mf & lin, h.) 33 (curran associates, inc., 2020), 1877–1901. *arXiv preprint arXiv:2005.14165*.

Zihang Chen, Hongbo Zhang, Xiaoji Zhang, and Leqi Zhao. 2017. Quora question pairs.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457.

Elias Frantar and Dan Alistarh. 2023. Massive language models can be accurately pruned in one-shot. *arXiv preprint arXiv:2301.00774*.

Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. 2018. Born again neural networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1607–1616. PMLR.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. Knowledge distillation of large language models. *arXiv preprint arXiv:2306.08543*.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531.

Woosuk Kwon, Sehoon Kim, Michael W Mahoney, Joseph Hassoun, Kurt Keutzer, and Amir Gholami. 2022. A fast post-training pruning framework for transformers. *Advances in Neural Information Processing Systems*, 35:24101–24116.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023a. Llm-pruner: On the structural pruning of large language models. *arXiv preprint arXiv:2305.11627*.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023b. LLM-pruner: On the structural pruning of large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *ArXiv*, abs/1609.07843.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. 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*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2014. Fitnets: Hints for thin deep nets. *CoRR*, abs/1412.6550.

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

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Victor Sanh, Thomas Wolf, and Alexander Rush. 2020. Movement pruning: Adaptive sparsity by fine-tuning. In *Advances in Neural Information Processing Systems*, volume 33, pages 20378–20389. Curran Associates, Inc.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. 2012. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

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

Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. 2023. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*.

Lintang Sutawika, Leo Gao, Hailey Schoelkopf, Stella Biderman, Jonathan Tow, Baber Abbasi, ben fattori, Charles Lovering, farzanehnakhaee70, Jason Phang, Anish Thite, Fazz, Aflah, Niklas Muennighoff, Thomas Wang, sdtblck, nopperl, gakada, tttyuntian, researcher2, Chris, Julen Etxaniz, Zdeněk Kasner, Khalid, Jeffrey Hsu, AndyZwei, Pawan Sasanka Ammanamanchi, Dirk Groeneveld, Ethan Smith, and Eric Tang. 2023. Eleutherai/lm-evaluation-harness: Major refactor.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023b. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023c. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Fred Tung and Greg Mori. 2019. Similarity-preserving knowledge distillation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1365–1374.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. *SuperGLUE: a stickier benchmark for general-purpose language understanding systems*. Curran Associates Inc., Red Hook, NY, USA.

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

BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Ying Zhang, Tao Xiang, Timothy M. Hospedales, and Huchuan Lu. 2017. Deep mutual learning. *CoRR*, abs/1706.00384.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.

A Experimental Details

All the experiments are run on 8 NVIDIA A100GPUs. The experiments runing time is 55 minutes in 7B models and 30 minutes in 1.1B model. The ablation experiments setting is $1e-7$ in hard label weight, 10 in soft label weight.