BayesKD: Bayesian Knowledge Distillation for Compact LLMs in Constrained Fine-tuning Scenarios

Wei Li[1]* **Lujun Li**[2]* **Mark Lee**[1] **Shengjie Sun**[3] **Lei Zhang**[4] **Wei Xue**[2] **Yike Guo**[2]

[1]University of Birmingham [2]HKUST [3]AISpeech Co., Ltd. [4]University of Exeter

WXL885@student.bham.ac.uk, lilujunai@gmail.com, M.G.Lee@bham.ac.uk

shengjie.sun@aispeech.com, l.zhang6@exeter.ac.uk, weixue@ust.hk, yikeguo@ust.hk

Abstract

Large language models (LLMs) have revolutionized various domains with their remarkable capabilities, but their massive parameter sizes pose significant challenges for fine-tuning and inference, especially in resource-constrained environments. Conventional compression methods often result in substantial performance degradation within LLMs and struggle to restore model quality during fine-tuning. To address this challenge, we present Bayesian Knowledge Distillation (BayesKD), a novel distillation framework meticulously designed for compact LLMs in resource-constrained fine-tuning scenarios. Departing from conventional LLM distillation methods that introduce time-consuming paradigms and fail to generalize in compressed LLM fine-tuning scenarios, our BayesKD develops the Logits Dual-Scaling, Knowledge Alignment Module, and Bayesian Distillation Optimization. In particular, our Logits Dual-Scaling strategy adaptively aligns the strength of the teacher's knowledge transfer, while the Knowledge Alignment Module bridges the gap between the teacher and student models by projecting their knowledge representations into a shared interval. Additionally, we employ Logits-Aware Bayesian Optimization to swiftly identify optimal settings based on these strategies, thereby enhancing model performance. Extensive experiments across diverse tasks demonstrate that BayesKD consistently outperforms baseline methods on various state-of-the-art LLMs, including LLaMA, Qwen2, Bloom, and Vicuna. Notably, our BayesKD achieves average accuracy gains of 2.99% and 4.05% over standard KD for the 8B parameter LLaMA and Qwen2 model. Codes are available in the supplementary materials.

1 Introduction

The rapid advancements in Large Language Models (LLMs) have revolutionized natural language processing (Wei et al., 2022b,a), yet the substantial size and computational demands of these models pose significant challenges for practical deployment (Zhang et al., 2022). To address these challenges, various LLM compression techniques (Zhang et al., 2023b), such as pruning (Sun et al., 2024a), have emerged to develop streamlined LLMs, thereby boosting inference efficiency. Nonetheless, these compressed models often experience notable performance degradation and struggle to regain their original quality through additional fine-tuning (Ma et al., 2023).

Table 1: Hyperparameter sensitivity experiments: Performance generally improves with higher scaling value in TinyLLaMA1.1B, but tasks like BoolQ and OpenbookQA show a decline, indicating potential hindrances in knowledge transfer.

Scaling Values	BoolQ	ARC_C	OPQA	PIQA	SST-2
8	60.08	26.56	22.55	69.81	58.55
16	56.27	26.61	22.11	70.82	75.92
20	54.31	26.54	21.95	70.95	80.56

Problem Statement: A promising approach to mitigating such performance gaps is to leverage the original dense model as a teacher for Knowledge Distillation (KD). However, current LLM distillation methods (Gu et al., 2023) encounter two key limitations: *(1)* They impose substantial computational overhead due to complex distillation paradigms, and *(2)* they are primarily designed for normal LLMs rather than compressed ones. For example, while MiniLLM (Gu et al., 2023) improves the performance of the original LLaMA 7B by 5%, its gain diminishes to only 1% on the sparse LLaMA 7B (Table 9). These failures are likely attributable to the immense gap between compressed LLMs and their teacher counterparts, which poses difficulties for general distillation methods.

While recent KD approaches have explored various angles. Teacher Assistant (TA)-based

*Equal contribution. Correspondence to: Mark Lee and Yike Guo.

Findings of the Association for Computational Linguistics: ACL 2025, pages 138–152

methods (Mirzadeh et al., 2020; Zhang et al., 2023a) improve distillation performance by inserting intermediate-capacity models between the teacher and student. However, identifying and deploying a TA model with an optimal capacity ratio (e.g., 1.5 times the student's size) is computationally prohibitive at LLM scales, where even minor capacity adjustments lead to massive parameter counts and increased training costs. Similarly, logit standardization (Sun et al., 2024b), multi-level logit distillation (Jin et al., 2023), and intermediate-layer alignment methods (e.g., Universal-KD (Wu et al., 2021)) have delivered modest gains in smaller or less complex models but fail to yield substantial improvements for LLMs and compressed LLMs. For instance, applying these techniques (logit standardization, multi-level logit distillation, and Universal-KD) to Tiny-LLaMA1.1B and LLaMA-13B improves accuracy to only 52.33%, 51.89%, and 52.53%, respectively, across ten tasks (same tasks with Table 4 except Wiki). Such outcomes highlight that once models undergo compression or is LLM, established KD methods become less robust and struggle to restore substantial performance. Another issue arises from the sensitivity of KD to hyperparameters such as scaling values or temperature factors. Although adjusting logits can enhance certain tasks, it can also degrade others. Table 1 shows this challenge: increasing the scaling value from 8 to 20 boosts SST-2 accuracy from 58.55% to 80.56% but degrades results on BoolQ and OpenBookQA. These findings underscore that a one-size-fits-all scaling strategy is inadequate, and a more adaptive, task-aware tuning mechanism is required.

These observations raise two critical questions: ***Why do compact LLMs struggle to learn knowledge from their teacher models, and can a more specialized distillation paradigm address these issues effectively for such compressed model fine-tuning scenarios.?***

Our New Observations: We identify two factors that significantly influence KD performance in compressed LLMs: **(1) Massive Logit Gap.** The distributional discrepancy in logits between sparse and dense models is larger than that between dense models of different sizes. This gap intensifies with increasing model scale, contributing to the limited efficacy of current KD methods (Table 4). **(2) Hyperparameter Sensitivity.** As shown in Table 1, the optimal scaling value for knowledge distillation varies across different tasks. This issues significantly affect the performance of conventional distillation methods.

Our New Search Framework: Building upon these observations, we propose **Bayesian Knowledge Distillation (BayesKD)**, a novel framework designed for LLMs and their sparse models under resource-constrained fine-tuning conditions. BayesKD integrates three core components: **First**, a *logits dual-scaling* technique dynamically adjusts teacher and student logits based on their standard deviations, thereby narrowing distributional gaps and enabling more effective, task-aware scaling. **Second**, a *knowledge alignment* module employs min-max normalization to better align the intermediate-layer representations of teacher and student models, enhancing the transferability of critical features. **Third**, to tackle the hyperparameter-sensitive issue. we first drew the paradigm of Bayes search in the LLM-KD field and propose Logits-Aware Bayesian optimization (SABO). Our SABO first builds a search space with Logits Dual-Scaling, Knowledge Alignment Module as the core and different key hyper-parameters distillation position, loss weight, temperature factor as options. To improve the search efficiency, we employ an advanced Bayesian search with faster convergence than random search. To optimize the search cost, we select only 5% of the sub-dataset for searching, which speeds up the search by 20 times compared to searching directly on the original dataset. Finally, we deeply analyze searched distillers and get some guidance: compressed LLM distillation always favors deep intermediate knowledge, logits dual-scaling, and smaller loss weights.

