Tsinghua Science and Technology
Doi: 10.26599/TST.2026.9010077

Research Article

AKDF: Adaptive knowledge distillation for secure, efficient, copyright-protected model publishing

Feng Jiang, Honghui Xu, Daehee Seo, Yongjoon Joe, Wonbin Kim, and Zhipeng Cai(✉)

Abstract The rise of large language models (LLMs) has transformed natural language processing, powering applications from creative writing to code generation. However, their vast size and proprietary nature present two major challenges, including efficient deployment on limited hardware and secure protection of intellectual property. This work introduces the Adaptive Knowledge Distillation Framework (AKDF), a unified training approach that simultaneously compresses LLMs and embeds ownership signals for copyright assurance. AKDF employs parameter-efficient Low-Rank Adaptation (LoRA) to distill a 1-billion-parameter student model from an 8-billion-parameter teacher while integrating an output-level watermarking module directly into the distillation process. This design reduces trainable parameters and encodes verifiable ownership signatures without altering the frozen base weights. Experiments on ARC-Easy, PIQA, and WMT16 show that the student retains approximately 76% of the teacher's performance while maintaining competitive reasoning and translation quality under substantial compression. AKDF also enables secure and communication-efficient model publishing, providing a practical path toward bandwidth-efficient and ownership-aware deployment of large-scale language models.

Keywords Knowledge Distillation, Large Language Models, Copyright Protection, Watermarking, Low-Rank Adaptation

1 Introduction

The rapid evolution of large language models (LLMs) [1] has driven remarkable advances in natural-language understanding and generation. These models now power systems for complex question answering, dialogue, and code synthesis. State-of-the-art models such as GPT-4, LLaMA, and their derivatives contain billions of parameters trained on massive corpora with extensive computational resources. However, this scale introduces major challenges for deployment in real-world environments. Full-precision models require specialized accelerators and high-bandwidth infrastructure. Such demands limit their use on edge devices, mobile platforms, and latency-sensitive applications. As LLM capabilities continue to expand, the ability to compress and adapt these models efficiently without degrading reasoning or generative quality has become a key research objective.

While much effort has focused on scalability and efficiency, an equally pressing challenge lies in ownership security. As LLMs become critical commercial assets, protecting their intellectual property (IP) is as important as optimizing their performance [2]. Model weights and generated outputs both carry proprietary value, exposing systems to model extraction, replication, and unauthorized redistribution. Addressing efficiency and IP protection together, within a single framework, is essential for trustworthy and scalable AI deployment.

Knowledge distillation (KD) provides a strong foundation for model compression. It transfers predictive behavior, or dark knowledge, from a large teacher model to a smaller student model. The student is trained using the teacher's soft output distributions, which encode richer inter-class relationships and uncertainty than hard labels. This allows the student to approximate the teacher's decision boundaries with far fewer parameters. Modern KD variants go further by incorporating intermediate-layer representations and attention maps. These enhancements significantly narrow the performance gap between teacher and student models and improve transfer fidelity in transformer architectures.

Recent advances in parameter-efficient fine-tuning, particularly Low-Rank Adaptation (LoRA), have further amplified the effectiveness of KD. LoRA introduces low-dimensional trainable matrices on top of frozen pre-trained weights, reducing the number of trainable parameters by orders of magnitude while preserving representational fidelity. The combination of KD and LoRA thus offers an efficient paradigm for producing compact, high-performing models that can be deployed on

• F. Jiang and Z. Cai are with the Department of Computer Science, Georgia State University, Atlanta, GA, 30303, USA. Emails: fjiang4@student.gsu.edu, yili@gsu.edu.
• H. Xu is with Department of Information Technology, Kennesaw State University, Marietta, GA, 30060, USA. Email: hxu10@kennesaw.edu.
• D. Seo and W. Kim are with Department of Artificial Intelligence and Data Engineering, Sangmyung University, Seoul, 03016, Republic of Korea. Emails: daehseo@smu.ac.kr, binn.kim29@gmail.com.
• Y. Joe is Director of LSware Inc., Seoul, 08504, Republic of Korea. Email:research@yongjoon.net.
* To whom correspondence should be addressed.
Manuscript received: 20xx-xx-xx; revised: 20xx-xx-xx; accepted: 20xx-xx-xx

Just Accepted

constrained hardware without substantial accuracy loss. However, while such techniques improve computational efficiency, they do not address the ownership and accountability aspects essential to secure model deployment.

In parallel, the open availability and rapid commercialization of LLMs have intensified IP-related risks [3]. These models encode valuable data, architectures, and fine-tuning strategies. As a result, their outputs can reveal or replicate proprietary information [4]. Output-level watermarking provides a practical and efficient way to mitigate this risk. It embeds subtle, statistically verifiable patterns in generated text. These watermarks allow post-hoc ownership verification without modifying model parameters. In practice, watermarking works by slightly biasing token selection probabilities during generation, embedding covert but recoverable signatures. This lightweight approach provides an effective layer of attribution and accountability while preserving the naturalness of generated text.

Despite the individual maturity of KD for compression and watermarking for copyright protection, these two research lines have evolved independently. Most current systems employ a sequential pipeline, either applying watermarking after distillation or distilling from an already watermarked teacher. Such two-stage workflows introduce redundancy, validation overhead, and potential quality degradation. They can also weaken watermarks during compression or retraining. Moreover, decoupling efficiency and protection limits real-world applicability, where performance and ownership assurance must coexist.

To address these limitations, we propose the Adaptive Knowledge Distillation Framework (AKDF), a unified solution that integrates model compression and copyright protection within a single training process. AKDF combines the efficiency of LoRA-based parameter adaptation with probabilistic watermark embedding into one cohesive pipeline. By jointly performing knowledge distillation and watermark integration at the output-logit level, AKDF maintains model fidelity while encoding verifiable ownership signatures. This unified process strengthens resistance to model theft, unauthorized redistribution, and adversarial manipulation. At the same time, it preserves the accuracy and efficiency required for real-world deployment. Unlike post-hoc watermarking schemes, AKDF embeds ownership signals directly during knowledge transfer, ensuring that watermark detectability persists even after compression or text modification.

The main contributions of this work are as follows:

- **Unified framework.** We propose AKDF, a single pipeline that integrates knowledge distillation, low-rank adaptation (LoRA), and probabilistic watermarking to jointly achieve model compression and ownership protection.
- **Robust ownership embedding.** AKDF embeds verifiable watermark signals during distillation, ensuring persistent and tamper-resistant ownership verification under compression and diverse decoding settings.
- **Comprehensive evaluation.** Experiments on ARC-Easy, PIQA, and WMT16 show that a 1B-parameter student retains approximately 76% of the teacher's performance while maintaining consistent watermark detectability and efficiency.

The remainder of this paper is organized as follows. Section 2 reviews related work on knowledge distillation and watermarking for LLMs. Section 3 presents the AKDF architecture. Section 4 describes the experimental setup. Section 5 reports the performance and communication-efficiency analyzes, followed by a discussion of the key findings. Section 6 concludes the paper.

2 Related Work

This section reviews prior research related to the proposed Adaptive Knowledge Distillation Framework (AKDF). The discussion focuses on two major areas: (1) knowledge distillation for large language models, which explores model compression and parameter-efficient learning; and (2) output-level watermarking for large language models, which addresses ownership verification and intellectual-property protection in generative models.

