LLM-Oriented Token-Adaptive Knowledge Distillation

Xurong Xie[1] *Zhucun Xue*[1] *Jiafu Wu*[2] *Jian Li*[2] *Yabiao Wang*[2]
Xiaobin Hu[2] *Yong Liu*[1] *Jiangning Zhang*[1,2]

[1]*Zhejiang University* [2]*Tencent Youtu Lab*

Knowledge Distillation (KD) is a key technique for compressing Large-scale Language Models (LLMs), yet prevailing logit-based methods typically employ static strategies that are misaligned with the dynamic learning process of student models. These methods typically treat all tokens indiscriminately and apply a single, fixed temperature, resulting in suboptimal knowledge transfer. To address these limitations, we propose LLM-oriented token-**Ada**ptive **K**nowledge **D**istillation (**AdaKD**), a novel framework that adapts the distillation process to the real-time learning state of each token. AdaKD consists of two synergistic modules driven by a unified token difficulty metric. First, our Loss-driven Adaptive Token Focusing (LATF) module dynamically adjusts the distillation focus by monitoring the student's learning stability, concentrating computational resources on the most valuable tokens at each training phase. Second, we introduce Inverse Difficulty Temperature Scaling (IDTS), a counterintuitive yet effective token-level temperature strategy. It employs low temperatures for difficult tokens for targeted error correction, and high temperatures for easy tokens to encourage the student to learn from the teacher's complete and smooth output distribution, thereby enhancing generalization. As a plug-and-play framework, AdaKD can consistently improve the performance of various distillation methods on multiple model architectures and benchmarks.

Date: October 13, 2025
Code: https://github.com/SassyRong/AdaKD

1 Introduction

Large Language Models (LLMs) have made significant advancements in recent years. They perform excellently on many natural language processing tasks, such as text generation, comprehension, and reasoning [1, 2, 3]. This success is mainly due to their *extensive parameter sizes* and the pre-training they undergo on *vast amounts of data* [4]. However, this powerful capability comes at the cost of *enormous computational and storage resources*. These requirements create significant barriers to ***deployment on edge devices in low-latency scenarios and to achieving widespread accessibility***, limiting the practical reach of LLMs [5, 6, 7].

To solve above challenges, Knowledge Distillation (KD) has emerged as a promising solution for model compression and acceleration. Our work focuses on logit-based distillation, a prevalent white-box approach that directly transfers knowledge by matching the output distributions of the teacher and student models. While conceptually simple and effective, we argue that current logit-based methods still face two key limitations in adapting to the dynamic learning process of the student model:

1) **Indiscriminate token treatment.** Most methods treat all tokens indiscriminately, applying a uniform distillation objective across the entire sequence. This lack of differentiation is misaligned with the student's real-time learning progress, resulting in suboptimal knowledge transfer and potentially introducing noise from tokens that are already well-mastered.

To better understand the consequences of this uniform treatment, we first investigate the learning dynamics of an instruction-following task at the token level (Fig. 1). As shown in the Fig. 1a, the difficulty of tokens for the student

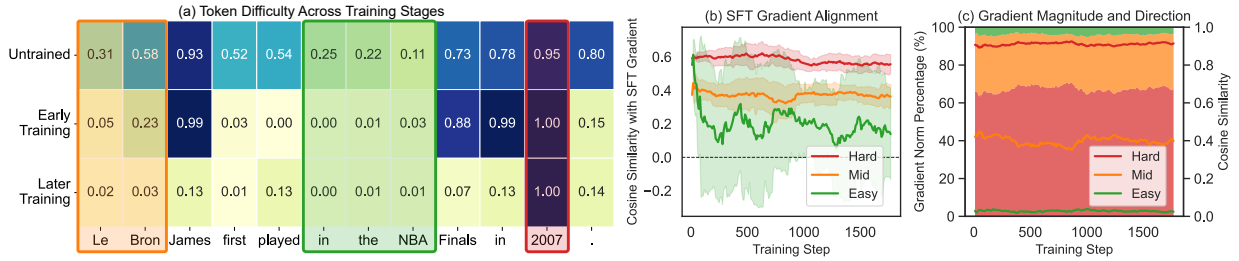

Figure 1. Analysis of token difficulty and gradient dynamics. Tokens are grouped into *Hard*, *Mid*, and *Easy* based on difficulty (Hellinger distance). **(a)** Evolution of token difficulty across training stages. **(b)** Cosine similarity of each token group's gradient with the SFT gradient. **(c)** Each group's gradient norm percentage and its cosine similarity with the total batch gradient.

model is not static but evolves throughout the training process. Some tokens are persistently challenging (*e.g.*, the token 2007, highlighted in the red box), requiring continuous focus. Others see their difficulty change dynamically (*e.g.*, Le and Bron, in the orange box), while many "easy" tokens are quickly mastered in the early training stages (*e.g.*, NBA and in, in the green box). This complex dynamic suggests that a static approach is suboptimal and motivates a token-wise, adaptive strategy.

Furthermore, we question the utility of continuing to train on "easy" tokens. We categorize tokens into "hard", "mid", and "easy" groups based on their difficulty and analyze their gradients. As shown in Fig. 1c, easy tokens contribute negligibly to the parameter update, with their gradient magnitude being very small and their direction being nearly orthogonal to the overall batch gradient. More critically, Fig. 1b reveals that the gradients of these easy tokens are unstable and poorly aligned with the supervised fine-tuning (SFT) direction, sometimes even moving in the opposite direction (negative cosine similarity). This evidence suggests that easy tokens provide limited learning value post-initial learning and may hinder knowledge transfer efficiency and stability via small, unstable gradients introducing conflicting signals.

To address above two limitations, we propose a novel Token-**Ada**ptive **K**nowledge **D**istillation (AdaKD) framework, which introduces a unified token difficulty metric driving two adaptive modules: ***1)*** Loss-driven Adaptive Token Focusing (LATF) module dynamically selects the most valuable tokens for training at each stage, ***2)*** while the Inverse Difficulty Temperature Scaling (IDTS) module taps temperature scaling's potential by assigning individual temperatures to tokens according to their learning difficulty.

In summary, our contributions are threefold.

- We introduce a novel adaptive token selection mechanism that improves distillation efficiency by dynamically adjusting its focus based on the student model's learning stability.
- A novel token-level temperature scaling strategy that inversely correlates temperature with token difficulty to achieve both targeted error correction and enhanced generalization.
- Extensive empirical validation of AdaKD as a versatile and plug-and-play enhancement that consistently improves a variety of distillation baselines and architectures.