Evaluation and Results: We rigorously evaluate BayesKD on diverse tasks, including WikiText-2 (Merity et al., 2016), OpenBookQA (Mihaylov et al., 2018), HellaSwag (Zellers et al., 2019) and others, utilizing various teacher-student pairs such as LLaMA 13B for Tiny-LLaMA1.1B, LLaMA 7B, Vicuna 7B, and Bloom 7B; LLaMA-3 70B for LLaMA-3 8B; and Qwen-2 72B for Qwen-2 7B. Across these scenarios, BayesKD consistently outperforms baseline methods, delivering an average improvement of 4.4% over standard KD and a 2.6% advantage relative to vanilla LoRA fine-tuning baseline. Notably, the improvements are more pronounced for larger models, highlighting the scalability and robustness of the proposed framework.

The remainder of this paper is organized as follows: Section 2 reviews related work, Section 3 explains our methodology, Section 4 discusses results, and Section 5 concludes.

2 Related Work

Knowledge distillation (KD) (Hinton et al., 2015) methods can be broadly categorized into response-based methods focused on model output (Turc et al., 2020; Li et al., 2020; Jiao et al., 2020) and feature-based methods extracting intermediate layer features (Romero et al., 2015; Sun et al., 2019a; Tang et al., 2019). While these traditional approaches prove effective for BERT-scale models, they become inadequate for LLMs (Gu et al., 2023), where the vast capacity disparities present unprecedented challenges.

The capacity gap between teacher and student models remains a fundamental challenge, first systematically studied through Teacher Assistant Knowledge Distillation (TAKD) (Mirzadeh et al., 2020). Though TAKD effectively bridges smaller model gaps using intermediate networks, it becomes impractical for LLMs where the capacity gap between teacher and assistant often exceeds several orders of magnitude. This limitation significantly impacts the effectiveness of knowledge transfer in the LLM era.

Recent works have addressed these challenges with varying success. Pro-KD (Rezagholizadeh et al., 2022) introduces progressive distillation along the teacher's training trajectory but focuses mainly on BERT-scale models, leaving larger capacity gaps unexplored. Their method also requires multiple checkpoints, causing significant storage overhead for LLMs. (Zhang et al., 2023a) establishes a linear law for capacity gaps, but tests only models under 3B parameters inheriting TAKD's limitations in LLM distillation. Newer LLM-specific methods using reinforcement learning, which can better align the outputs of teacher and student LLMs, continue to face efficiency challenges (Ko et al.; Zhong et al., 2024).

Existing knowledge distillation methods face computational overhead and limited effectiveness for compressed LLMs. Capacity gaps challenge smaller models in replicating larger ones. Our BayesKD method bridges the logits distribution gap, enhancing student performance.

3 Methodology

Motivations and Overall Framework: Our BayesKD addresses a critical challenge in KD for LLMs and their sparse models: the pronounced logits distribution discrepancy between sparse and dense models. This discrepancy, which intensifies with model scale, significantly impairs traditional distillation methods. Our empirical analysis (Table. 4 and varying sparsity table in Appendix) reveals that single-temperature approaches, effective for dense model pairs, fail to adequately capture the complexity of compact LLMs. To overcome these limitations, we introduce a three-pronged approach. First, a Logits Dual-Scaling strategy bridges the teacher-student logits distribution gap while addressing the task-dependent scaling value sensitivity. Second, a Knowledge Alignment Module reconciles intermediate layer disparities. Inspired by prior works (Hou et al., 2020; Sun et al., 2019b), we focus on aligning the middle and final layers, which have been shown to capture task-specific knowledge effectively. Finally, a Logits-Aware Bayesian Distillation Optimization method ensures efficient hyperparameter tuning. This comprehensive framework specifically targets the unique challenges of compact LLMs distillation in resource-constrained environments.

3.1 Logits Dual-Scaling Strategy

The Logits Dual-Scaling strategy adjusts logits separately for the teacher and student models to address logits distribution discrepancies and task-dependent sensitivity in distillation. Scaling values are dynamically updated based on logits' standard deviations during the process, allowing optimal knowledge transfer between models. The loss function $\mathcal{L}_d$ is defined as:

$$\mathcal{L}_d = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{M} \gamma \cdot KL\left[P_{t,i,j} \| Q_{s,i,j}\right] \tag{1}$$

$$\begin{aligned} P_{t,i,j} &= \text{softmax}\left(\frac{T_{i,j}}{\sigma_{t,i}}\right), \\ Q_{s,i,j} &= \text{softmax}\left(\frac{S_{i,j}}{\sigma_{s,i}}\right), \end{aligned} \tag{2}$$

Where $P_{t,i,j}$ and $Q_{s,i,j}$ are the teacher and student logits, N and M represent batch size and token length per sample, and γ is a scaling factor. $T_{i,j}$ and $S_{i,j}$ are the logits, while $\sigma_{t,i}$ and $\sigma_{s,i}$ are their respective dynamic standard deviations. This dynamic adjustment ensures optimal knowledge transfer between teacher and student models.

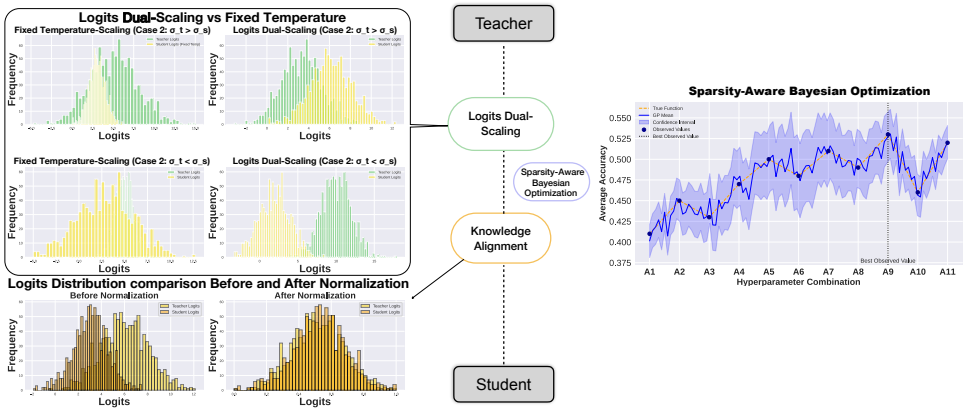

Figure 1: Schematic overview of our framework, which involves Logits Dual-Scaling, Knowledge Alignment, and Logits-Aware Bayesian optimization. Our Logits Dual-Scaling scaling and Knowledge Alignment enable the student to learn complex relationships and key features, while Bayesian optimization efficiently searches for observed optimal hyperparameters to maximize student performance.