2.1 Knowledge distillation for LLMs

Large language models (LLMs) such as GPT, LLaMA, and BERT have transformed natural language understanding and reasoning, yet their immense scale introduces prohibitive computational and memory costs. Bridging the gap between capability and efficiency has made model compression a central challenge in modern AI research. Knowledge distillation (KD) has emerged as a leading approach to address this issue, transferring the predictive behavior of large teacher models to smaller student networks optimized for efficient deployment.

KD relies on the soft probability outputs, logits, or temperature-scaled distributions of a high-capacity teacher as supervisory signals for the student. Unlike hard labels in standard supervised learning, these soft targets encode nuanced relationships among classes and uncertainty levels. This "dark knowledge", first introduced by Hinton *et al.* [5], enables the student to approximate the teacher's decision boundaries with far fewer parameters, achieving high accuracy at a fraction of the computational cost.

Just Accepted

Over time, KD has progressed beyond basic output imitation to include intermediate-layer and attention-based distillation [6–9]. These extensions allow the student to internalize hierarchical and contextual features, capturing richer representations from transformer-based architectures [10]. The choice of transfer signal has itself become a design dimension: MFD-KD, for instance, transfers knowledge in the frequency domain rather than the spatial domain, applying a Fourier transform to isolate informative frequency components while suppressing noise, and reports consistent gains across both convolutional and transformer students [11]. Such results indicate that the representation in which teacher knowledge is expressed matters as much as the layer from which it is drawn, which motivates our decision to perform distillation at the output-logit level, where the watermark bias can be injected into the same signal. In parallel, parameter-efficient fine-tuning (PEFT) techniques such as Low-Rank Adaptation (LoRA) [12, 13] have greatly increased the practicality of KD. LoRA introduces compact, trainable low-rank matrices on top of frozen pre-trained weights, significantly reducing trainable parameters while maintaining expressive capacity. The synergy between KD and LoRA thus enables efficient adaptation of large-scale models without compromising performance.

In recent years, KD has been increasingly specialized for language models [14]. Early work in this direction targeted encoder architectures such as BERT, showing that linguistic competence can be retained in a much smaller student [15–17]. More recent methods extend the idea to generative LLMs: MiniLLM replaces the forward Kullback–Leibler objective with a reverse formulation better suited to open-ended generation [18], and KD-LoRA couples low-rank adaptation with distillation, reporting accuracy comparable to full fine-tuning at a fraction of the training and memory cost [19]. Compact open models such as TinyLlama [20] demonstrate that architectures at the one-billion-parameter scale can reach useful quality, although they are pre-trained from scratch rather than distilled from a teacher. Surveys of efficient LLMs and of model compression place these lines of work in a common frame [21, 22]: restricting adaptation to a carefully selected subset of parameters balances compactness against fidelity in a way that was previously difficult to achieve.

Contemporary research has also expanded the focus of KD toward deployment realism, emphasizing latency, energy efficiency, and bandwidth optimization for on-device and real-time inference [23]. In addition, multi-teacher and adaptive distillation strategies [24–26] have extended the applicability of KD across architectures and training regimes. Collectively, these developments have positioned KD as a cornerstone of scalable and accessible language modeling [27].

However, most existing KD–LoRA frameworks treat compression and adaptation as independent processes, limiting their ability to dynamically adjust during training. To overcome this limitation, our Adaptive Knowledge Distillation Framework (AKDF) integrates knowledge transfer and parameter-efficient adaptation into a single adaptive pipeline. AKDF preserves the reasoning fidelity of large teacher models while achieving lightweight, deployable, and robust compression suitable for diverse, resource-constrained environments.

2.2 Output-level watermarking for LLMs

As LLMs increasingly drive applications from content generation to automation, ensuring ownership and authenticity of their outputs has become a critical challenge. The rapid spread of open-source and commercial LLMs heightens concerns about intellectual-property protection, attribution, and accountability. To address these issues, watermarking techniques [28, 29] have emerged as essential safeguards, linking generated text to its source model while preserving usability and performance.

Watermarking methods for neural models generally fall into two categories: parameter-level and output-level watermarking. Parameter-level watermarking, first introduced for convolutional networks by Uchida *et al*. [30] and since adapted to language models, embeds identifiable signatures within model weights through structured perturbations or fine-tuning, enabling ownership verification at the model level but often increasing retraining and computational costs. The same distinction, and the same preference for marking the artifact rather than the weights, appears in the image domain: Liu *et al*. [31] embed an adversarial watermark in the frequency domain so that a single mark both deters unauthorized use by deep networks and remains accurately extractable for ownership verification, without degrading perceptual quality. In contrast, output-level watermarking focuses on embedding subtle, statistically detectable signals within generated text [32]. This approach enables post-hoc ownership verification without requiring access to internal model parameters, making it highly practical for deployment in distributed or privacy-sensitive environments.

Output-level watermarking has become popular for its flexibility, efficiency, and non-intrusive design [33]. Instead of altering model parameters, these methods act during inference by adjusting token selection probabilities [32, 33]. Slight biases toward predefined vocabulary subsets embed statistical patterns that are invisible to users but verifiable through analysis. Detection remains reliable under minor textual

Just Accepted

edits—formally, under corruptions of bounded edit distance [33]—while the naturalness of the generated text is preserved. Among the various approaches, one family of algorithms has proven especially effective: the KGW family, named after the green-list watermarking scheme of Kirchenbauer et al. [32].

The KGW family exemplifies simplicity and practicality. KGW selects specific tokens and biases their generation probabilities, creating identifiable statistical fingerprints within text, and detection is performed by a statistical test on the token distribution of a suspected corpus [32]; open-source toolkits such as MarkLLM supply reference implementations and a common evaluation protocol [34]. KGW retains detectability under token-level perturbations such as reordering and substitution, but its robustness is not unconditional: because the green-list partition is derived from the preceding context, strong paraphrasing attacks can substantially degrade detection [28, 35]. Complementary cryptographic schemes, such as the undetectable watermarks of Christ et al. [36], instead derive the token partition from pseudorandom keys, providing formal undetectability guarantees.

Closest to our setting, several recent efforts have explored knowledge distillation itself as a vehicle for copyright protection. KDGAN trains a distilled surrogate model to stand in for the original network, enabling secure and communication-efficient model publishing [37], a lightweight super-resolution model has been distilled to protect the complete functionality of vision models [38], and KDLLM extends copyright-preserving distillation to large language models [39]. Moreover, recent evidence shows that output-level watermarks are themselves learnable by student models during distillation [40]. These studies, however, still decouple ownership protection from the compression objective itself.

In this work, we adopt an output-level watermarking algorithm derived from the KGW family, chosen for its proven robustness, efficiency, and ease of integration [32, 34]. Within our Adaptive Knowledge Distillation Framework (AKDF), the watermark is embedded directly into the distillation process rather than applied post-hoc. This joint optimization ensures that ownership signatures persist even after model compression, preserving verifiable traceability without sacrificing performance. Consequently, AKDF offers a secure, lightweight, and deployable solution for real-world LLMs, combining interpretability, accountability, and protection of model provenance in a unified framework.

3 Methods

LLMs deliver strong reasoning and generation quality, but their scale makes them expensive to train, adapt, and deploy in constrained environments. Our goal is to compress such models while preserving performance. To this end, we propose the Adaptive Knowledge Distillation Framework (AKDF), which integrates knowledge distillation and low-rank parameter adaptation. AKDF transfers knowledge from a large teacher model into a significantly smaller student model while restricting updates to low-rank subspaces for efficiency. Fig. 1 summarizes the full training and supervision pipeline.