2 Related Work

2.1 Knowledge Distillation for LLMs.

Knowledge Distillation (KD) transfers knowledge from a large teacher to a smaller student. Methods are broadly divided into black-box and white-box distillation. Black-box approaches [8, 9, 10] use only the teacher's final outputs, making them suitable for closed-source models [1, 11, 12] with less practical utility. Our work is in white-box distillation, which accesses teacher internals. Within this setting, feature-based distillation aligns intermediate hidden states, but this often requires complex, architecture-specific layer matching [13, 14, 15]. Logit-based distillation, in contrast, offers a simpler approach by matching the final output distributions using a divergence measure. Beyond the foundational Forward KL Divergence (FKD)[16] and Reverse KL Divergence (RKD)[17], much recent work has focused on developing more advanced objective functions [18, 19, 20]. Our approach is *a plug-and-play framework that can be flexibly combined with these different objective functions*.

2.2 Selective Token Distillation.

Traditional KD treats all tokens equally, though not all tokens are equally informative [21]. Consequently, many selective strategies have been proposed. One major direction is to focus the distillation loss on a subset of "important" tokens, identified based on metrics like difficulty or contribution to the teacher's prediction. For example, Selective Knowledge Distillation [22] uses a fixed ratio of tokens selected via a cross-entropy metric, which ignores the teacher's full distribution. A more advanced approach, AdaDS [23], dynamically adapts the difficulty metric (e.g., cross-entropy, confidence) using a lightweight RL selector. However, this strategy still relies on a pre-defined, fixed data selection ratio. Another line of work operates at the vocabulary level, for instance by preserving the relative order of top predictions [24, 25] or distilling only the top-k logits for efficiency [26, 27]. A common limitation in these approaches is the reliance on static or scheduled criteria. In contrast, *our framework's LATF module dynamically adapts the token selection ratio itself based on the evolving training loss*, avoiding the static criteria of prior work.

2.3 Adaptive Temperature Scaling.

The distillation temperature is a key hyperparameter that modulates knowledge transfer by smoothing logits. Most methods employ a fixed temperature, which struggles to adapt to the student's evolving learning state. Consequently, dynamic temperature scaling has been well-explored, particularly in computer vision. Some strategies involve using different temperatures to normalize teacher and student logits [28, 29], while others adopt curriculum-based approaches that adjust the temperature to create an easy-to-difficult learning path [30]. However, such adaptive strategies are less common in LLM distillation. A representative work is Annealing KD [31], which lowers the temperature according to a predefined schedule. Such scheduled approaches are not adaptive to the model's real-time needs. Closer to our work in spirit, methods such as ATD [32] and Mkdat [33] also adapt temperature according to the hardness of the sample. They typically rely on the cross-entropy loss to judge hardness and then use distinct functions to map this score to a temperature value. In contrast, *our novel token-level IDTS module derives temperatures from a direct teacher-student discrepancy via a unique inverse scaling strategy*.

3 Methodology

3.1 Preliminary of Knowledge Distillation in LLM

Inference in Large Language Models (LLMs) is a sequential vocabulary classification task. Given a pair of prompt and target response, denoted as $(\mathbf{x}, \mathbf{y})$, where $\mathbf{y} = (y_1, \ldots, y_L)$ is the target output sequence of length L, LLMs aim to

predict the conditional probability distribution $p(\cdot|\mathbf{x}, y_{<i})$ over the vocabulary $\mathcal{V}$ for each token $y_i \sim p(\cdot|\mathbf{x}, y_{<i})$. KD minimizes the difference between the distributions predicted by the teacher p and the student q_θ (parameterized by θ). These distributions are obtained by applying a softmax function to the model output logits z, scaled by a distillation temperature τ: $P(\cdot|\mathbf{x}, y_{<i}; \tau) = \text{softmax}(z_P(\cdot|\mathbf{x}, y_{<i})/\tau)$, where $P \in \{p, q_\theta\}$. The distillation loss is typically the average of token-level divergences computed using these temperature-scaled distributions, for each token y_i in the ground-truth response $\mathbf{y}$. Thus, the classic FKD distillation loss [16] is defined as:

$$\mathcal{L}_{\text{FKD}} = \frac{1}{L} \sum_{i=1}^{L} D_{KL}(p(\cdot|\mathbf{x}, y_{<i}; \tau) \parallel q_\theta(\cdot|\mathbf{x}, y_{<i}; \tau)). \tag{1}$$

The KL divergence is computed over the vocabulary $\mathcal{V}$. Notably, the inclusion of temperature τ in the softmax function leads to a τ^2 scaling factor in the final loss computation:

$$D_{KL}(p \parallel q_\theta) = \tau^2 \sum_{y_i \in \mathcal{V}} p(y_i|\mathbf{x}, y_{<i}; \tau) \log \frac{p(y_i|\mathbf{x}, y_{<i}; \tau)}{q_\theta(y_i|\mathbf{x}, y_{<i}; \tau)}. \tag{2}$$

Conversely, RKD loss [17] swaps the order of the distributions in the KL divergence, focusing on matching the modes of the teacher's distribution. These divergence measures form the basis of the distillation loss.

3.2 AdaKD: Token-Adaptive Knowledge Distillation

Building upon the insights from our analysis of token-level learning dynamics (Fig. 1), we introduce Token-Adaptive Knowledge Distillation (AdaKD). Instead of a static approach, AdaKD is designed to dynamically tailor the distillation process—both its focus and intensity—to the real-time learning difficulty of each individual token. A detailed comparison of our framework with other relative methods is deferred to the appendix.

The entire AdaKD procedure is described in Fig. 2. Our framework is driven by two synergistic modules: Loss-driven Adaptive Token Focusing (LATF), which selects the most valuable tokens for training at each phase, and Inverse Difficulty Temperature Scaling (IDTS), which assigns a tailored temperature to each selected token. The foundation for both modules is a robust token difficulty indicator, which we will describe first.

3.3 Choice of Difficulty Indicator

The effectiveness of AdaKD depends on a metric that accurately quantifies token-level learning difficulty. We define this difficulty using the Hellinger distance [34], which measures the divergence between the teacher's and student's output probability distributions. For the i-th output token y_i, its difficulty score s_i is calculated as:

$$s_i = \frac{1}{\sqrt{2}} \sqrt{\sum_{y_i \in \mathcal{V}} \left(\sqrt{p(y_i|\mathbf{x}, y_{<i})} - \sqrt{q_\theta(y_i|\mathbf{x}, y_{<i})} \right)^2}. \tag{3}$$

The resulting score s_i is bounded within the range of $[0, 1]$. This indicator is chosen for its advantageous properties. First, its symmetry provides an unbiased measure of discrepancy, avoiding the inherent mode- or mean-seeking tendencies of asymmetric metrics like FKD and RKD. Second, its square-root operation compares the entire output distributions and is particularly sensitive to disagreements on low-probability candidates, thus providing a more comprehensive difficulty signal that captures subtle deviations in the student's replication of the teacher's full output distribution. *This difficulty indicator* $\mathbf{s} = (s_1, \ldots, s_L)$ *then serves as the sole driving signal to synergistically guide the following two innovative modules that we designed.*