Proof that Logits Dual-Scaling Bridges Teacher-Student Gap: Let the gap between the teacher model P and student model Q be:

$$\begin{aligned} G(P,Q) &= D_{KL}[P||Q], \\ &= \mathbb{E}_u[\log P(v|u) - \log Q(v|u)], \end{aligned} \tag{3}$$

Next, let us set the standard deviations of the teacher and student models, σ_t and σ_s as scaling value for teacher and student, respectively, such that:

$$t_t = \sigma_t, \tag{4}$$

$$t_s = \sigma_s, \tag{5}$$

where T and S are the logits from the teacher and student models, and t_t and t_s are the corresponding scaling value coefficients. The standard deviations σ_t and σ_s of the teacher and student models, and substituting into the gap formula, we get:

$$\begin{aligned} G\left(P,Q\right) = \mathbb{E}_u \Bigg[&\left(\frac{1}{\sigma_t} - \frac{1}{\sigma_s}\right) T \\ &- \left(\log \sum_j \exp\left(\frac{T_j}{\sigma_t}\right) - \log \sum_k \exp\left(\frac{S_k}{\sigma_s}\right)\right)\Bigg] \end{aligned} \tag{6}$$

From this expression, we can observe two key scenarios: **Scenario 1** ($\sigma_t > \sigma_s$)**:** The overall discrepancy $G(P,Q)$ increases, primarily due to the dominance of the second term's positive contribution. Despite this increase, the teacher model's smoother output distribution facilitates the transfer of more complex information, enhancing the student model's learning process. **Scenario 2** ($\sigma_s > \sigma_t$)**:** The overall discrepancy $G(P,Q)$ decreases, driven by the second term's negative contribution. In this case, the teacher model provides concentrated, key decision information, while the student model's smoother distribution effectively captures essential features, leading to improved performance. **Analyses:** The impact of the discrepancy variation on knowledge transfer depends on the relative standard deviations of the teacher and student models. When the teacher model's standard deviation is greater, the second term in the discrepancy $G(P,Q)$ contributes more positively, increasing the overall discrepancy. Despite this, the smoother distribution of the teacher model provides richer information, improving the student's ability to learn complex relationships. Conversely, when the student model's standard deviation is greater, the negative contribution of the second term becomes more significant, decreasing $G(P,Q)$. In this case, the teacher's output is more concentrated, providing key decision information, while the student's smoother distribution better mimics the teacher's. This dynamic, data-driven adjustment of scaling values softens the output distributions during the learning process, further enhancing knowledge transfer, as shown in Fig. 1.

3.2 Knowledge Alignment Strategy

During the distillation process, we align the varying intermediate distributions from the student and teacher models to a uniform size, employing a fixed-dimension method for consistency. The motivation behind this approach is to bridge the gap between the teacher and student models' distributions, preventing gradient explosion and ensuring

Hyperparameter	Values
Hard Label Weight	$0.1, 1, 10, 20$
Soft Label Weight	$1e-8, 1e-7, 1e-3, 1, 10, 20$
Starting scaling value	$1, 2, 3.....20$
Logits Normalization	"none", "Max-Min", "standardize"
Hidden States Normalization	"none", "Max-Min", "standardize"
Knowledge Alignment Strategy[2]	None, Last_Layer, Last_Middle Layer
KD Type	KL, Logits Dual-Scaling

Table 2: Overview of hyperparameters search space. KLD is Kullback–Leibler divergence. Knowledge Alignment Strategy choices represent a range of layers instead of specific the last layer or the middle layer.

effective knowledge transfer. We define the loss function $\mathcal{L}_k$ as follows:

$$\mathcal{L}_k = \sum_{l=1}^{L} \frac{\beta_l}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} \|h_s[l,i,j] - h_t[l,i,j]\|_2^2 \quad (7)$$

$$\mathcal{L}_{total} = \mathcal{L}_k + \mathcal{L}_d \quad (8)$$

The total loss combines $\mathcal{L}_k$ and $\mathcal{L}_d$ as shown in Equation 8. In Equation 7, L, β_l, h_s, and h_t represent the selected layers, scaling factors, and hidden states of the student and teacher models. Since their intermediate layers often differ in distribution, we apply min-max normalization to align these before using Equation 7.

$$L_i' = \frac{x_i - \min(L)}{\max(L) - \min(L) + \epsilon} \quad (9)$$

x_i denotes the distribution value, $\min(L)$ and $\max(L)$ are the distribution's minimum and maximum, and ϵ is a small constant for numerical stability. This normalization, applied to intermediate layers, narrows the distribution gap between teacher and student models and prevents gradient explosion.

3.3 Logits-Aware Bayesian Optimization

We utilize Bayesian optimization during the distillation phase to streamline the hyperparameter selection process and boost the model performance in Table. 2. The overview searching process is in Fig. 1 Bayes Optimization part.

Search Space: The hyperparameter search space encompasses a range of options, as summarized in Table. 2. It includes Logits-Aware parameters such as pruning thresholds and sparse regularization factors, alongside traditional distillation parameters like scaling value and layer configurations.

Search Method: We employ a modified Gaussian Process (GP) as a surrogate model, specifically tailored for logits discrepancy. The GP, denoted as $f \sim GP(\mu, K)$, uses a mean function μ initialized to reflect model characteristics and a covariance function K designed to capture sparse parameter interactions. This GP is iteratively updated based on hyperparameter evaluations (X, Y), refining its understanding of the model-specific hyperparameter space. To balance exploration and exploitation, we implement an adaptive sampling strategy, ensuring efficient optimization.

To guide the search for optimal hyperparameter configurations, we introduce a Logits-Aware Expected Improvement (LAEI) acquisition function:

$$\begin{aligned} LAEI(x^*) =& (\mu(x^*) - f(x^+) - \xi)\Phi(Z) \\ &+ \sigma(x^*)\phi(Z) + \alpha S(x^*) \end{aligned} \quad (10)$$

$$Z = \frac{\mu(x^*) - f(x^+) - \xi}{\sigma(x)} \quad (11)$$

Here, $\mu(x^*)$ and $\sigma(x^*)$ represent the predicted mean and variance, $f(x^+)$ is the best observed value, and ξ is a small parameter for balancing exploration and exploitation. Φ and ϕ represent the cumulative and probability density functions of the standard normal distribution. The novel term $S(x^*)$ integrates information from the Logits Dual-Scaling strategy and Knowledge Alignment Module, with α serving as an overall balancing factor for sparse models:

$$S(x^*) = \lambda_1 D(\sigma_t, \sigma_s) + \lambda_2 A(h_t, h_s) \quad (12)$$

where $D(\sigma_t, \sigma_s) = \left| \frac{1}{\sigma_t} - \frac{1}{\sigma_s} \right|$ measures logits discrepancy, and $A(h_t, h_s) = \sum_{l=1}^{L} ||h_t[l] - h_s[l]||_2^2$ quantifies hidden state alignment across L layers. Coefficients λ_1 and λ_2 balance these terms.

Advantages: LAEI offers key advantages for sparse models by incorporating sparsity-related information into the hyperparameter search. It optimizes knowledge transfer through dynamic scaling value adjustment ($D(\sigma_t, \sigma_s)$) and hidden state alignment, while balancing exploration and exploitation. This approach enables LAEI to effectively navigate the complex landscape of compact LLMs, improving distillation performance in resource-constrained scenarios. A detailed theoretical analysis is provided in Appendix.