3.1 Teacher–student distillation setup

AKDF consists of three core components: a teacher network N_T, a student network N_S, and a Low-Rank Adaptation module (LoRA). The teacher network N_T is a large-scale, pre-trained language model that serves as the source of supervision. Given an input instance $\boldsymbol{x}$, the teacher produces logits $\boldsymbol{z}_T$, which capture rich semantic and structural information learned from large-scale corpora.

The student network N_S is a compact model with substantially fewer parameters, intended for efficient inference and deployment. For the same input $\boldsymbol{x}$, the student produces logits $\boldsymbol{z}_S$. The goal of training is to align N_S with N_T while also respecting the true task labels y.

Following standard knowledge distillation, the student is optimized to mimic the teacher's behavior [5]. In practice, N_S is trained to reduce the discrepancy between $\boldsymbol{z}_S$ and $\boldsymbol{z}_T$, while staying faithful to the ground-truth labels. This allows the student to inherit the decision behavior of a powerful teacher without needing to match its full parameter count.

3.2 Low-rank adaptation

To make training efficient, we equip the student with Low-Rank Adaptation (LoRA). Instead of updating all parameters during training, LoRA constrains the trainable parameter updates to live in a low-rank subspace. Concretely, the change in weights at step t is written as $\boldsymbol{\Delta W}_t = \boldsymbol{A}_t \boldsymbol{B}_t^\top$, where $\boldsymbol{A}_t \in \mathbb{R}^{m\times r}$ and $\boldsymbol{B}_t \in \mathbb{R}^{n\times r}$ are low-rank matrices and $r \ll \min(m, n)$. The effective model weights are then $\boldsymbol{W}_t = \boldsymbol{W}_0 + \boldsymbol{\Delta W}_t$, where $\boldsymbol{W}_0$ are frozen pre-trained weights and $\boldsymbol{\Delta W}_t$ is the learned low-rank offset.

This design significantly reduces the number of trainable parameters, improves training speed [12], and lowers memory requirements while maintaining performance. All symbols used in this section are summarized in Table 1.

3.3 Unified optimization

We present a unified update view that captures both standard KD and LoRA-style adaptation. Let $\boldsymbol{W}_t = \boldsymbol{W}_0 + \boldsymbol{\Delta W}_t$, where $\boldsymbol{W}_0$ represents the frozen base weights and $\boldsymbol{\Delta W}_t$ the

Just Accepted

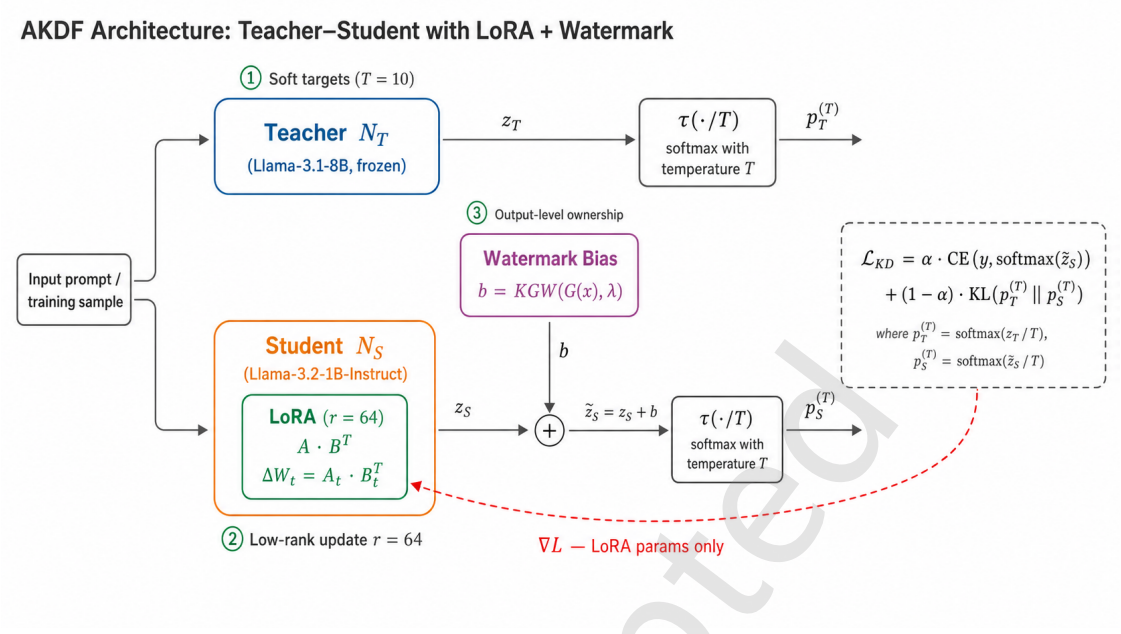

Fig. 1 AKDF architecture.

Table 1 List of notations used in this paper.

Symbol	Description
N_T	Teacher network
N_S	Student network
$\boldsymbol{x}$	Input data
y	Ground-truth labels
$\boldsymbol{z}_T$	Logits from teacher network
$\boldsymbol{z}_S$	Logits from student network
$\boldsymbol{W}_t$	Model parameters at step t
$\boldsymbol{W}_0$	Frozen pre-trained base weights
$\boldsymbol{\Delta W}_t$	Parameter update at step t
$\boldsymbol{A}_t, \boldsymbol{B}_t$	Low-rank matrices
r	LoRA rank
η	Learning rate
T	Temperature parameter
$\mathcal{L}_{\mathrm{KD}}$	Knowledge-distillation loss
$\mathcal{L}_{\mathrm{CE}}$	Cross-entropy loss
$\mathcal{L}_{\mathrm{KL}}$	Kullback–Leibler divergence
$\mathcal{L}_{\mathrm{total}}$	Overall training loss
α	Loss-balance hyperparameter
λ	Regularization weight
δ	Watermark logit bias strength
$\boldsymbol{b}$	Watermark logit-bias vector
$\tilde{\boldsymbol{z}}_S$	Watermark-biased student logits
$\mathcal{G}(\boldsymbol{x})$	KGW key generator (green-list selector)

trainable low-rank update. We write the general update rule as

$$\boldsymbol{W}_{t+1} \leftarrow \boldsymbol{W}_t - \eta \frac{\partial \mathcal{L}_{\mathcal{D}}}{\partial\big(\boldsymbol{W}_0 + f(\boldsymbol{\Delta W}_t)\big)}, \tag{1}$$

where $\mathcal{L}_{\mathcal{D}}$ is a loss over dataset $\mathcal{D}$, and $f(\cdot)$ specifies the constraint (identity for unconstrained KD, low-rank for LoRA). This view allows us to express multiple adaptation mechanisms within one update formula.