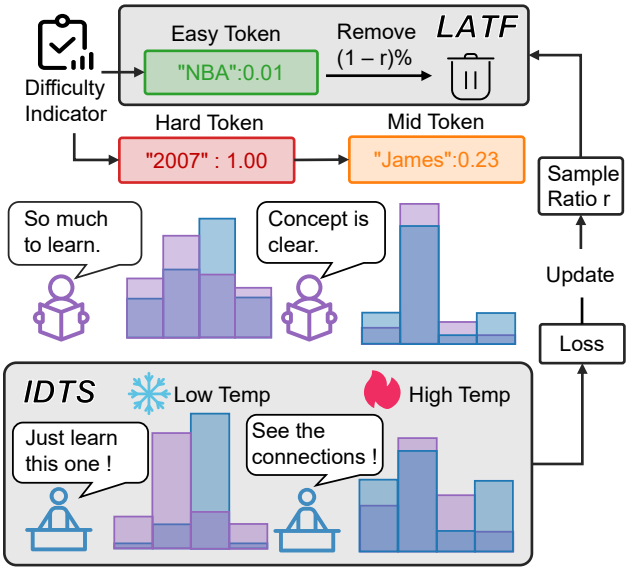

Figure 2. Illustration of the AdaKD framework. The bar charts visualize simplified teacher (blue) and student (purple) probability distributions. The top charts depict the initial learning gaps for "hard" and "mid-difficulty" tokens. After the LATF module filters tokens based on difficulty calculated via indicator, the IDTS module (bottom) applies low temperature to hard tokens for a sharp, corrective signal, and high temperature to easier tokens for a smoother distribution that enhances generalization.

Algorithm 1 Training Procedure of AdaKD.

Input: Teacher p, student q_{θ_0}, dataset $\mathcal{D}$, total iterations T, temperature scale c, EMA decay rate β, tolerance ϵ, step size δ, warm-up steps T_{warmup}
Output: Trained student model q_{θ_T}.
Initialize $t = 1$, sample ratio $r_0 = 1.0$, compute initial loss $\bar{\mathcal{L}}_0$ with q_{θ_0}, $\mathcal{L}_{\text{ref}} = \infty$.
while $t < T$ **do**
  Sample batch $(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}$; Compute logits $\mathbf{z}_p, \mathbf{z}_{q_{\theta_t}}$
  Compute per-token difficulty scores $\mathbf{s}$ using Eq. 3
  Update focusing ratio r_t using Eq. 7
  if $t > T_{\text{warmup}}$ and $r_t \neq r_{t-1}$ **then**
    $\mathcal{L}_{\text{ref}} \leftarrow \bar{\mathcal{L}}_{t-1}$.
  end if
  Compute per-token temperatures τ using Eq. 10
  $p \leftarrow \text{softmax}(\mathbf{z}_p / \tau)$
  $q_\theta \leftarrow \text{softmax}(\mathbf{z}_{q_{\theta_t}} / \tau)$.
  Compute $\mathcal{L}_{\text{AdaKD}}$ using Eq. 4
  Update $\theta : \theta_t \leftarrow \theta_{t-1} - \eta \cdot \nabla_{\theta_t} \mathcal{L}_{\text{AdaKD}}$
  $\bar{\mathcal{L}}_t = \beta \cdot \bar{\mathcal{L}}_{t-1} + (1 - \beta) \cdot \mathcal{L}_{\text{AdaKD}}$
  $t \leftarrow t + 1$
end while

3.4 Loss-driven Adaptive Token Focusing (LATF)

The gradient analysis in Fig. 1b and Fig. 1c reveals that training on "easy" tokens becomes inefficient and potentially unstable as training progresses. This strongly suggests that selectively focusing the distillation loss on a more valuable subset of tokens is beneficial. We implement this by applying the loss only to the top-r% of tokens with the highest difficulty scores:

$$\mathcal{L}_{\text{distill}} = \frac{1}{L * r\%} \sum_{i=1}^{L} I_{r\%}(y_i) \cdot D_{KL}(q_\theta \parallel p), \tag{4}$$

where $L * r\%$ is the number of tokens that fall within the top-r% of the difficulty metric. The indicator function $I_{r\%}(y_i)$ is defined as:

$$I_{r\%}(y_i) = \begin{cases} 1 & \text{if } s_i \text{ ranks in the top } r\% \text{ of } \mathbf{s} \\ 0 & \text{otherwise} \end{cases}. \tag{5}$$

However, using a fixed sample ratio r is suboptimal. A static ratio cannot adapt to the model's changing learning state. To address this, we introduce LATF to adjust the focusing ratio r_t dynamically. LATF operates via a simple feedback loop that monitors learning stability through the distillation loss. To obtain a stable signal, we first compute the exponential moving average (EMA) of the loss, denoted as $\bar{\mathcal{L}}_t$:

$$\bar{\mathcal{L}}_t = \beta \cdot \bar{\mathcal{L}}_{t-1} + (1 - \beta) \cdot \mathcal{L}_{\text{distill},t}, \tag{6}$$

where β is the decay rate of EMA and $\bar{\mathcal{L}}_0$ is the distillation loss when the student model is untrained. After an warm-up phase (where $r_t = 1.0$), we set a loss reference point $\mathcal{L}_{\text{ref}}$, which is initialized with the current EMA loss $\bar{\mathcal{L}}_t$. At each subsequent training step, LATF dynamically adjusts r_t by comparing the latest $\bar{\mathcal{L}}_t$ to $\mathcal{L}_{\text{ref}}$ within a tolerance ϵ.

Specifically, the update rules be described as:

$$r_t = \begin{cases} r_{t-1} \cdot (1-\delta) & \text{if } \bar{\mathcal{L}}_{t-1} < \mathcal{L}_{\text{ref}} \cdot (1-\epsilon) \\ \min(1.0, r_{t-1} \cdot (1+\delta)) & \text{if } \bar{\mathcal{L}}_{t-1} > \mathcal{L}_{\text{ref}} \cdot (1+\epsilon) \\ r_{t-1} & \text{otherwise,} \end{cases} \tag{7}$$

where δ is a small step size that controls the magnitude of adjustment. This rule creates an intuitive feedback loop. We decrease the selection ratio r_t to focus on more challenging tokens when the learning state is stable ($\bar{\mathcal{L}}_t$ drops below the lower bound). Conversely, we increase r_t to incorporate simpler tokens for stabilization when the model struggles. The ratio remains unchanged within the tolerance zone to prevent over-reaction to normal training oscillations. After any adjustment to r_t, the reference point $\mathcal{L}_{\text{ref}}$ is reset to the current $\bar{\mathcal{L}}_t$, keeping the performance baseline adaptive.

3.5 Inverse Difficulty Temperature Scaling (IDTS)

Once LATF selects the tokens, IDTS determines the optimal temperature for distilling each one. Contrary to the conventional approach of using a high temperature to soften the teacher's distribution [31], we propose an inverse strategy: applying low temperatures to difficult tokens and high temperatures to easier ones.