Compact LLM distillation guidelines: We summarize some deep insights based on search results in Fig. 2: Compressed LLM distillation always favors deep intermediate knowledge, greater scaling

Model	Method	OPQA	Hella-SWAG	Wino-grande	PIQA	ARC_E	BoolQ	ARC_C	QNLI	QQP	SST-2	Aver-age	Wiki-Text2
LLaMA3 70B	Teacher Model	48.60	84.92	81.14	84.49	85.94	85.26	64.33	52.88	65.92	79.70	73.32	2.92
LLaMA3 8B	Dense Model	45.00	79.11	73.24	80.74	77.86	81.16	53.16	50.49	57.31	68.23	66.63	10.18
	Sparse Model (wo LoRA)	39.00	67.67	69.22	76.61	60.86	65.99	37.88	51.42	62.67	51.83	58.32	20.24
	Sparse Model (w LoRA)	39.55	70.85	69.06	76.62	66.55	69.87	39.84	48.68	62.16	64.51	60.77	17.00
	Stand KD	38.2	71.79	71.36	75.18	68.76	66.43	40.86	47.76	**63.59**	60.78	60.47	18.18
	BayesKD(Random Search)	37.45	72.37	70.28	76.69	67.17	68.48	42.53	49.29	55.04	62.31	60.16	17.21
	BayesKD (SABO)	**41.80**	**74.32**	**73.24**	**78.29**	**71.13**	**76.24**	**45.56**	**49.86**	56.37	**67.78**	**63.46**	**15.77**
Qwen2 72B	Teacher Model	49	85.59	79.16	83.24	80.64	89.54	60.15	73.7	73.24	93.58	76.78	6.65
Qwen2 7B	Dense Model	44.40	78.77	71.9	81.18	74.58	84.89	50.00	59.40	73.28	92.43	71.08	9.32
	Sparse Model (wo LoRA)	38.52	67.37	68.11	75.31	58.17	68.64	35.64	61.63	63.68	87.84	62.49	18.55
	Sparse Model (w LoRA)	39.06	71.46	69.93	77.02	63.62	72.61	38.52	58.65	70.41	90.02	65.13	15.55
	Stand KD	37.73	70.53	68.29	75.57	65.72	69.26	38.47	57.31	73.03	88.19	64.41	16.62
	BayesKD(Random Search)	39.28	69.66	70.06	76.78	64.84	76.99	37.99	56.66	74.52	90.18	65.70	15.74
	BayesKD (SABO)	**41.27**	**73.98**	**71.9**	**78.7**	**67.97**	**79.53**	**42.92**	**58.5**	**77.56**	**92.31**	**68.46**	**14.42**

Table 3: The main results (Qwen and LLaMA3) from our multi-task testing, with the exception of Wikitext-2, were derived from the Language Model Evaluation Harness[1]. For Wikitext-2, the Perplexity (PPL) metric was employed, whereas accuracy served as the metric for all other tasks. Sparsity ratio is 25%. *BayesKD (SABO)* employs *Logits-Aware Bayesian Optimization* for hyperparameter tuning, while *BayesKD (Random Search)* uses *random search*.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.

value sparsity, and smaller loss weights. This is evident from our exploration of different "KD Layers" options, including "Middle& Last" and "Max-Min Normalization", which suggest leveraging intermediate representations from the teacher model's deeper layers. Additionally, the wide range of scaling values, reaching up to 16, indicates a preference for higher scaling value sparsity. Furthermore, KD Loss Weight with options like potentially small values implies that compressed LLM distillation may benefit from using smaller loss weights, potentially to avoid overriding the student model's original capabilities.

4 Experiments

Experimental Setup We extracted 13,000 training samples and 2,000 validation samples from the cleaned Alpaca dataset[3] 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 as introduced in the Introduction section. We established a comprehensive experimental setup using a diverse range of LLMs. Our student models include Bloom-7b1 (Workshop et al., 2022), LLaMA-7b-hf (Touvron et al., 2023), Vicuna-7b-v1.1 (Zheng et al., 2023), TinyLLaMA-1.1b (Zhang et al., 2024), LLaMA-3 8B (Dubey et al., 2024), and Qwen-2 7B (Yang et al., 2024). These were paired with appropriate teacher models: LLaMA-13b for the first four, LLaMA-3 70B for LLaMA-3 8B, and Qwen-2 72B for Qwen-2 7B. This setup allows for a comprehensive evaluation across various model architectures and sizes. All experiments were conducted on 8 NVIDIA A100 GPUs with a sparsity ratio of 25% for the student models.

4.1 Results and Analysis

In Table 4, we evaluate BayesKD on TinyLLaMA1.1B, LLaMA 7B, Vicuna 7B, and Bloom 7B models. Across these configurations, BayesKD consistently outperforms baseline methods. For instance, with the LLaMA 7B model, BayesKD achieves an average score of 60.08%, surpassing Standard KD (55.68%) and Sparse Model with LoRA (57.48%) by significant margins. Similar trends are observed for other models, with

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

[2] The "Intermediate Layer Config" options allow flexible specification or removal of layers between teacher and student models. "Last" and "Middle" refer to regions in the architecture, not specific layer counts.

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

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
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
	BayesKD(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
	BayesKD (SABO)	34.00	**54.34**	**58.25**	70.78	**49.45**	54.19	**29.78**	**53.22**	**57.36**	**80.96**	**54.23**	**22.00**
LLaMA 7b	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	**77.40**	57.48	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
	BayesKD(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
	BayesKD (SABO)	**44.60**	**72.01**	**65.36**	**78.10**	**65.81**	**67.80**	**39.59**	52.08	55.36	60.09	**60.08**	**19.05**
Vincuna 7b	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	20.95
	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
	BayesKD(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
	BayesKD (SABO)	**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	190.57
	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	152.67
	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
	BayesKD(Random Search)	31.60	38.12	56.35	67.79	46.84	61.96	26.71	49.33	38.13	61.01	47.99	135.66
	BayesKD (SABO)	**32.20**	**39.36**	55.09	**68.55**	**46.89**	**61.98**	**27.65**	**50.25**	39.39	**61.35**	**48.25**	**124.49**

Table 4: The main results (LLaMA1, TinyLLaMA, Vicuna) from our multi-task testing derived from the Language Model Evaluation Harness. Sparsity ratio is 25%. *BayesKD (SABO)* employs *Logits-Aware Bayesian Optimization* for hyperparameter tuning, while *BayesKD (Random Search)* uses *random sampling*.

Type	**Model**	**Base Parameters (B)**	**Pruned Parameters (B)**	**Final Pruned Ratio**
Teacher	LLaMA13B	13B	-	-
Student	Tiny-LLaMA1.1B	1.1B	0.961B	0.8735
	Bloom7b	7.069B	6.282B	0.8887
	Vicuna7b	6.738B	5.423B	0.8048
	LLaMA7b	6.738B	5.423B	0.8048

Table 5: Overview of model parameter adjustments. All parameter values are expressed in billions (B).

BayesKD showing particular strength in challenging tasks such as ARC-challenge and HellaSwag. Table 3 extends our analysis to the latest models, LLaMA-3 and Qwen-2. For the LLaMA-3 8B model, BayesKD achieves an average accuracy of 63.46% across tasks, outperforming Standard KD (60.47%) and Sparse Model with LoRA (60.77%). Notably, on complex reasoning tasks like ARC-Challenge and HellaSwag, BayesKD shows substantial improvements of 4.7% and 2.53% respectively over Standard KD. The Qwen-2 7B results further corroborate BayesKD's effectiveness. Our method achieves an average accuracy of 68.48%, significantly surpassing Standard KD (64.41%) and Sparse Model with LoRA (65.13%). Particularly impressive are the improvements in tasks like BoolQ and QQP, where BayesKD outperforms Standard KD by 10.27% and 4.53% respectively.