3.4 Distillation objective

The training signal in AKDF is defined through a compound loss that balances task supervision and teacher imitation. Let $\boldsymbol{b}$ denote the KGW watermark bias and $\tilde{\boldsymbol{z}}_S = \boldsymbol{z}_S + \boldsymbol{b}$ the watermark-biased student logits. Here $\boldsymbol{b}$ is the logit offset produced by the KGW procedure: a key generator $\mathcal{G}(\boldsymbol{x})$ maps the current decoding context $\boldsymbol{x}$ to a context-dependent hash key that deterministically splits the vocabulary into a pseudo-random "green" list and its complementary "red" list, and $\textsc{KGWBias}(\mathcal{G}(\boldsymbol{x}), \delta)$ adds the bias δ to the green-list logits (and zero elsewhere) to form $\boldsymbol{b}$, as used in Algorithm 1. Let $\boldsymbol{p}_T^{(T)} = \mathrm{softmax}(\boldsymbol{z}_T/T)$ and $\boldsymbol{p}_S^{(T)} = \mathrm{softmax}(\tilde{\boldsymbol{z}}_S/T)$ denote the temperature-scaled teacher and student distributions with temperature T. We refer to the resulting compound objective as the knowledge-distillation loss

$$\begin{aligned}\mathcal{L}_{\mathrm{KD}} = \alpha\, \mathcal{L}_{\mathrm{CE}}\big(y,\, \mathrm{softmax}(\tilde{\boldsymbol{z}}_S)\big) \\ + (1-\alpha)\, T^2\, \mathcal{L}_{\mathrm{KL}}\big(\boldsymbol{p}_T^{(T)} \,\big\|\, \boldsymbol{p}_S^{(T)}\big),\end{aligned} \tag{2}$$

where $\alpha \in [0, 1]$ controls the trade-off between ground-truth cross-entropy and alignment with the teacher's softened output distribution, and the T^2 factor follows standard practice in knowledge distillation [5] so that the gradient magnitude of the soft-target term remains comparable to that of the hard-label term across temperatures. Both the task loss and the distillation loss are computed on the watermark-biased logits $\tilde{\boldsymbol{z}}_S$, so that the LoRA update is aware of the watermark

Just Accepted

bias and learns to embed the ownership signal jointly with knowledge transfer, rather than treating watermarking as a post-hoc step. This term encourages the student to acquire information that would not be available from hard labels alone.

During standard LoRA-style fine-tuning without KD, the update would be driven only by cross-entropy:

$$\boldsymbol{W}_{t+1} \leftarrow \boldsymbol{W}_t - \eta \frac{\partial \mathcal{L}_{\mathrm{CE}}(\boldsymbol{x}, y)}{\partial \big(\boldsymbol{W}_0 + \boldsymbol{A}_t \boldsymbol{B}_t^{\top}\big)}. \tag{3}$$

In AKDF, we replace this with an update on the full AKDF objective $\mathcal{L}_{\mathrm{total}}$ defined in Eq. (5):

$$\boldsymbol{W}_{t+1} \leftarrow \boldsymbol{W}_t - \eta \frac{\partial \mathcal{L}_{\mathrm{total}}}{\partial \boldsymbol{\Delta W}_t}, \tag{4}$$

where the gradient flows through the watermark-biased logits $\tilde{\boldsymbol{z}}_S$ in both the cross-entropy and the KL term of Eq. (2), so the low-rank adapter parameters are jointly optimized for task fidelity, teacher alignment, and watermark embedding within a single update.

3.5 Overall objective

To regularize the adaptation and promote compact, efficient updates, we define the overall training loss as

$$\mathcal{L}_{\mathrm{total}} = \lambda \mathcal{L}_{\mathrm{KD}} + (1-\lambda) \|\boldsymbol{\Delta W}_t\|_F^2, \tag{5}$$

where $\lambda \in [0,1]$ balances fidelity to the teacher with a Frobenius-norm penalty on the magnitude of the low-rank update. This encourages stable, low-overhead adapters that can be deployed or transmitted efficiently.

In summary, AKDF trains a compact student model by combining (i) supervision from a powerful teacher, (ii) low-rank parameter adaptation for efficiency, and (iii) a loss that explicitly constrains adaptation strength. The full optimization loop is given in Algorithm 1.

Algorithm 1 Adaptive Knowledge Distillation Framework (AKDF).

Input: batches $\{(\boldsymbol{x}, y)\}$, teacher N_T, student N_S, key-generator $\mathcal{G}(\boldsymbol{x})$, temperature T, weights α, λ, and δ, learning rate η.

Output: trained student parameters $\boldsymbol{W}_{N_S}$.

Freeze base weights of N_S; enable LoRA adapters; set N_T to evaluation mode; initialise AdamW on LoRA parameters with learning rate η;

for each minibatch $(\boldsymbol{x}, y)$ **do**

- $\boldsymbol{z}_T \leftarrow N_T(\boldsymbol{x})$; `// teacher forward, no gradient`
- $\boldsymbol{z}_S \leftarrow N_S(\boldsymbol{x})$; `// student forward`
- $\boldsymbol{b} \leftarrow \mathrm{KGWBias}\big(\mathcal{G}(\boldsymbol{x}), \delta\big)$;
- $\tilde{\boldsymbol{z}}_S \leftarrow \boldsymbol{z}_S + \boldsymbol{b}$; `// watermark-biased logits`
- $\boldsymbol{p}_T^{(T)} \leftarrow \mathrm{softmax}(\boldsymbol{z}_T / T)$;
- $\boldsymbol{p}_S^{(T)} \leftarrow \mathrm{softmax}(\tilde{\boldsymbol{z}}_S / T)$;
- $\mathcal{L}_{\mathrm{KD}} \leftarrow \alpha\, \mathrm{CE}\big(y, \mathrm{softmax}(\tilde{\boldsymbol{z}}_S)\big) + (1-\alpha)\, T^2\, \mathrm{KL}\big(\boldsymbol{p}_T^{(T)} \,\|\, \boldsymbol{p}_S^{(T)}\big)$;
- $\mathcal{L}_{\mathrm{total}} \leftarrow \lambda \mathcal{L}_{\mathrm{KD}} + (1-\lambda) \|\boldsymbol{\Delta W}_t\|_F^2$;
- Update LoRA parameters using $\nabla \mathcal{L}_{\mathrm{total}}$;

end for

return $\boldsymbol{W}_{N_S}$;

4 Experiment Settings

Fig. 2 provides task-level examples on the three benchmarks used in our evaluation. The trained AKDF student (1B, orange) integrates LoRA adapters and an output-level watermark module (WM), and is evaluated on three benchmarks: ARC-Easy (logical reasoning, accuracy), PIQA (physical commonsense, accuracy), and WMT16 De→En (translation, BLEU). Bars report relative performance normalized by the Teacher (8B = 100%); absolute scores are shown beneath each pair. AKDF retains approximately 76% of the teacher's performance on average across the three tasks (77%, 72%, and 80% respectively) while using one eighth of its parameters. Detailed comparisons against Untrained Student (1B), Vanilla KD, and MARKLLM (KGW)—with the Teacher as upper bound—are given in Table 4 and Fig. 4.

4.1 Datasets and evaluation benchmarks

We fine-tune the student model under the AKDF framework using the BAAI Infinity-Instruct dataset, comprising approximately 80 000 instruction–response pairs. To assess the generalization and effectiveness of AKDF, we evaluate the distilled model across three diverse benchmarks: ARC-Easy for logical reasoning, PIQA for physical-commonsense reasoning, and WMT16 for German–English translation. Performance is measured using accuracy for the multiple-choice tasks and BLEU for the translation task, collectively capturing the model's reasoning and generative capabilities.

4.2 Hyper-parameter settings

To identify optimal hyper-parameters for AKDF, we fine-tune the student model (`Llama-3.2-1B-Instruct`) distilled from the teacher (`Llama-3.1-8B`) using LoRA adapters. A systematic grid search explores LoRA ranks from 2 to 256 and learning rates between 1×10^{-4} and 1×10^{-3}. Fig. 3 shows that increasing the LoRA rank improves accuracy up to rank 64, which offers the best balance between performance and efficiency. Learning rates near 2×10^{-4} yield stable convergence, whereas higher values (e.g., 1×10^{-3}) lead to degraded performance, particularly at larger ranks.