Consider the information entropy [35] of a probability distribution $\mathbf{p} = (p_1, \cdots, p_\mathcal{V})$, defined as $H(\mathbf{p}) = -\sum_i p_i \ln(p_i)$, which quantifies the uncertainty of the distribution. The relationship between entropy and temperature can be precisely described by the derivative:

$$dH/d\tau = \text{Var}_{p(\tau)}(z)/\tau^3, \tag{8}$$

where $\text{Var}_{p(\tau)}(z)$ denotes the variance of logit z under the distribution p generated with temperature τ. As variance is non-negative and $\tau > 0$, the derivative in Equation (8) is always non-negative, indicating that entropy is a monotonically increasing function of temperature.

Our IDTS module leverages this mathematical principle. For difficult tokens (high s_i), a low τ_i reduces the entropy, simplifying the learning objective into a sharp, corrective signal that focuses the student on matching the teacher's single best prediction. For easy tokens (low s_i), a high τ_i increases entropy, changing the objective to be more extractive . This encourages the student to learn the broader shape of the teacher's distribution, thereby enhancing generalization.

The implementation begins by converting the raw difficulty score s_i into a normalized learning state $\hat{s}_i \in [-1, 1]$. This process is designed for robustness and stability: we first compute the ratio of s_i to the batch median, chosen for its robustness to outliers. We then apply a log function to compress the long-tail distribution of these ratios, followed by a tanh function to smoothly map the result into the bounded range:

$$\hat{s}_i = \tanh\left(\log\left(s_i / \text{median}(\mathbf{s})\right)\right). \tag{9}$$

Subsequently, this learning state $\hat{s}_i$ dynamically modulates a base temperature τ_{base} via an exponential function:

$$\tau_i = \tau_{\text{base}} \cdot \exp(-c \cdot \hat{s}_i.\text{detach}()). \tag{10}$$

Here, the negative sign enacts our inverse difficulty principle, with the hyperparameter c controlling the modulation intensity. We chose this multiplicative approach because it makes the scaling effect robust to the specific value of τ_{base} and naturally constrains the final temperature τ_i to the predictable range of $[\tau_{base} \cdot e^{-c}, \tau_{base} \cdot e^{c}]$. The entire calculation is detached from the computation graph, treating the resulting temperatures as fixed supervisory signals. The full AdaKD procedure, combining LATF and IDTS, is detailed in Algorithm 1.

3.6 Gradient Analysis of IDTS

The loss function activates only high-difficulty tokens (where $I_{r\%}(y_i) = 1$). For these tokens, we compute the temperature τ_i scaling gradient of the KL divergence $D_{\mathrm{KL}}(q_\theta \parallel p)$ with respect to student logits z_q:

$$\frac{\partial D_{\mathrm{KL}}^{(\tau_i)}}{\partial z_q(y_j)} = \frac{1}{\tau_i}\left(q_\theta^{(\tau_i)}(y_j) - p^{(\tau_i)}(y_j)\right). \tag{11}$$

The update magnitude is governed by the gradient norm:

$$\left\|\nabla D_{\mathrm{KL}}^{(\tau_i)}\right\|^2 = \sum_{y_j \in \mathcal{V}} \left(\frac{\partial D_{\mathrm{KL}}^{(\tau_i)}}{\partial z_q(y_j)}\right)^2 = \frac{1}{\tau_i^2}\left\|q_\theta^{(\tau_i)} - p^{(\tau_i)}\right\|_2^2. \tag{12}$$

To minimize $\mathcal{L}_{\text{distill}}$, we need to maximize this gradient magnitude for accelerated convergence:

$$\min \mathcal{L}_{\text{distill}} \implies \max \sum_{\substack{i \\ I_{r\%}(y_i)=1}} \frac{1}{\tau_i^2}\left\|q_\theta^{(\tau_i)} - p^{(\tau_i)}\right\|_2^2. \tag{13}$$

The difficulty metric s_i is defined as the Hellinger distance:

$$s_i = \frac{1}{\sqrt{2}}\left\|\sqrt{p} - \sqrt{q_\theta}\right\|_2 \implies \left\|\sqrt{p} - \sqrt{q_\theta}\right\|_2^2 = 2s_i^2. \tag{14}$$

Temperature scaling modifies the distribution discrepancy:

$$\left\|q_\theta^{(\tau_i)} - p^{(\tau_i)}\right\|_2^2 \propto \frac{1}{\tau_i^2}\left\|q_\theta - p\right\|_2^2. \tag{15}$$

Combining these relationships yields:

$$\left\|q_\theta^{(\tau_i)} - p^{(\tau_i)}\right\|_2^2 \propto \frac{s_i^2}{\tau_i^2}, \quad \left\|\nabla D_{\mathrm{KL}}^{(\tau_i)}\right\|^2 \propto \frac{1}{\tau_i^2}\cdot\frac{s_i^2}{\tau_i^2} = \frac{s_i^2}{\tau_i^4}. \tag{16}$$

Thus, the KL loss gradient is inversely related to the temperature τ. For difficult tokens, there is a larger discrepancy between the output distributions of the student and the teacher, student model require a larger gradient to approximate the teacher's distribution, which corresponds to a lower temperature. For easy tokens, the output distributions of the student and teacher are more similar, the student model need a smaller gradient to prevent itself from diverging, which corresponds to a higher temperature.

4 Experimental Results

4.1 Experimental Setups

Datasets and Models. Following the widely-adopted setup from Gu et al. [17], Ko et al. [19], we use the databricks-dolly-15k dataset [36] for training and evaluate on five instruction-following benchmarks: Dolly-eval, Self-Instruct [37], Vicuna-eval [38], Super-Natural Instructions (S-NI) [39], and Unnatural Instructions [40]. We demonstrate the generalizability of our framework on two modern model families: Qwen2-7B distilled to Qwen2-1.5B and OpenLLaMA2-7B to OpenLLaMA2-3B.

Baselines and Implementation Details. We compare AdaKD with supervised fine-tuning (SFT) and state-of-the-art KD methods, including FKD, RKD [17], ABKD [20], GKD [41], and DistiLLM [19]. For a fair comparison, all baselines are reproduced using their official implementations and meticulously tuned.

Our experiments were conducted on a setup of 4 or 8 NVIDIA H20 80GB GPUs. For all distillation methods, we perform a hyperparameter search for the learning rate within the range of {1e-4, 5e-4, 1e-5, 5e-5} and for the global batch size within {16, 32, 64, 128}. Following the search, we set the learning rate to 5e-4 and the global batch size to 128 for all main experiments to ensure consistency. For the Qwen2 model, we use a batch size of 64 with a gradient accumulation factor of 2.

Following standard practice, we train the smaller GPT-2 model for 20 epochs, while the larger Qwen2 and OpenLLaMA2 models are trained for 10 epochs. For the Qwen2 and OpenLLaMA2 models, we employ Low-Rank Adaptation (LoRA) with a rank of 16 for parameter-efficient distillation. The final model checkpoints for evaluation are selected based on the highest ROUGE-L scores on the validation set.