Across all model sizes and architectures, we observe that the performance gap between BayesKD and baseline methods widens as model size increases in both general NLU benchmarks and instruct-follow benchmarks 8. The consistent performance improvements validate the generalizability and robustness of BayesKD.

4.2 Ablation Study

We conducted an ablation study to evaluate the individual contributions of Logits Dual-Scaling (DS), the Knowledge Alignment Strategy (KAS), and Kullback–Leibler (KL) divergence to the distillation performance of the TinyLLaMA1.1B model.

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
+KL	33.20	53.52	58.09	69.53	46.55	54.92	29.03	49.28	45.07	69.27	51.59	25.85
+ KL + KAS	34.40	54.07	57.46	71.22	49.24	53.88	29.18	53.18	55.73	71.56	53.10	22.15
+ KL + DS	33.00	54.11	57.3	70.89	50.00	56.97	30.2	54.55	55.96	74.2	53.72	23.93
+ KL + KAS + DS	34.00	54.34	58.25	70.78	49.45	54.19	29.78	53.21	57.36	80.96	54.23	22.00

Table 6: Ablation study results in 0.25 sparsity ratio in TinyLLaMA1.1b on various NLP tasks. KL is KL divergence, KAS is Knowledge Alignment Strategy and DS is Logits Dual-Scaling. All parameters are the same and searched from Logits-Aware Bayesian Opitmization

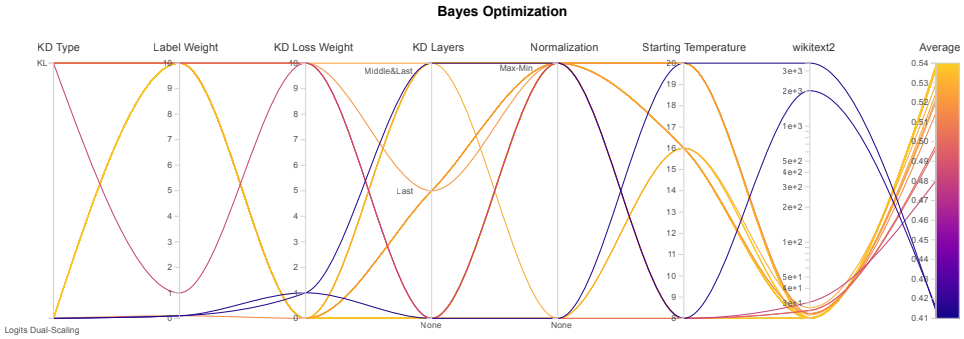

Figure 2: Visualization of search results during Bayesian distillation optimization in LLaMA13B-TinyLLaMA1.1B. The yellow line illustrates the Logits Dual-Scaling Strategy approach combined with max-min normalization (Knowledge Alignment Strategy), leading to lower perplexity on WikiText-2 and higher accuracy across tasks.

Method	Average	Wiki-text2
Naïve Bayesian Search	53.37	24.25
SABO	54.23	22.00

Table 7: Comparison of performance between SABO and Naïve Bayesian Search in 0.25 sparsity ratio in TinyLLaMA1.1b on Wiki-text2.

Starting from the full configuration, each strategy was removed in turn. As shown in Table 6, excluding DS reduced average accuracy by 1.13% across 10 tasks and increased perplexity on WikiText-2 by 2.29 points. Removing KAS produced similar effects, with a 1.13% drop in accuracy and a 2.85-point perplexity increase. Eliminating KL divergence resulted in a 1.41% accuracy decrease and a 4.44-point perplexity increase, demonstrating the effectiveness of incorporating KD regularization beyond the baseline sparse model. Furthermore, the main results in Table 4 already demonstrate the superiority of Bayesian search over random search, making a separate ablation study for Bayesian search unnecessary.

These findings confirm the contributions of Logits Dual-Scaling, Knowledge Alignment Strategy, and Bayesian optimization to the model's overall performance, revealing their relative importance during the model training process.

Table 8: Instruction-following datasets in sparse model. GPT-3 Translation Tasks are constructed using datasets such as WMT14 (French-English), WMT16 (Romanian-English, German-English), and IWSLT2017 (English-Arabic). Each task is augmented with instruction prompts to adapt the dataset for instruction-following evaluation.

Model	**Method**	**gpt3_translation_tasks**
LLaMA3 70B	Teacher Model	54.31
LLaMA3 8B	Dense Model	31.94
	Sparse Model (wo LoRA)	19.26
	Sparse Model (w LoRA)	22.14
	Stand KD	21.88
	BayesKD (SABO)	**24.59**
Qwen2 72B	Teacher Model	31.76
Qwen2 7B	Dense Model	12.62
	Sparse Model (wo LoRA)	7.62
	Sparse Model (w LoRA)	8.75
	Stand KD	8.65
	BayesKD (SABO)	**9.72**

5 Conclusion

In this paper, we propose BayesKD, a novel framework for distilling LLMs onto compact student models. Our approach introduces three key strategies: 1) a logits dual-scaling mechanism to bridge the logit distribution gap between teacher and stu-

dent models across tasks, 2) a knowledge alignment module using min-max normalization to align intermediate layer distributions, and 3) a logit-aware Bayesian optimization search to efficiently identify optimal hyperparameters tailored for compact model distillation.

6 Acknowledgements

The research was supported by Theme-based Research Scheme (T45-205/21-N) from Hong Kong RGC, and Generative AI Research and Development Centre from InnoHK.

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

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

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.

Lu Hou, Zhiqi Huang, Lifeng Shang, Xin Jiang, Xiao Chen, and Qun Liu. 2020. Dynabert: Dynamic bert with adaptive width and depth. *Advances in Neural Information Processing Systems*, 33:9782–9793.

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

Ying Jin, Jiaqi Wang, and Dahua Lin. 2023. Multi-level logit distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24276–24285.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. Distillm: Towards streamlined distillation for large language models. In *Forty-first International Conference on Machine Learning*.

Jianquan Li, Xiaokang Liu, Honghong Zhao, Ruifeng Xu, Min Yang, and Yaohong Jin. 2020. BERT-EMD: Many-to-many layer mapping for BERT compression with earth mover's distance. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3009–3018, Online. Association for Computational Linguistics.

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

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.

Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. 2020. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 5191–5198.

Mehdi Rezagholizadeh, Aref Jafari, Puneeth S.M. Saladi, Pranav Sharma, Ali Saheb Pasand, and Ali Ghodsi. 2022. Pro-KD: Progressive distillation by following the footsteps of the teacher. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4714–4727, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Julio Ariel Romero, Roberto Sanchis, and Elena Arrebola. 2015. Experimental study of event based pid controllers with different sampling strategies. application to brushless dc motor networked control system. In *2015 XXV international conference on information, communication and automation technologies (ICAT)*, pages 1–6. IEEE.

Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. 2024a. A simple and effective pruning approach for large language models. In *The Twelfth International Conference on Learning Representations*.

Shangquan Sun, Wenqi Ren, Jingzhi Li, Rui Wang, and Xiaochun Cao. 2024b. Logit standardization in knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15731–15740.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019a. Patient knowledge distillation for BERT model compression. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4323–4332, Hong Kong, China. Association for Computational Linguistics.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019b. Patient knowledge distillation for BERT model compression. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4323–4332, Hong Kong, China. Association for Computational Linguistics.

Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. 2019. Distilling task-specific knowledge from bert into simple neural networks. *arXiv preprint arXiv:1903.12136*.

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. 2023. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2020. Well-read students learn better: On the importance of pre-training compact models.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022a. Emergent abilities of large language models. *Transactions on Machine Learning Research*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems (NeurIPs)*, 35:24824–24837.

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*.

Yimeng Wu, Mehdi Rezagholizadeh, Abbas Ghaddar, Md Akmal Haidar, and Ali Ghodsi. 2021. Universal-KD: Attention-based output-grounded intermediate layer knowledge distillation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7649–7661, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

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.

Chen Zhang, Dawei Song, Zheyu Ye, and Yan Gao. 2023a. Towards the law of capacity gap in distilling language models. *arXiv preprint arXiv:2311.07052*.

Mingyang Zhang, Chunhua Shen, Zhen Yang, Linlin Ou, Xinyi Yu, Bohan Zhuang, et al. 2023b. Pruning meets low-rank parameter-efficient fine-tuning. *arXiv preprint arXiv:2305.18403*.

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*.

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*.

Qihuang Zhong, Liang Ding, Li Shen, Juhua Liu, Bo Du, and Dacheng Tao. 2024. Revisiting knowledge distillation for autoregressive language models. *arXiv preprint arXiv:2402.11890*.

A Detailed Proof of Double Scaling Values Coefficients

The discrepancy between the teacher model P and the student model Q is defined as:

$$\begin{aligned} G(P,Q) &= D_{KL}[P||Q] \\ &= \mathbb{E}_u[\log P(v|u) - \log Q(v|u)] \end{aligned} \tag{13}$$

In this context, D_{KL} denotes the Kullback-Leibler divergence, which measures how one probability distribution diverges from a second, expected probability distribution. Here, u represents the input data, while v denotes the corresponding output.

Given that T and S represent the logits of the teacher and student models, respectively, and t_t and t_s are the corresponding scaling value coefficients, we can express the softmax functions for the teacher and student models. The softmax function is defined as:

$$\text{softmax}(z_i) = \frac{\exp(z_i)}{\sum_j \exp(z_j)}$$

By substituting the logits and scaling value coefficients into the definition of the discrepancy, we have:

$$\begin{aligned} G(P,Q) &= \mathbb{E}_u \left[\log \text{softmax}\left(\frac{T}{t_t}\right) - \log \text{softmax}\left(\frac{S}{t_s}\right) \right] \\ &= \mathbb{E}_u \left[\frac{T}{t_t} - \log \sum_j \exp\left(\frac{T_j}{t_t}\right) \right. \\ &\quad \left. - \left(\frac{S}{t_s} - \log \sum_k \exp\left(\frac{S_k}{t_s}\right) \right) \right] \end{aligned} \tag{14}$$

Next, let us introduce the standard deviations of the teacher and student models, σ_t and σ_s, respectively. These standard deviations are related to the scaling value coefficients as follows:

$$\begin{aligned} t_t &= \sigma_t t \\ t_s &= \sigma_s t \end{aligned} \tag{15}$$

where t is a baseline scaling value. Substituting these expressions into the discrepancy formula, we get:

$$\begin{aligned} G\left(P,Q\right) &= \mathbb{E}_u \left[\left(\frac{1}{\sigma_t} - \frac{1}{\sigma_s} \right) T \right. \\ &\quad \left. - \left(\log \sum_j \exp\left(\frac{T_j}{\sigma_t}\right) - \log \sum_k \exp\left(\frac{S_k}{\sigma_s}\right) \right) \right] \end{aligned} \tag{16}$$

To further analyze the impact of the standard deviations on the discrepancy, we consider two scenarios:

Scenario 1: $\sigma_t > \sigma_s$

In this scenario, the standard deviation of the teacher model is greater than that of the student model. This implies:

$$\frac{1}{\sigma_t} < \frac{1}{\sigma_s}$$

For the first term in the discrepancy formula:

$$\left(\frac{1}{\sigma_t} - \frac{1}{\sigma_s} \right) < 0$$

Since T is generally positive, it follows that:

$$\left(\frac{1}{\sigma_t} - \frac{1}{\sigma_s} \right) T < 0$$

This means that the first term contributes negatively to $G(P,Q)$, thereby reducing the discrepancy.

For the second term:

$$\log \sum_j \exp\left(\frac{T_j}{\sigma_t}\right) - \log \sum_k \exp\left(\frac{S_k}{\sigma_s}\right)$$

Since $\sigma_t > \sigma_s$, the logits of the teacher model are less magnified compared to those of the student model. Therefore, we have:

$$\log \sum_j \exp\left(\frac{T_j}{\sigma_t}\right) < \log \sum_k \exp\left(\frac{S_k}{\sigma_s}\right)$$

This indicates that the second term contributes positively to $G(P,Q)$. To prove that the second term's contribution is larger, we consider the properties of the log-sum-exp function. The log-sum-exp function is more sensitive to changes in variance compared to a linear function. Specifically, the log-sum-exp function can be approximated by the maximum logit value when the scaling value is low, making it highly sensitive to the largest logits. Using this approximation:

$$\log \sum_j \exp\left(\frac{T_j}{\sigma_t}\right) \approx \frac{1}{\sigma_t} \max_j T_j$$

$$\log \sum_k \exp\left(\frac{S_k}{\sigma_s}\right) \approx \frac{1}{\sigma_s} \max_k S_k$$

Thus, the second term becomes:

$$\frac{1}{\sigma_t} \max_j T_j - \frac{1}{\sigma_s} \max_k S_k$$

Given $\frac{1}{\sigma_t} < \frac{1}{\sigma_s}$, we have:

$$\left| \frac{1}{\sigma_t} \max_j T_j - \frac{1}{\sigma_s} \max_k S_k \right| = \frac{1}{\sigma_s} \max_k S_k - \frac{1}{\sigma_t} \max_j T_j$$

For the first term:

$$\left| \left(\frac{1}{\sigma_t} - \frac{1}{\sigma_s} \right) T \right| = \left(\frac{1}{\sigma_s} - \frac{1}{\sigma_t} \right) T$$

Comparing the two terms:

$$\left(\frac{1}{\sigma_s} - \frac{1}{\sigma_t} \right) T \quad \left(\frac{1}{\sigma_s} \max_k S_k - \frac{1}{\sigma_t} \max_j T_j \right)$$

Since $\max_j T_j$ and $\max_k S_k$ are typically much larger than T, the second term's contribution is usually significantly larger than the first term's contribution.

Summary:

In this scenario, the negative contribution of the first term is smaller in magnitude, while the positive contribution of the second term is larger. This leads to an overall increase in the discrepancy $G(P, Q)$.

Scenario 2: $\sigma_t < \sigma_s$

In this scenario, the standard deviation of the student model is greater than that of the teacher model. This implies:

$$\frac{1}{\sigma_t} > \frac{1}{\sigma_s}$$

For the first term in the discrepancy formula:

$$\left(\frac{1}{\sigma_t} - \frac{1}{\sigma_s}\right) > 0$$

Since T is generally positive, it follows that:

$$\left(\frac{1}{\sigma_t} - \frac{1}{\sigma_s}\right) T > 0$$

This means that the first term contributes positively to $G(P, Q)$, thereby increasing the discrepancy.