Just Accepted

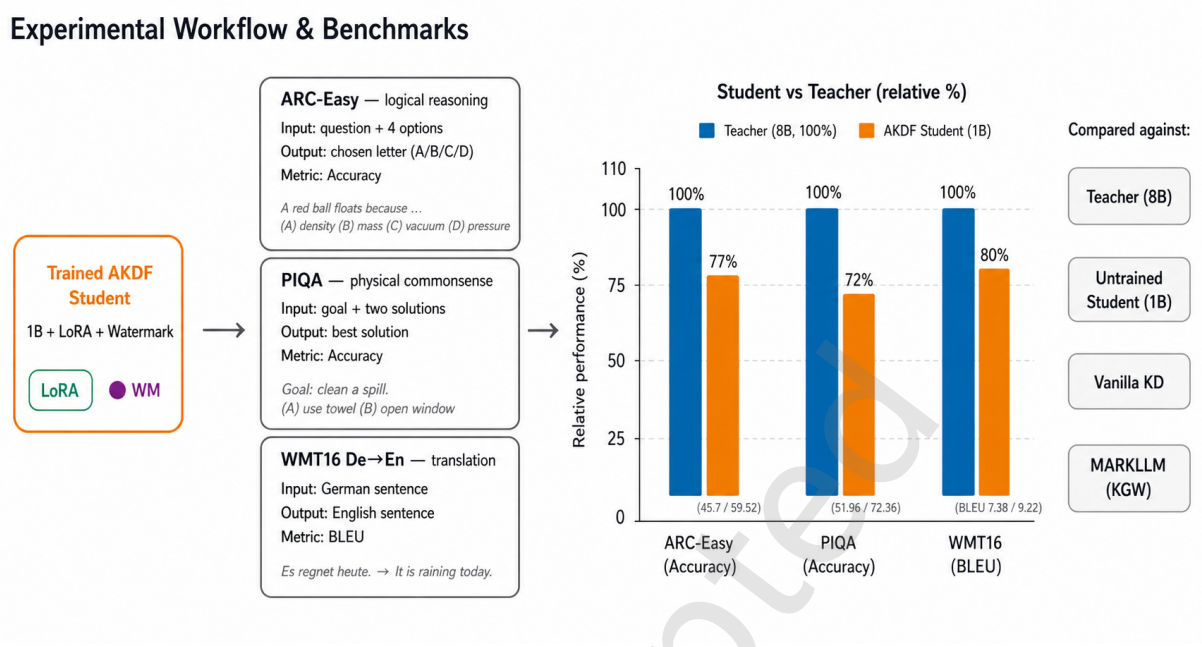

Fig. 2 Experimental setup overview.

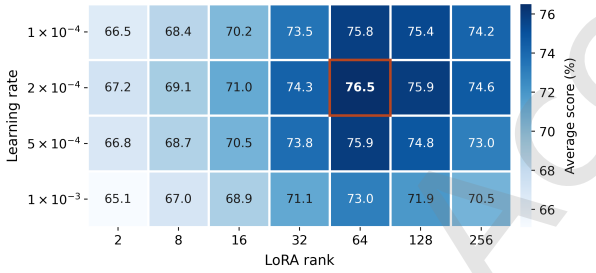

Fig. 3 Grid-search results over LoRA ranks (2–256) and learning rates (1×10^{-4}–1×10^{-3}). Each cell reports the average of 10 reasoning metrics on our internal evaluation set; darker shades indicate better performance. The best configuration, rank 64 with $\eta = 2 \times 10^{-4}$, is outlined. The same values are listed numerically in Table 5.

Training is performed for two epochs with an effective batch size of 64 (per-GPU batch size of 2 and gradient accumulation of 16). Input sequences are truncated or padded to a maximum length of 2048 tokens, and the distillation and classification losses are equally weighted. All experiments are conducted on a workstation with four NVIDIA Tesla V100 GPUs (32 GB each) using the Hugging Face Transformers and PEFT libraries.

4.3 Baselines

We compare AKDF against four representative baselines to ensure a fair and comprehensive evaluation. The Teacher (8B) baseline uses the original Llama-3.1-8B model as an upper bound, representing the full-capacity system before compression. The Untrained Student (1B) baseline employs a randomly initialized student model of equivalent size (1B parameters) to provide a lower bound without knowledge transfer. The Vanilla Knowledge Distillation (KD) baseline trains the student using traditional KD to capture the benefits of standard distillation without watermark integration. Finally, the MARKLLM (KGW) baseline applies the green-list watermarking scheme of Kirchenbauer *et al.* [32], using the reference implementation provided by the MarkLLM toolkit [34]; the teacher's outputs are embedded with KGW-style statistical watermarks before distillation, so this baseline represents the state of the art in output-level ownership protection. For brevity we label it MARKLLM (KGW) in the tables and figures, with the understanding that KGW is the method and MarkLLM the implementation. Together, these baselines provide a comprehensive foundation to assess the effectiveness of AKDF in terms of model compression, knowledge-transfer fidelity, and watermark robustness. The architectural contrast between the teacher and the LoRA-enhanced student is summarized in Table 2.

4.4 Model-architecture comparison

Table 2 highlights the architectural contrast between the teacher (`Llama-3.1-8B`) and the LoRA-enhanced student (`Llama-3.2-1B-Instruct`). The teacher is a full-capacity decoder-only transformer with 8 billion parameters and no constraints during inference. In contrast, the student integrates Low-Rank Adaptation (LoRA, rank = 64), yielding roughly

Just Accepted

Table 2 Architectural comparison of teacher and student models.

Component	Teacher (Llama-3.1-8B)	Student (Llama-3.2-1B-Instruct)
Model size	8B parameters	1B parameters
LoRA applied	No	Yes (rank = 64)
Trainable parameters	Full	∼10M LoRA params
Token limit	2048	2048
Architecture type	Decoder-only	Decoder-only
Embedding & vocabulary	Shared	Shared
Distillation loss	–	CE + KL divergence
Watermarking	Logit bias (MARKLLM)	Embedded in outputs
ARC accuracy	59.52%	45.7%
WMT16 BLEU	9.22	7.38

Table 3 Teacher–student performance retention per benchmark. Retention is the ratio of the AKDF student score to the teacher score; the mean is the unweighted average over the three benchmarks.

Benchmark	Teacher	AKDF student	Retention
ARC (accuracy)	59.52%	45.7%	76.8%
PIQA (accuracy)	72.36%	51.96%	71.8%
WMT16 (BLEU)	9.22	7.38	80.0%
Mean			76.2%

10 million trainable parameters while keeping the base weights frozen.

Despite the drastic reduction in trainable parameters, the student preserves the teacher's 2048-token context window, shared vocabulary, and transformer backbone, ensuring structural alignment for effective knowledge transfer. It also embeds output-level watermarking during distillation, providing verifiable ownership and traceability.

The student retains approximately 76% of the teacher's performance on average across the three benchmarks (Table 3), demonstrating the ability of AKDF to achieve substantial compression with minimal loss in reasoning and translation quality.

5 Results and Discussion

5.1 Performance evaluation

We evaluate AKDF against the four baselines introduced above (Teacher (8B), Untrained Student (1B), Vanilla KD, and MARKLLM (KGW)) to ensure a fair and comprehensive comparison. Table 4 reports results on three benchmarks: ARC for reasoning accuracy, PIQA for commonsense inference, and WMT16 (German–English) for translation quality measured by BLEU score.