Evaluation. We report the **ROUGE-L** [45] score to measure the quality of generated text. Following standard practice, we generate responses from all models with the decoding temperature and top-p both set to 1.0. To ensure statistical robustness, we use five different random seeds: {10, 20, 30, 40, 50} and report the averaged ROUGE-L scores.

We also employ a powerful large language model, Qwen3-32B, as an impartial judge to perform pairwise comparisons between the responses generated by a baseline method and its enhanced version with AdaKD. To mitigate position bias, the order of the two responses is swapped in a second evaluation round, and only consistent judgments are retained. The results, presented as win/tie/loss percentages for AdaKD, are reported on the Dolly-eval, Self-Instruct, and UnNI benchmarks.

4.2 Quantitative Results

Table 1 validates AdaKD as a universal plug-and-play enhancement. While advanced objectives (*e.g.*, RKD, ABKD) already outperform foundational methods and can even surpass complex Student-Generated Outputs (SGOs) based approaches(*e.g.*, GKD), AdaKD consistently elevates all of them to new state-of-the-art performance. This universal improvement demonstrates that dynamically adapting the distillation process to the student's real-time learning state is a robust and crucial element for effective knowledge transfer, offering a fundamental enhancement regardless of the underlying distillation objective.

Table 2 presents the results of the LLM-as-a-Judge evaluation. The findings reveal that the effectiveness of AdaKD varies between methods. Techniques like GKD and DistiLLM, which introduce SGOs that bring new instability, benefit significantly, as our LATF component filters the resulting noisy gradients for a more stable performance gain. In contrast, the improvement for FKD is marginal. This is attributed to a conceptual conflict: FKD's mean-seeking objective clashes with our IDTS's targeted correction for hard tokens.

4.3 Ablation Studies and Analyses

We conduct ablation studies on the Qwen2-7B $\rightarrow$ Qwen2-1.5B distillation task using RKD as the baseline to dissect the contribution of each component in AdaKD. For clarity, the following tables detail results on Dolly, S-NI, and UnNI, the three benchmarks with the most extensive test items.

Impact of Core Components in AdaKD. Tab. 3a reveals the synergy between our components. While integrating IDTS alone brings a substantial performance boost, LATF alone yields no improvement. This is consistent with our

Table 1. Comparison of ROUGE-L scores for various KD methods on five instruction-following benchmarks. All experiments were conducted using five different random seeds, with results reported as 'mean ± standard deviation'. For each student model configuration, optimal and sub-optimal results are highlighted in **bold** and underline. 'w/ AdaKD' denotes our proposed plug-and-play enhancement, which **consistently improves performance across different base models**.

Model	Parameters	Method	Dolly	Self-Inst	Vicuna Eval	S-NI	UnNI	Avg.	
Qwen2 [42]	7B	Teacher	29.29±0.56	24.01±0.63	20.18±0.72	40.74±0.63	37.21±0.76	30.29	
	1.5B	SFT (LoRA)	24.82±0.30	18.79±0.55	17.99±0.65	32.13±0.93	31.04±0.60	24.95	
		FKD	25.72±0.85	20.13±0.97	18.24±0.30	35.77±0.37	32.93±1.21	26.56	
		w/ AdaKD	25.94±0.31	19.75±0.24	18.36±0.18	35.88±0.53	33.21±0.51	26.63	(↑0.07)
		RKD	29.52±0.50	**24.92**±0.66	22.50±0.51	41.68±0.67	39.90±0.49	31.70	
		w/ AdaKD	30.03±0.40	24.88±0.71	22.97±0.58	43.82±0.78	**43.17**±0.32	**32.97**	(↑1.27)
		ABKD	29.43±0.60	23.45±0.63	22.72±0.60	41.60±0.79	40.34±0.47	31.51	
		w/ AdaKD	**30.44**±0.50	23.60±0.75	**23.40**±0.77	**44.23**±1.39	42.54±0.78	32.84	(↑1.33)
		GKD	27.13±0.47	20.89±0.90	19.41±0.39	38.25±0.84	35.01±0.59	28.14	
		w/ AdaKD	27.98±0.58	23.00±0.78	19.62±0.17	40.31±0.97	37.77±0.71	29.74	(↑1.60)
		Distillm	29.10±0.51	22.92±0.64	21.79±0.44	41.26±0.46	38.80±0.73	30.77	
		w/ AdaKD	29.69±0.40	23.55±0.96	22.11±0.55	42.91±0.80	40.73±0.82	31.80	(↑1.03)
OpenLLaMA2 [43]	7B	Teacher	28.16±0.60	20.40±0.92	17.62±0.48	30.45±0.82	33.18±0.47	25.96	
	3B	SFT (LoRA)	26.54±0.13	17.45±0.42	16.87±0.27	31.64±0.88	30.64±0.49	24.63	
		FKD	26.56±0.38	18.11±0.60	16.78±0.40	31.94±0.79	30.97±0.52	24.87	
		w/ AdaKD	26.96±0.58	18.75±0.55	16.64±0.47	32.78±0.92	31.64±0.65	25.35	(↑0.48)
		RKD	29.13±0.34	20.08±0.66	19.49±0.28	35.20±0.60	37.60±0.62	28.30	
		w/ AdaKD	29.81±0.35	20.00±0.55	19.49±0.37	36.80±1.13	**40.26**±0.54	29.27	(↑0.97)
		ABKD	29.45±0.77	20.96±0.76	**19.78**±0.26	35.98±0.74	38.60±0.63	28.95	
		w/ AdaKD	**30.19**±0.50	20.65±0.32	19.55±0.28	36.38±0.30	39.82±0.56	29.32	(↑0.37)
		GKD	29.23±0.41	19.96±0.80	18.10±0.75	34.68±0.58	35.05±0.63	27.40	
		w/ AdaKD	29.48±0.15	20.96±0.56	19.07±0.32	**37.60**±0.43	39.31±0.27	29.28	(↑1.88)
		Distillm	29.50±0.56	20.67±0.86	19.09±0.44	35.58±0.66	37.39±1.13	28.45	
		w/ AdaKD	29.52±0.63	**22.13**±0.47	19.50±0.50	37.23±0.72	40.14±0.71	**29.70**	(↑1.25)
GPT-2 [44]	1.5B	Teacher	28.17±0.30	15.93±0.57	16.99±0.37	29.08±0.22	34.53±0.22	24.94	
	0.1B	SFT	24.08±0.42	10.39±0.53	15.39±0.16	19.21±0.73	23.72±0.40	18.56	
		FKD	23.90±0.65	10.30±0.15	14.93±0.32	19.07±0.22	23.81±0.33	18.40	
		w/ AdaKD	24.12±0.36	10.21±0.49	14.93±0.44	19.37±0.63	24.39±0.79	18.60	(↑0.20)
		RKD	26.05±0.26	12.24±0.11	15.89±0.38	25.24±0.69	31.05±0.49	22.11	
		w/ AdaKD	26.11±0.25	11.89±0.32	15.84±0.37	26.61±0.27	33.29±0.49	22.75	(↑0.64)
		ABKD	26.01±0.50	12.30±0.81	15.90±0.60	27.99±0.92	32.35±0.74	22.91	
		w/ AdaKD	26.00±0.20	12.48±0.43	15.08±0.41	29.50±0.33	35.12±0.27	23.64	(↑0.73)
		GKD	25.81±0.47	13.12±0.57	16.39±0.17	24.86±0.58	29.63±0.31	21.96	
		w/ AdaKD	**27.20**±0.25	13.34±0.36	**17.63**±0.30	28.29±0.54	33.92±0.72	24.08	(↑2.12)
		Distillm	26.81±0.14	12.56±0.68	16.58±0.20	26.28±0.71	31.63±0.82	22.77	
		w/ AdaKD	26.62±0.51	**13.51**±0.49	16.21±0.29	**29.54**±0.46	**35.43**±0.80	**24.26**	(↑1.49)