For the second term:

$$\log \sum_j \exp\left(\frac{T_j}{\sigma_t}\right) - \log \sum_k \exp\left(\frac{S_k}{\sigma_s}\right)$$

Since $\sigma_s > \sigma_t$, the logits of the teacher model are more magnified compared to those of the student model. Therefore, we have:

$$\log \sum_j \exp\left(\frac{T_j}{\sigma_t}\right) > \log \sum_k \exp\left(\frac{S_k}{\sigma_s}\right)$$

This indicates that the second term contributes negatively to $G(P, Q)$.

Why the Second Term's Contribution is Larger:

Again, considering the properties of the log-sum-exp function, we use the same approximation:

$$\log \sum_j \exp\left(\frac{T_j}{\sigma_t}\right) \approx \frac{1}{\sigma_t} \max_j T_j$$

$$\log \sum_k \exp\left(\frac{S_k}{\sigma_s}\right) \approx \frac{1}{\sigma_s} \max_k S_k$$

Thus, the second term becomes:

$$\frac{1}{\sigma_t} \max_j T_j - \frac{1}{\sigma_s} \max_k S_k$$

Given $\frac{1}{\sigma_t} > \frac{1}{\sigma_s}$, we have:

$$\left|\frac{1}{\sigma_t} \max_j T_j - \frac{1}{\sigma_s} \max_k S_k\right| = \frac{1}{\sigma_t} \max_j T_j - \frac{1}{\sigma_s} \max_k S_k$$

For the first term:

$$\left|\left(\frac{1}{\sigma_t} - \frac{1}{\sigma_s}\right) T\right| = \left(\frac{1}{\sigma_t} - \frac{1}{\sigma_s}\right) T$$

Comparing the two terms:

$$\left(\frac{1}{\sigma_t} - \frac{1}{\sigma_s}\right) T$$
$$\left(\frac{1}{\sigma_t} \max_j T_j - \frac{1}{\sigma_s} \max_k S_k\right)$$

Since $\max_j T_j$ and $\max_k S_k$ are typically much larger than T, the second term's contribution is usually significantly larger than the first term's contribution.

Summary:

In this scenario, the positive contribution of the first term is smaller in magnitude, while the negative contribution of the second term is larger. This leads to an overall decrease in the discrepancy $G(P, Q)$.

Conclusion

The analysis reveals that the impact of the discrepancy variation on knowledge transfer differs depending on the relative standard deviations of the teacher and student models:

- When the standard deviation of the teacher model (σ_t) is greater than that of the student model (σ_s), the positive contribution of the second term is more significant. This results in an overall increase in the discrepancy $G(P, Q)$. Although the discrepancy increases, the teacher model's output distribution is smoother and contains more information. This helps the student model to learn more details and complex relationships, thereby improving its performance.

- Conversely, when the standard deviation of the student model (σ_s) is greater than that of the teacher model (σ_t), the negative contribution of the second term is more significant. This results in an overall decrease in the discrepancy $G(P, Q)$. Although the

Model	Method	Average	WikiText2
LLaMA 13B	Teacher Model	64.32	11.58
LLaMA 7B	Sparse Model	54.71	22.54
	Dense Model	63.78	12.62
	MiniLLM	55.10	22.15
	BayesKD (SABO)	**60.08**	**19.05**

Table 9: MiniLLM in sparse model

discrepancy decreases, the teacher model's output is more concentrated, providing key decision information. The student model's output distribution is smoother and more accurately mimics the teacher model's output distribution, capturing the main decision features and thus improving its performance.

Therefore, the impact of the discrepancy variation on knowledge transfer requires a detailed analysis of the specific probability distributions of the teacher and student models. Understanding these dynamics is crucial for optimizing the performance of the student model in various training scenarios.

B Experimental Details

All the experiments are run on 8 A100 GPUs. The experiments running time is 55 minutes in 7B models and 30 minutes in 1.1B model. The memory requires 75GB if we load 13B and 5.4B(compressed) model in 4 bytes/parameter to train in Batch size 4.

Ablation study with error bar:

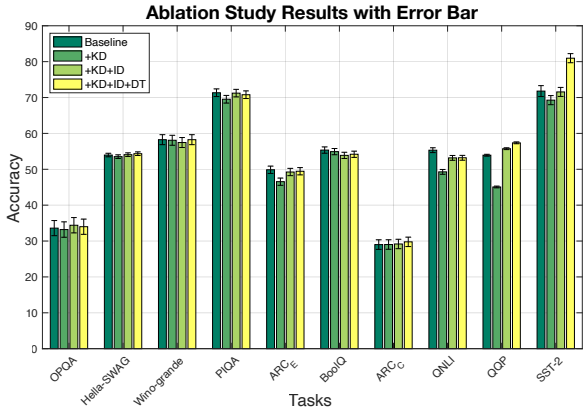

Figure 3: The ablation study results with error bar

P-values: We calculated the variance and mean based on 6 extra experiments in LLaMA 7B group. Then, we used independent samples t-test to statistically determine whether BayesKD (SABO) has a significant advantage in Table 10. The table shows the P-values of BayesKD (SABO) compared to other methods and whether it has a significant advantage.

Based on the P-values table (Table 10), we can summarize the performance of BayesKD (SABO) compared to Stand KD, Sparse Model (w LoRA), and BayesKD (Random Search) as follows:

BayesKD (SABO) vs. Stand KD: BayesKD (SABO) demonstrates significant advantages over Stand KD on 8 out of 10 datasets, with the exceptions being SST-2 and ARC_C. This indicates that BayesKD (SABO) generally outperforms Stand KD in most cases, suggesting its effectiveness as an improved knowledge distillation method.

BayesKD (SABO) vs. Sparse Model (w LoRA): BayesKD (SABO) shows significant advantages over Sparse Model (w LoRA) on all datasets except SST-2, where Sparse Model (w LoRA) significantly outperforms BayesKD (SABO). This suggests that while BayesKD (SABO) is generally more effective than Sparse Model (w LoRA), there may be specific cases where Sparse Model (w LoRA) is more suitable.

BayesKD (SABO) vs. BayesKD (Random Search) : BayesKD (SABO) significantly outperforms BayesKD (Random Search) on 7 out of 10 datasets, with the exceptions being Winogrande, BoolQ, and QQP, where the differences are not statistically significant. This demonstrates the superiority of the Bayesian search approach over the random search method in most cases, highlighting the importance of an efficient search strategy in sparse knowledge distillation.

Overall, these comparisons provide strong evidence for the effectiveness and advantages of BayesKD (SABO) as a knowledge distillation method. However, it is important to note that there are specific cases where other methods may be more suitable, such as Stand KD for ARC_C and Sparse Model (w LoRA) for SST-2. Therefore, the choice of knowledge distillation method should take into account the characteristics of the target dataset and the specific requirements of the application.

Table 11 shows the mean and variance of different methods on different datasets corresponding to the P-values table.