The teacher model, with 8 billion parameters, achieved the best performance across all tasks. It reached 59.52% accuracy on ARC, 72.36% on PIQA, and a BLEU score of 9.22 on

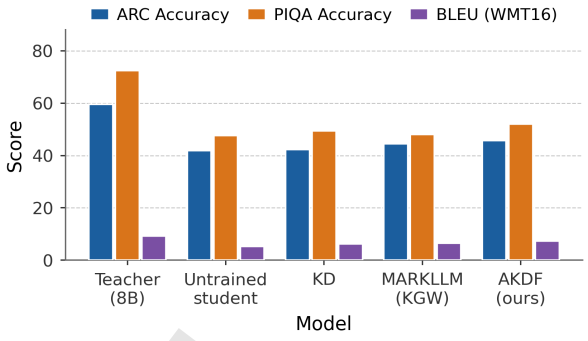

Fig. 4 Performance comparison of different models across the three benchmark tasks: ARC (logical reasoning), PIQA (physical-commonsense reasoning), and WMT16 (German-to-English translation).

WMT16. These results confirm the advantage of large-scale models in capturing complex linguistic and semantic patterns.

The untrained student model performed significantly worse, illustrating the limits of smaller models without knowledge transfer. It achieved 41.8% on ARC, 47.62% on PIQA, and a BLEU score of 5.23 on WMT16. This gap highlights the need for effective distillation to transfer capability from large teachers to compact students.

Training the student with vanilla KD led to small but consistent improvements. It achieved 42.23% on ARC, 49.34% on PIQA, and a BLEU score of 6.24. These gains indicate that traditional KD helps transfer some generalization ability but fails to fully capture fine-grained linguistic and reasoning features.

MARKLLM, which applies KGW-style watermark embedding, reached 44.48% on ARC, 48.04% on PIQA, and 6.52 BLEU on WMT16. It therefore improved over vanilla KD on ARC and WMT16 but fell slightly below it on PIQA, suggesting that watermarking adds minor gains overall while introducing trade-offs that can affect downstream accuracy.

AKDF achieved the best overall performance among student models. It obtained 45.7% on ARC, 51.96% on PIQA, and 7.38 BLEU on WMT16. Compared with vanilla KD and MARKLLM, AKDF improved reasoning accuracy and translation quality while maintaining stable performance across all benchmarks. Fig. 4 shows consistent gains under all evaluation settings.

The improvements of AKDF likely stem from combining low-rank parameter adaptation with softened teacher supervision. This setup enhances knowledge-transfer efficiency while avoiding the performance degradation often caused by watermark embedding. The framework effectively balances compression, fidelity, and ownership signaling.

In summary, AKDF approximates the teacher's perfor-

Just Accepted

Table 4 Performance comparison across baselines and AKDF.

Metric	Teacher (8B)	Untrained student	KD	MARKLLM (KGW)	Ours (AKDF)
Accuracy (ARC)	59.52%	41.8%	42.23%	44.48%	**45.7%**
Accuracy (PIQA)	72.36%	47.62%	49.34%	48.04%	**51.96%**
BLEU (WMT16)	9.22	5.23	6.24	6.52	**7.38**

Table 5 Average score (%) by LoRA rank and learning rate, averaged over the 10 reasoning metrics of our internal evaluation set. These are the values plotted in Fig. 3.

LoRA rank	$\eta = 1 \times 10^{-4}$	$\eta = 2 \times 10^{-4}$	$\eta = 5 \times 10^{-4}$	$\eta = 1 \times 10^{-3}$
2	66.5	67.2	66.8	65.1
8	68.4	69.1	68.7	67.0
16	70.2	71.0	70.5	68.9
32	73.5	74.3	73.8	71.1
64	75.8	**76.5**	75.9	73.0
128	75.4	75.9	74.8	71.9
256	74.2	74.6	73.0	70.5

mance despite a major parameter reduction. It surpasses conventional KD and watermarking baselines in maintaining semantic integrity and computational efficiency, making it a practical solution for secure and resource-aware deployment.

5.2 Ablation study

We performed an ablation study to analyze how architectural and training hyper-parameters affect the AKDF framework. The analysis focused on two main factors: (1) the choice of LoRA rank and learning rate, and (2) the influence of distillation temperature and the loss-balancing factor on student performance.

5.2.1 *Effect of LoRA rank and learning rate*

We varied the LoRA rank from 2 to 256 and the learning rate from 1×10^{-4} to 1×10^{-3}, performing a grid search to identify optimal settings. As shown in Fig. 3, increasing the LoRA rank improves performance up to rank 64 because of the greater expressive capacity of the adapter, after which performance declines slightly. Rank 64 therefore provides the best trade-off between efficiency and accuracy. A moderate learning rate of 2×10^{-4} is best at every rank, outperforming both the smaller 1×10^{-4} and the larger 5×10^{-4} and 1×10^{-3} settings; the degradation at 1×10^{-3} is most pronounced at larger ranks, suggesting that aggressive updates can harm generalization during low-rank adaptation.

Table 5 lists the same grid numerically. The best configuration is LoRA rank 64 with a 2×10^{-4} learning rate, which attains an average score of 76.5%; the weakest setting, rank 2 with 1×10^{-3}, reaches 65.1%. Learning rates above 5×10^{-4} consistently reduce stability and accuracy.

5.2.2 *Effect of temperature and loss balancing*

We next analyze the effect of the temperature T in the softmax operation and the weighting factor α in the distillation loss

Table 6 Student performance under different distillation temperatures.

Temperature	ARC accuracy	PIQA accuracy	WMT16 BLEU
2	44.5	50.0	6.80
5	45.3	51.2	7.12
10	**45.7**	**51.96**	**7.38**
20	44.9	50.9	7.10
30	44.3	49.8	6.95

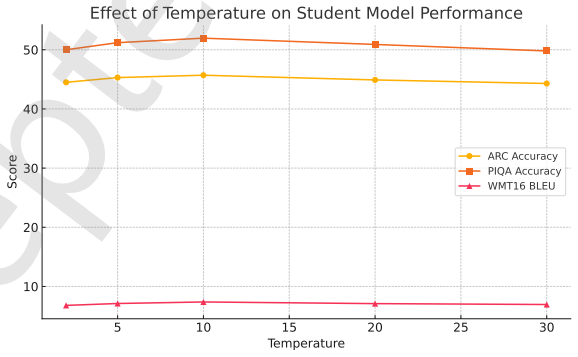

Fig. 5 Effect of distillation temperature on ARC and PIQA accuracy and on WMT16 BLEU score. Temperature $T = 10$ provides the best trade-off.

$\mathcal{L}_{\text{KD}}$ defined in Eq. (2). A moderate temperature ($T = 10$) gave the best trade-off across all benchmarks, improving BLEU scores and accuracy on ARC and PIQA, as visualized in Fig. 5. Low temperatures ($T = 2$) made the soft labels too sharp and reduced diversity, while high values ($T = 20$ or 30) blurred probability distributions and weakened supervision.

We also varied the distillation weight α from 0.1 to 0.9. The balanced setting $\alpha = 0.5$ performed best, indicating that equal weight on ground-truth labels and teacher guidance yields stable learning. Very small or large α values caused under-utilization of teacher knowledge or overfitting to noisy soft labels.

Table 6 shows that a moderate temperature ($T = 10$) achieves the best results on ARC, PIQA, and WMT16. Low temperatures ($T = 2$) make distributions overly sharp, while high temperatures dilute the information in teacher signals.