gradient analysis (Fig. 1): LATF's primary role is to stabilize training by filtering out mastered tokens with unstable gradients, rather than directly advancing performance. The full AdaKD model achieves the best results, confirming a crucial synergy: LATF first removes noise to stabilize the learning process, which then allows IDTS to more effectively apply its adaptive teaching strategy to the remaining high-value tokens.

Analysis of the Difficulty Indicator. We evaluated several distribution metrics as the difficulty indicator, with results presented in Tab. 3c. The results highlight that the optimal metric for an indicator differs from the distillation loss itself; for instance, FKL is a much more effective indicator than RKD; furthermore, symmetric metrics like Hellinger distance and JS-Divergence show a clear advantage over asymmetric ones on the large-scale S-NI and UnNI benchmarks. We also observe that Cross-Entropy, measured against the ground truth, performs best on the Dolly dataset, while

Table 2. LLM-as-a-Judge evaluation results for the Qwen2-1.5B distillation task. The table presents the win, tie, and loss rates (%) of models enhanced with AdaKD against their respective baselines.

	Dolly			S-NI			UnNI		
Method	Win%	Tie%	Loss%	Win%	Tie%	Loss%	Win%	Tie%	Loss%
FKD	18.80	59.00	22.20	19.42	54.25	26.33	16.96	65.14	17.90
RKD	19.80	65.80	14.40	23.55	59.68	16.77	15.83	72.09	12.07
ABKD	20.80	62.60	16.60	21.14	60.86	18.00	10.19	80.59	9.22
GKD	20.00	64.20	15.80	22.43	62.75	14.82	18.30	69.60	12.10
DistiLLM	24.40	55.00	20.60	25.15	56.14	18.71	18.26	68.47	13.27

Table 3. Comprehensive ablation studies of AdaKD (ROUGE-L scores). (a) Core components analysis. (b) Ablation on temperature scaling strategies. (c) Comparison of different difficulty indicators. (d) Evaluation of LATF designs.

(a) Core Components

Method	**Dolly**	**S-NI**	**UnNI**	**Avg.**
RKD (Baseline)	29.52	41.68	39.90	37.03
+ IDTS	**30.12**	43.70	41.83	38.55
+ LATF	29.50	41.88	39.82	37.07
AdaKD (Full)	30.03	**43.82**	**43.17**	**39.01**

(b) Temperature Scaling

Method	**Dolly**	**S-NI**	**UnNI**	**Avg.**
$T = 1.0$	29.50	41.88	39.82	37.07
AdaKD ($c = 0.5$)	30.03	**43.82**	**43.17**	**39.01**
Inv. Scaling ($-c$)	28.93	40.02	38.06	35.67
$T = 0.8$	30.00	41.41	40.10	37.17
$T = 1.2$	29.55	42.07	40.05	37.22
$T \approx e^{-0.5}$	30.19	42.87	41.38	38.15
CTKD	**30.22**	41.99	40.30	37.50
Logit Std.	26.74	40.08	37.46	34.76

(c) Difficulty Indicator

Method	**Dolly**	**S-NI**	**UnNI**	**Avg.**
RKD (Baseline)	29.52	41.68	39.90	37.03
AdaKD with different metrics:				
+ FKD	30.29	42.81	42.34	38.48
+ RKD	29.93	43.02	42.23	38.39
+ Cross-Entropy	**30.46**	43.46	42.63	38.85
+ JS-Divergence	30.15	43.39	42.58	38.71
+ NMTKD	30.27	43.61	42.64	38.84
+ Hellinger (Ours)	30.03	**43.82**	**43.17**	**39.01**

(d) LATF Design

Method	**Dolly**	**S-NI**	**UnNI**	**Avg.**
fixed $r = 1.0$	30.12	43.70	41.83	38.55
LATF	30.03	**43.82**	**43.17**	**39.01**
fixed $r = 0.75$	30.27	43.49	42.02	38.59
linear $r : 1.0 \to 0.75$	29.63	43.37	41.83	38.28
cosine $r : 1.0 \to 0.75$	**30.29**	43.61	42.70	38.87
cosine $r : 1.0 \to 0.5$	29.74	42.71	41.74	38.06

NMTKD [24], which focuses on aligning the top-k (k=5) predictions of each token, also demonstrates competitive performance. Ultimately, Hellinger distance achieves the highest average score, validating its use to provide the balanced and comprehensive disagreement signal crucial for our adaptive framework.

Analysis of LATF's Design. We compare LATF against static and scheduled strategies in Tab. 3d, using a target ratio $r = 0.75$ for a fair comparison based on LATF's observed final value. While these schedules prove competitive, LATF is ultimately more robust on challenging benchmarks. The reason is visualized in Fig. 3(a,b). For LATF (Fig. 3a), the sample ratio adapts to the training loss in real-time. The temporary increase in loss is an expected outcome of the model tackling a harder and more focused curriculum, a challenge it successfully overcomes. In contrast, scheduled methods (Fig. 3b) exhibit rising loss in later stages as they blindly enforce difficulty. This real-time adaptation makes LATF a more robust solution that eliminates the need for schedule-specific tuning.

The LATF module includes four hyperparameters. The most critical are the tolerance (ϵ) and step size (δ), which govern the feedback loop. As shown in Table 4, our grid search reveals stable performance when $\delta \leq \epsilon$, leading us to select $\epsilon = 0.05$ and $\delta = 0.05$ to balance responsiveness and stability. The other parameters, the EMA decay rate (β) and the warm-up ratio, were set to 0.97 and 0.05, respectively, to ensure a smooth training process.