BayesKD (SABO) vs. Stand KD in Translation tasks (Table 12 and Table 13): BayesKD (SABO) consistently outperforms Stand KD across all translation tasks for both LLaMA3 and Qwen2 models. For example, in Table 12, BayesKD (SABO) achieves an average score of 24.25 compared to Stand KD's 21.63, while in Table 13, it improves the average score to 13.37, outperforming

Dataset	vs. BayesKD (Random Search)	vs. Sparse Model (w LoRA)	vs. Sparse Model (wo LoRA)	vs. Stand KD
OPQA	**5.55e-16 (Significant)**	**0.0018 (Significant)**	**1.11e-16 (Significant)**	**0.0002 (Significant)**
Hella-SWAG	**1.07e-08 (Significant)**	**0.0008 (Significant)**	**2.14e-10 (Significant)**	**0.0002 (Significant)**
Winogrande	0.0584 (Not Significant)	**0.0064 (Significant)**	**1.23e-06 (Significant)**	**0.0005 (Significant)**
PIQA	**3.58e-07 (Significant)**	**0.0005 (Significant)**	**4.41e-05 (Significant)**	**4.61e-05 (Significant)**
ARC_E	**1.04e-12 (Significant)**	**0.0042 (Significant)**	**3.79e-15 (Significant)**	**0.0002 (Significant)**
BoolQ	0.6056 (Not Significant)	**1.03e-10 (Significant)**	**1.39e-08 (Significant)**	**0.0034 (Significant)**
ARC_C	**5.50e-07 (Significant)**	**0.0179 (Significant)**	**1.15e-07 (Significant)**	0.1008 (Not Significant)
QQP	0.5941 (Not Significant)	**6.57e-07 (Significant)**	**6.02e-11 (Significant)**	**2.90e-09 (Significant)**
QNLI	**1.36e-06 (Significant)**	**2.06e-06 (Significant)**	**0.0207 (Significant)**	**0.0034 (Significant)**
SST-2	**6.51e-11 (Significant)**	5.17e-12 (Significantly Lower)	8.78e-10 (Significantly Lower)	0.1765 (Not Significant)

Table 10: P-values of independent samples t-test comparing the performance of BayesKD (SABO) with other methods on various datasets in the LLaMA7B model. Values in parentheses indicate whether the difference is statistically significant at $\alpha = 0.05$. Significant results are shown in **bold**.

Dataset	Metric	BayesKD (SABO)	Dense Model	Sparse Model (w LoRA)	BayesKD (Random Search)	Stand KD	Sparse Model (wo LoRA)
OPQA	Variance	2.089256	11.51863	1.666222	0.096833	1.394181	0.318456
	Mean	44.72667	44.48	39.51667	39.31	38.53167	36.02667
Hella-SWAG	Variance	9.417667	9.729522	2.284522	2.719881	1.034756	0.539322
	Mean	72.62	76.80667	67.07333	64.48167	67.24333	62.51667
Winogrande	Variance	2.964481	5.893789	2.681358	1.211381	0.361258	1.394889
	Mean	65.69833	70.01333	62.185	63.75167	62.805	59.56667
PIQA	Variance	5.300647	1.431247	0.897581	0.2142	0.401414	1.021881
	Mean	78.46167	79.40833	74.83833	73.24	74.00167	73.39833
ARC_E	Variance	7.640856	13.64682	15.51068	2.113767	1.390933	0.471181
	Mean	66.22333	73.01333	59.16833	52.72	59.72	51.28167
BoolQ	Variance	1.347033	30.72663	0.929522	6.743989	6.166856	2.347567
	Mean	68.04	75.935	57.55667	67.33667	62.61667	58.83
ARC_C	Variance	2.630947	10.13075	3.144925	1.903247	1.539881	0.712847
	Mean	39.97167	45.45833	37.435	33.62833	38.22833	32.92167
QNLI	Variance	0.779525	0.698489	0.009356	0.031822	0.067422	1.713947
	Mean	52.235	51.33333	51.54667	51.43667	51.65667	53.45167
QQP	Variance	11.05059	1.177981	2.143789	3.280289	0.970514	0.223514
	Mean	57.01667	48.18833	48.96667	57.72667	46.66167	46.13833
SST-2	Variance	5.483689	3.024456	17.58463	1.010225	9.477692	12.31193
	Mean	60.31333	76.67667	85.685	49.285	57.355	74.49

Table 11: Variance and mean for each dataset across different methods in LLaMA7B group.

	gpt3_trans	en-fr	fr-en	de-en	en-de	en-ro	ro-en	Avg
Dense Model	31.94	29.93	38.54	41.40	21.76	19.99	38.04	31.66
Sparse Model (wo LoRA)	19.26	7.83	31.72	32.05	7.10	7.07	28.96	19.14
Sparse Model (w LoRA)	22.14	15.26	32.58	32.31	11.16	8.03	29.64	21.59
Stand KD	21.88	17.48	32.23	32.40	11.98	7.00	28.43	21.63
BayesKD (SABO)	**24.59**	**19.54**	**34.16**	**35.45**	**13.92**	**10.21**	**31.84**	**24.25**

Table 12: Translation tasks on LLaMA3 8B and LLaMA3 70B models.

	gpt3_translation	en-fr	fr-en	de-en	en-de	en-ro	ro-en	Avg
Dense Model	12.62	8.55	18.44	20.92	11.40	8.35	27.86	15.45
Sparse Model (wo LoRA)	7.62	2.24	16.51	16.82	6.68	4.17	20.34	10.24
Sparse Model (w LoRA)	8.75	4.37	17.09	18.24	5.85	4.21	21.54	11.44
Stand KD	8.65	5.00	16.90	17.00	6.28	3.67	20.63	11.16
BayesKD (SABO)	**9.72**	**9.59**	**17.91**	**19.59**	**8.30**	**5.36**	**23.12**	**13.37**

Table 13: Translation tasks on Qwen2 7B and Qwen2 72B models.

	arc_challenge	arc_easy	boolq	hellaswag	openbookqa	piqa	Avg	wikitext
25%-Logits Dual-Scaling	45.56	71.13	76.24	74.32	41.80	78.29	**64.56**	**15.77**
25%-Single-Temperature	40.86	68.76	66.43	71.79	38.20	75.18	60.20	18.18
Dense-Logits Dual-Scaling	56.81	80.39	84.00	82.72	46.18	84.19	**72.38**	**8.84**
Dense-Single-Temperature	54.35	78.59	82.00	80.67	45.70	82.37	70.61	10.38
Original Dense Model	53.16	77.86	81.16	79.11	45.00	80.74	69.50	10.18

Table 14: Comparison of different sparsity rates and temperature/scaling value strategies on LLaMA3 models.

Stand KD's 11.16.

BayesKD (SABO) vs. Sparse Model (w LoRA) in Translation tasks (Table 12 and Table 13): Similarly, BayesKD (SABO) demonstrates clear advantages over Sparse Model (w LoRA) on both LLaMA3 and Qwen2 across all translation tasks. Specifically, in Table 12, BayesKD (SABO) achieves 24.25 compared to 21.59 for Sparse Model (w LoRA), and in Table 13, it achieves 13.37, surpassing the 11.44 scored by Sparse Model (w LoRA).

Comparison of different sparsity rates and temperature/scaling value strategies (Table 14): In general tasks, BayesKD (SABO) with logits dual-scaling strategies, as shown in Table 14, outperforms single-temperature configurations. For example, the logits dual-scaling strategy achieves an average score of 64.56 for 25% sparsity, compared to 60.20 for the single-temperature approach. This demonstrates the robustness of the logits dual-scaling configuration in resource-constrained settings.