Overall, AKDF achieves its best performance with LoRA rank 64, a learning rate of 2×10^{-4}, temperature $T = 10$, and loss-balancing factor $\alpha = 0.5$. These settings maximize compression and transfer efficiency, making AKDF suitable for deployment in environments with limited computational

Just Accepted

Table 7 Model-publishing time at different bandwidths.

Bandwidth	Llama-3.1-8B	MarkLLM	Ours (AKDF)
0.5 MB/s	61 248 s	61 248 s	9 426 s (−84.6%)
1 MB/s	30 624 s	30 624 s	4 712 s (−84.6%)
2 MB/s	15 312 s	15 312 s	2 357 s (−84.6%)

and bandwidth resources.

5.3 Communication efficiency

We analyze the communication cost of publishing each model by projecting its upload time directly from its checkpoint size. A checkpoint of fixed size s takes s/B seconds to upload at bandwidth B, so once B is fixed the publishing time follows deterministically from the model's on-disk footprint. We report this projection for the teacher model (`Llama-3.1-8B`), the watermarking-based model (`MARKLLM`), and the AKDF-compressed student model at three representative upload speeds, 0.5 MB/s, 1 MB/s, and 2 MB/s.

As shown in Table 7, the AKDF-compressed checkpoint requires about one-sixth of the teacher's projected upload time. Because `MARKLLM` embeds an output-level watermark without changing the parameter count, its checkpoint footprint—and hence its projected upload time—coincides with the teacher's, so AKDF attains the same 84.6% reduction relative to both. This reduction is a property of the checkpoint-size ratio rather than of the link: the bandwidth B cancels in the ratio, which is why the projected saving is identical at 0.5, 1, and 2 MB/s. The distilled student's smaller footprint therefore translates directly into proportionally faster publishing under any fixed bandwidth, which is most beneficial in bandwidth-constrained settings.

5.4 Discussion

AKDF provides a unified approach for compressing LLMs while embedding copyright-protecting watermarks during training. The framework integrates knowledge distillation with low-rank parameter adaptation (LoRA), reducing model size substantially while maintaining competitive performance. Experiments show that AKDF consistently improves over traditional distillation and watermarking baselines, preserving approximately 76% of the teacher's performance across benchmark tasks.

The approach benefits from the parameter-efficient design of LoRA and the use of Kullback–Leibler divergence as a stable distillation objective under moderate temperature scaling. This configuration enables effective knowledge transfer without significant computational overhead, making AKDF practical for deployment in bandwidth-limited or latency-sensitive environments.

This study focuses on classification and translation tasks. Extending AKDF to generative applications such as dialogue, summarization, or long-form text generation will require addressing additional challenges, including sequence-level coherence and contextual consistency. Potential directions include integrating sequence-level objectives, teacher-forcing strategies, or reinforcement-learning-based distillation.

Future work may explore alternative LoRA configurations and watermarking strategies to enhance robustness against adversarial modifications. Applying AKDF to federated or multimodal learning systems offers a promising path toward secure, distributed deployment where model ownership and data privacy are critical.

Overall, AKDF demonstrates that compression and copyright preservation can be jointly optimized, paving the way for more accessible and trustworthy language-model deployment.

6 Conclusion

This paper introduced the Adaptive Knowledge Distillation Framework (AKDF), a unified approach for compressing large language models while preserving intellectual property through embedded watermarking. AKDF integrates parameter-efficient low-rank adaptation (LoRA) with an output-level watermarking mechanism during distillation, enabling high-capacity teacher models to be compressed into smaller student models. This design facilitates deployment in resource-constrained environments while embedding copyright protection directly into the model outputs. Comprehensive evaluations on ARC, PIQA, and WMT16 show that AKDF outperforms traditional knowledge-distillation and watermarking baselines. The 1-billion-parameter student model retains approximately 76% of the teacher's performance across tasks, achieving competitive accuracy and translation quality. AKDF also reduces communication overhead by about 84%, demonstrating its practical efficiency for model publishing under limited bandwidth. Overall, AKDF provides a scalable and secure framework for deploying compressed language models without compromising performance or ownership integrity. Future work will focus on strengthening watermark robustness against adversarial manipulation and extending AKDF to broader domains such as generative and multimodal systems.

Acknowledgment

This work was supported by SW Copyright Ecosystem R&D Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports, and Tourism in 2024 (No. RS-2023-00224818).

Just Accepted

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

Reference

[1] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, Language models are few-shot learners, in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 1877–1901.

[2] Y. Yao, J. Duan, K. Xu, Y. Cai, Z. Sun, and Y. Zhang, A survey on large language model (LLM) security and privacy: The good, the bad, and the ugly, *High-Confidence Computing*, vol. 4, no. 2, p. 100211, 2024.

[3] B. Yan, K. Li, M. Xu, Y. Dong, Y. Zhang, Z. Ren, and X. Cheng, On protecting the data privacy of large language models (LLMs) and LLM agents: A literature review, *High-Confidence Computing*, vol. 5, no. 2, p. 100300, 2025.

[4] F. B. Mueller, R. Görge, A. K. Bernzen, J. C. Pirk, and M. Poretschkin, LLMs and memorization: On quality and specificity of copyright compliance, in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society (AIES)*, vol. 7, no. 1, 2024, pp. 984–996.

[5] G. Hinton, O. Vinyals, and J. Dean, Distilling the knowledge in a neural network, *arXiv preprint arXiv:1503.02531*, 2015.

[6] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, FitNets: Hints for thin deep nets, in *Proc. 3rd Int. Conf. Learning Representations (ICLR)*, 2015.

[7] S. Zagoruyko and N. Komodakis, Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer, in *Proc. 5th Int. Conf. Learning Representations (ICLR)*, 2017.

[8] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, TinyBERT: Distilling BERT for natural language understanding, in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 4163–4174.

[9] S. Sun, Y. Cheng, Z. Gan, and J. Liu, 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)*, 2019, pp. 4323–4332.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, Attention is all you need, in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 5998–6008.

[11] T. Dai, R. Huang, H. Guo, J. Wang, B. Li, and Z. Zhu, MFD-KD: Multi-scale frequency-driven knowledge distillation, *Big Data Mining and Analytics*, vol. 8, no. 3, pp. 563–574, 2025.

[12] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, LoRA: Low-rank adaptation of large language models, in *Proc. 10th Int. Conf. Learning Representations (ICLR)*, 2022.

[13] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. de Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, Parameter-efficient transfer learning for NLP, in *Proceedings of the 36th International Conference on Machine Learning*, 2019, pp. 2790–2799.

[14] X. Xu, M. Li, C. Tao, T. Shen, R. Cheng, J. Li, C. Xu, D. Tao, and T. Zhou, A survey on knowledge distillation of large language models, *arXiv preprint arXiv:2402.13116*, 2024.

[15] I. Turc, M.-W. Chang, K. Lee, and K. Toutanova, Well-read students learn better: The impact of student initialization on knowledge distillation, *arXiv preprint arXiv:1908.08962*, 2019.

[16] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter, *arXiv preprint arXiv:1910.01108*, 2019.

[17] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou, MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers, in *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[18] Y. Gu, L. Dong, F. Wei, and M. Huang, MiniLLM: Knowledge distillation of large language models, in *Proc. 12th Int. Conf. Learning Representations (ICLR)*, 2024.

[19] R. Azimi, R. Rishav, M. Teichmann, and S. Ebrahimi Kahou, KD-LoRA: A hybrid approach to efficient fine-tuning with LoRA and knowledge distillation, in *Proceedings of the 4th NeurIPS Efficient Natural Language and Speech Processing Workshop*, ser. Proceedings of Machine Learning Research, vol. 262. PMLR, 2024, pp. 73–80.