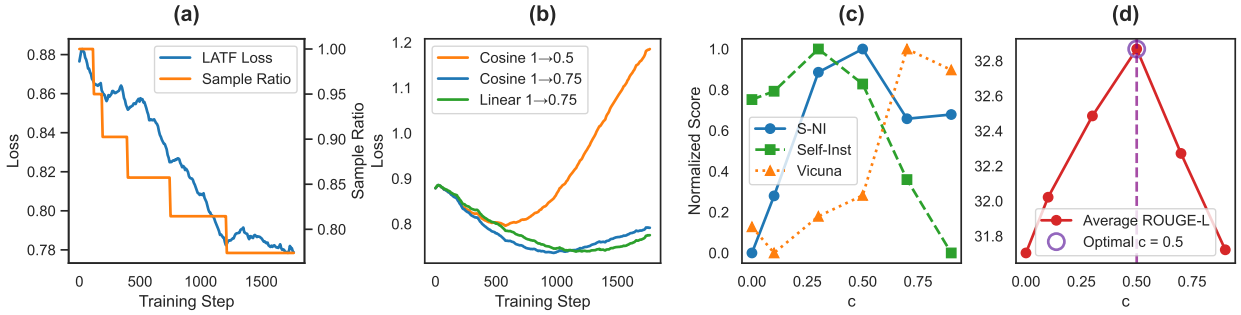

Figure 3. From left to right: (a) The loss and sample ratio of our adaptive LATF during training. (b) A comparison of loss curves for fixed scheduling strategies. (c) The effect of IDTS modulation intensity c on normalized scores for individual datasets. (d) The average ROUGE-L score across datasets, showing the optimal choice for c.

Table 4. LATF sensitivity analysis on UnNI benchmark.

Step (δ)	$\epsilon = 0.02$	$\epsilon = 0.05$	$\epsilon = 0.1$
0.02	42.74	**43.23**	42.93
0.05	42.44	43.17	43.06
0.1	41.73	42.12	42.94

Table 5. Training throughput (samples/sec) on Qwen2.

Method	**Baseline**	**w/ AdaKD**	**Change**
RKD	8.21	7.94	-3.29%
GKD	1.56	1.55	-0.87%
DistiLLM	3.88	3.82	-1.55%

Analysis of Key Design Choices within IDTS. Tab. 3b validates IDTS's design against various temperature strategies. The failure of "Inverse Scaling" confirms our hypothesis that low temperatures are crucial for difficult tokens. However, simply using a low temperature globally is insufficient; AdaKD surpasses not only fixed-temperature baselines but also one using our method's optimal lower bound ($T \approx e^{-0.5}$). This proves the dynamic, token-level application is the key to success. Furthermore, AdaKD's superior performance over other adaptive methods like CTKD [30] and Logit Std [29, 46] highlights the effectiveness of our specific token-level design.

We analyze the impact of the IDTS modulation intensity c in Fig. 3(c,d). While the optimal c varies for individual datasets (Fig. 3c), the average performance across all benchmarks (Fig. 3d) robustly peaks at $c = 0.5$. We therefore adopt this value for our main experiments.

Efficiency Comparison The additional computations in AdaKD for token difficulty and temperature have a negligible impact on training efficiency. This is because these steps are lightweight and detached from the computation graph, adding no overhead to backpropagation. Table 5 quantitatively validates this by comparing the training throughput on the Qwen2 model.

Analysis of the Dynamic Mechanisms in AdaKD. Fig. 4 illustrates the dynamic synergy of AdaKD's mechanisms by comparing distributions at the start and end of training. A key observation is that our IDTS module consistently aligns the information entropy of each part of tokens, regardless of the training stage. This dynamically adjusts the learning objective for each token, guiding the model's output distributions toward a uniform level of uncertainty, regardless of their initial difficulty.

The temperature distribution reflects this adaptive strategy: IDTS consistently assigns lower temperatures to hard tokens and higher temperatures to easy ones. However, the distribution's evolution, highlighted in the red circles, reveals the critical synergy with LATF. Early in training, the temperature for easy tokens peaks sharply, as the "easy" set contains many trivial examples. Conversely, late in training, LATF has removed these mastered tokens, causing IDTS to assign a smoother range of high temperatures to the remaining, non-trivial "easy" set. This demonstrates how LATF dynamically refines the learning process, enabling IDTS to apply its scaling more effectively on tokens that still offer learning value.

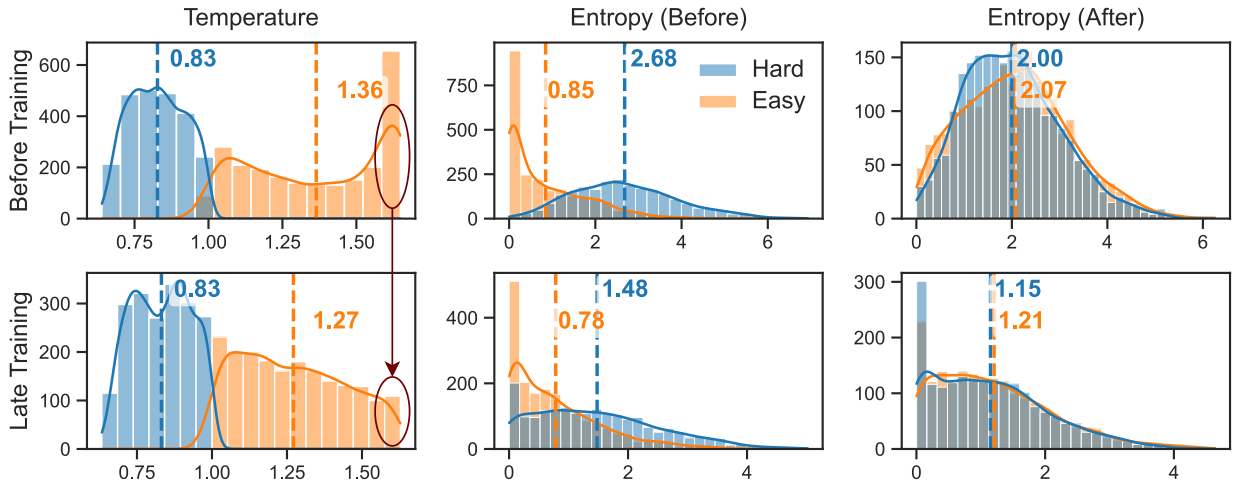

Figure 4. These histograms display the distribution of token counts (y-axis) across different metrics. The rows compare the model's state 'Before Training' (top) with 'Late in Training' (bottom). The columns, from left to right, show the distributions for assigned temperature, student's output entropy before IDTS, and entropy after IDTS. Tokens are categorized into 'hard' (blue) and 'easy' (orange) groups, with dashed vertical lines indicating their respective means.

5 Conclusion

In this paper, we introduced Token-Adaptive Knowledge Distillation (AdaKD), a novel framework that dynamically adapts the distillation process to each token's learning state, overcoming the limitations of static distillation strategies. AdaKD synergistically combines Loss-driven Adaptive Token Focusing (LATF) to concentrate on valuable tokens and Inverse Difficulty Temperature Scaling (IDTS) to apply a highly effective temperature strategy for both error correction and generalization. Extensive experiments demonstrate that AdaKD, as a plug-and-play enhancement, consistently improves the performance of various distillation methods across multiple architectures.