[20] P. Zhang, G. Zeng, T. Wang, and W. Lu, TinyLlama: An open-source small language model, *arXiv preprint arXiv:2401.02385*, 2024.

[21] Z. Wan, X. Wang, C. Liu, S. Alam, Y. Zheng, J. Liu, Z. Qu, S. Yan, Y. Zhu, Q. Zhang, M. Chowdhury, and M. Zhang, Efficient large language models: A survey, *Transactions on Machine Learning Research*, 2024.

[22] X. Zhu, J. Li, Y. Liu, C. Ma, and W. Wang, A survey on model compression for large language models, *Transactions of the Association for Computational Linguistics*, vol. 12, pp. 1556–1577, 2024.

[23] J. Xu, Z. Li, W. Chen, Q. Wang, X. Gao, Q. Cai, and Z. Ling, On-device language models: A comprehensive review, *arXiv preprint arXiv:2409.00088*, 2024.

[24] S. You, C. Xu, C. Xu, and D. Tao, Learning from multiple teacher networks, in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 1285–1294.

[25] S. Du, S. You, X. Li, J. Wu, F. Wang, C. Qian, and C. Zhang, Agree to disagree: Adaptive ensemble knowledge distillation in

Just Accepted

gradient space, in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 12 345–12 355.
[26] D. Y. Park, M.-H. Cha, C. Jeong, D. Kim, and B. Han, Learning student-friendly teacher networks for knowledge distillation, in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 13 292–13 303.
[27] J. Gou, B. Yu, S. J. Maybank, and D. Tao, Knowledge distillation: A survey, *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1789–1819, 2021.
[28] A. Liu, L. Pan, Y. Lu, J. Li, X. Hu, X. Zhang, L. Wen, I. King, H. Xiong, and P. S. Yu, A survey of text watermarking in the era of large language models, *ACM Computing Surveys*, vol. 57, no. 2, pp. 1–36, 2024.
[29] Y. Liang, J. Xiao, W. Gan, and P. S. Yu, Watermarking techniques for large language models: A survey, *Artificial Intelligence Review*, vol. 59, no. 2, p. 74, 2026.
[30] Y. Uchida, Y. Nagai, S. Sakazawa, and S. Satoh, Embedding watermarks into deep neural networks, in *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval (ICMR)*, 2017, pp. 269–277.
[31] Y. Liu, S. Ai, Z. Zhou, W. Pang, C. Dong, H. Ge, D. Liao, and Y. Huang, Image copyright dual-protection based on extractable and imperceptible adversarial watermark, *Big Data Mining and Analytics*, vol. 9, no. 3, pp. 719–734, 2026.
[32] J. Kirchenbauer, J. Geiping, Y. Wen, J. Katz, I. Miers, and T. Goldstein, A watermark for large language models, in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 202. PMLR, 2023, pp. 17 061–17 084.
[33] X. Zhao, P. Ananth, L. Li, and Y.-X. Wang, Provable robust watermarking for AI-generated text, in *Proc. 12th Int. Conf. Learning Representations (ICLR)*, 2024.
[34] L. Pan, A. Liu, Z. He, Z. Gao, X. Zhao, Y. Lu, B. Zhou, S. Liu, X. Hu, L. Wen, I. King, and P. S. Yu, MarkLLM: An open-source toolkit for LLM watermarking, in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2024, pp. 61–71.
[35] K. Krishna, Y. Song, M. Karpinska, J. Wieting, and M. Iyyer, Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense, in *Advances in Neural Information Processing Systems*, vol. 36, 2023, pp. 27 469–27 500.
[36] M. Christ, S. Gunn, and O. Zamir, Undetectable watermarks for language models, in *Proceedings of Thirty Seventh Conference on Learning Theory (COLT)*, ser. Proceedings of Machine Learning Research, vol. 247. PMLR, 2024, pp. 1125–1139.
[37] B. Xie, H. Xu, D. Seo, D. Shin, and Z. Cai, KDGAN: Knowledge distillation-based model copyright protection for secure and communication-efficient model publishing, *IET Communications*, vol. 18, no. 14, pp. 860–868, 2024.
[38] B. Xie, H. Xu, Y. Joe, D. Seo, and Z. Cai, Lightweight super-resolution model for complete model copyright protection, *Tsinghua Science and Technology*, vol. 29, no. 4, pp. 1194–1205, 2024.
[39] S. Shrestha, H. Xu, Z. Xie, D. Seo, Y. Joe, W. Kim, and Y. Li, KDLLM: Copyright-preserving LLM based on knowledge distillation, *Tsinghua Science and Technology*, 2025, just Accepted.
[40] C. Gu, X. L. Li, P. Liang, and T. Hashimoto, On the learnability of watermarks for language models, in *Proc. 12th Int. Conf. Learning Representations (ICLR)*, 2024.

Author biography

Feng Jiang is currently pursuing a Ph.D. degree in the Department of Computer Science at Georgia State University. He received his B.S. degree from Chongqing University. His research interests include differential privacy, large language models, knowledge distillation, and intellectual property protection of generative models.

Honghui Xu received a Ph.D. degree in computer science from Georgia State University, Atlanta, GA, USA, in 2023 and a Bachelor's degree from University of Electronic Science and Technology of China, in 2019. He is currently an Assistant Professor of Information Technology with Kennesaw State University, Kennesaw, GA, USA. His research focuses on Trustworthy AI, Efficient AI, Generative AI, and Internet of Things.

Daehee Seo received the B.S. degree in Electrical and Electronic Engineering from the Dongshin University, Naju, S.Korea, in 2001, and the M.S. and Ph.D. degrees in computing science from the Soonchunhyang University, S.Korea, in 2003 and 2006, respectively. He is currently an Assistant Professor with the National Center of Excellence in Software, SangMyung University, Seoul, S.Korea. His research interests include wireless networks, RegTech, Big data security analytics, blockchain security, Cryptography and Information theory.

Yongjoon Joe received his B.A. and M.Sc. in Information Science from Kyushu University, Japan, in 2011 and 2013, respectively. Since 2016 he has been a Director at LSware Inc. (Korea), where he leads R&D on trackable, reliable systems, including blockchain-based Software Bills of Materials (SBoM) for vulnerability and copyright management. His expertise encompasses blockchain, concurrency, security, and copyright, and his research interests extend to game theory and parallel simulation computing.

Just Accepted

Wonbin Kim received his B.S., M.S., and Ph.D. degrees in Computer Science and Engineering from Soonchunhyang University, Republic of Korea, in 2015, 2017, and 2022, respectively. From 2022 to 2024, he worked as a principal researcher at LSWare Inc., Republic of Korea, where he focused on research in copyright and privacy protection technologies. He is currently a postdoctoral researcher at Sangmyung University, Republic of Korea. His research interests include cryptography, data security, secure data sharing, cybersecurity, and space security.

Zhipeng Cai received his B.S. degree in Computer Science from Beijing Institute of Technology in 2001, his M.S. degree in Computing Science from the University of Alberta in 2004, and his Ph.D. degree in Computing Science from the University of Alberta in 2008. He is currently a Professor in the Department of Computer Science at Georgia State University and an Affiliate Professor in the Department of Computer Information Systems at the Robinson College of Business. He also serves as a Director at the INSPIRE Center. Dr. Cai's research expertise lies in the areas of resource management and scheduling, high-performance computing, privacy, networking, and big data. He is a recipient of the NSF CAREER Award.

Just Accepted