Limitation. Our work demonstrates the effectiveness of dynamically filtering tokens and dynamic temperature in distillation. Looking forward, there are clear paths for improvement.Currently, our Loss-driven Adaptive Token Focusing (LATF) module relies on a discrete adjustment mechanism, which can lead to slight training oscillations. Similarly, the formulation of our Inverse Difficulty Temperature Scaling (IDTS) module is based on heuristic principles and lacks a formal theoretical grounding. Future work can further explore the usage of continuous adjustment to obtain smoother convergence and more theoretical formulation like a direct control of the token-level information entropy for scaling. Lastly, our evaluation is limited to instruction-following tasks. Future work may expand the evaluation to challenging tasks including complex reasoning and code generation benchmarks.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

[3] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle,

Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[4] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[5] Zhongwei Wan, Xin Wang, Che Liu, Samiul Alam, Yu Zheng, Jiachen Liu, Zhongnan Qu, Shen Yan, Yi Zhu, Quanlu Zhang, et al. Efficient large language models: A survey. *arXiv preprint arXiv:2312.03863*, 2023.

[6] Yue Zheng, Yuhao Chen, Bin Qian, Xiufang Shi, Yuanchao Shu, and Jiming Chen. A review on edge large language models: Design, execution, and applications. *ACM Computing Surveys*, 57(8):1–35, 2025.

[7] Guangji Bai, Zheng Chai, Chen Ling, Shiyu Wang, Jiaying Lu, Nan Zhang, Tingwei Shi, Ziyang Yu, Mengdan Zhu, Yifei Zhang, et al. Beyond efficiency: A systematic survey of resource-efficient large language models. *arXiv preprint arXiv:2401.00625*, 2024.

[8] Ping Yu, Jing Xu, Jason Weston, and Ilia Kulikov. Distilling system 2 into system 1. *arXiv preprint arXiv:2407.06023*, 2024.

[9] Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. *arXiv preprint arXiv:2305.02301*, 2023.

[10] Namgyu Ho, Laura Schmid, and Se-Young Yun. Large language models are reasoning teachers. *arXiv preprint arXiv:2212.10071*, 2022.

[11] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[12] Anthropic. Introducing Claude. https://www.anthropic.com/news/introducing-claude, 2023.

[13] Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for bert model compression. *arXiv preprint arXiv:1908.09355*, 2019.

[14] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in neural information processing systems*, 33:5776–5788, 2020.

[15] Chen Liang, Simiao Zuo, Qingru Zhang, Pengcheng He, Weizhu Chen, and Tuo Zhao. Less is more: Task-aware layer-wise distillation for language model compression. In *International Conference on Machine Learning*, 2023.

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

[17] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *ICLR*, 2024.

[18] Taiqiang Wu, Chaofan Tao, Jiahao Wang, Runming Yang, Zhe Zhao, and Ngai Wong. Rethinking kullback-leibler divergence in knowledge distillation for large language models. *arXiv preprint arXiv:2404.02657*, 2024.

[19] 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*, 2024.

[20] Guanghui Wang, Zhiyong Yang, Zitai Wang, Shi Wang, Qianqian Xu, and Qingming Huang. Abkd: Pursuing a proper allocation of the probability mass in knowledge distillation via α-β-divergence. *arXiv preprint arXiv:2505.04560*, 2025.

[21] Steven T Piantadosi. Zipf's word frequency law in natural language: A critical review and future directions. *PBR*, 2014.

[22] Fusheng Wang, Jianhao Yan, Fandong Meng, and Jie Zhou. Selective knowledge distillation for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6456–6466, 2021.

[23] Qinhong Zhou, Peng Li, Yang Liu, Yuyang Guan, Qizhou Xing, Ming Chen, and Maosong Sun. Adads: Adaptive data selection for accelerating pre-trained language model knowledge distillation. *AI Open*, 4:56–63, 2023.

[24] Songming Zhang, Yunlong Liang, Shuaibo Wang, Wenjuan Han, Jian Liu, Jinan Xu, and Yufeng Chen. Towards understanding and improving knowledge distillation for neural machine translation. *arXiv preprint arXiv:2305.08096*, 2023.

[25] Tianyu Peng and Jiajun Zhang. Enhancing knowledge distillation of large language models through efficient multi-modal distribution alignment. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2478–2496, 2025.

[26] Mrigank Raman, Pranav Mani, Davis Liang, and Zachary Lipton. For distillation, tokens are not all you need. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*, 2023.

[27] Xiaoyu Liu, Yun Zhang, Wei Li, Simiao Li, Xudong Huang, Hanting Chen, Yehui Tang, Jie Hu, Zhiwei Xiong, and Yunhe Wang. Multi-granularity semantic revision for large language model distillation. *arXiv preprint arXiv:2407.10068*, 2024.

[28] Jia Guo. Reducing the teacher-student gap via adaptive temperatures. *arXiv preprint arXiv:2010.07485*, 2020.

[29] Zhihao Chi, Tu Zheng, Hengjia Li, Zheng Yang, Boxi Wu, Binbin Lin, and Deng Cai. Normkd: Normalized logits for knowledge distillation. *arXiv preprint arXiv:2308.00520*, 2023.

[30] Zheng Li, Xiang Li, Lingfeng Yang, Borui Zhao, Renjie Song, Lei Luo, Jun Li, and Jian Yang. Curriculum temperature for knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 1504–1512, 2023.

[31] Aref Jafari, Mehdi Rezagholizadeh, Pranav Sharma, and Ali Ghodsi. Annealing knowledge distillation. *arXiv preprint arXiv:2104.07163*, 2021.

[32] Shunzhi Yang, Xiong Yang, Jin Ren, Liuchi Xu, Jinfeng Yang, Zhenhua Huang, Zheng Gong, and Wenguang Wang. Adaptive temperature distillation method for mining hard samples' knowledge. *Neurocomputing*, 636: 129745, 2025.

[33] Jun Long, Zhuoying Yin, Yan Han, and Wenti Huang. Mkdat: Multi-level knowledge distillation with adaptive temperature for distantly supervised relation extraction. *Information*, 15(7):382, 2024.

[34] Ernst Hellinger. Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen. *Journal für die reine und angewandte Mathematik*, 1909(136):210–271, 1909.

[35] Claude E Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[36] Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world's first truly open instructiontuned llm. 2023.

[37] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*, 2022.

[38] Wei-Lin Chiang, Zhuohan Li, Ziqing Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6, 2023.

[39] Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. *arXiv preprint arXiv:2204.07705*, 2022.

[40] Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. Unnatural instructions: Tuning language models with (almost) no human labor. *arXiv preprint arXiv:2212.09689*, 2022.

[41] Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *ICLR*, 2024.

[42] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

[43] Xinyang Geng and Hao Liu. Openllama: An open reproduction of llama, 2023.

[44] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[45] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[46] Shangquan Sun, Wenqi Ren, Jingzhi Li, Rui Wang, and Xiaochun Cao. Logit standardization in knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15731–15740, 2024.