Progress Report

CS4681 - Advanced Machine Learning
Research Paper Assignment

Name:	Jayathilake N.C.
Index Number:	210257F
Research Centre:	Department of Computer Science and Engineering
Research Project Title:	EdgeMIN - Optimized Edge Deployment of MiniLM-Based Language Models
Primary Supervisor:	Dr. Uthayasanker Thayasivam

1 Project Review

This research project focuses on optimizing the deployment of MiniLM-based language models on edge devices through advanced compression techniques. The primary objective is to develop a comprehensive methodology that combines knowledge distillation, quantization-aware training, and structured pruning to create efficient, lightweight language models suitable for resource-constrained environments.

To date, the foundational literature review has been completed, identifying key compression techniques and establishing MiniLMv2 as the baseline approach. The theoretical framework has been developed, outlining a two-stage compression methodology. Current accomplishments include the comparative analysis of existing distillation methods and the design of the proposed optimization pipeline.

Outstanding aspects that require completion include: implementation of the MiniLMv2 distillation framework, development of the quantization-aware training pipeline, integration of structured pruning techniques, comprehensive evaluation on target edge hardware platforms, and validation across multiple downstream tasks. Additionally, access to high-performance computing resources for large-scale model training and a diverse set of edge devices for deployment testing will be essential for the project's success.

2 Literature Review

2.1 Introduction to Model Compression for Edge Deployment

Deploying large language models (LLMs) on edge devices remains an open challenge due to their immense parameter counts and computational demands. These models, such as BERT-Large and RoBERTa-Large, contain hundreds of millions or even billions of parameters, resulting in prohibitive memory footprints and inference latency for resource-constrained environments. To bridge this gap, the research community has converged on model compression as a critical enabler. Among the most promising strategies, knowledge distillation (KD)—where a smaller student model learns from a larger teacher—has been widely adopted for compressing Transformer-based architectures. While early KD methods focused on hidden-state alignment, recent advancements such as MiniLM shift the emphasis toward preserving self-attention relations, which are central to a Transformer's reasoning capability [1].

2.2 Early Knowledge Distillation Methods for Transformers

Traditional approaches like DistilBERT initiated the trend of task-agnostic distillation by combining objectives for language modeling, soft-target matching, and cosine similarity between intermediate hidden states [2]. While effective, such techniques often rely on rigid, layer-wise mappings between teacher and student, limiting flexibility under aggressive compression. TinyBERT improved on this by introducing a two-stage framework: general distillation followed by task-specific fine-tuning, with supervision at multiple representational levels (embeddings, attention matrices, logits). However, this still enforces explicit alignment across layers, constraining the architecture design space for the student [3]. Patient Knowledge Distillation (PKD) addressed this rigidity by supervising the student with carefully chosen intermediate layers from the teacher, employing strategies such as PKD-Last and PKD-Skip for deeper representations without overfitting early-stage patterns [4]. These works laid the groundwork for relational knowledge transfer but continued to prioritize hidden-state mimicry, which does not fully capture the inductive biases embedded in attention head interactions.

2.3 Advancements in Attention-Centric Distillation

MiniLM introduced a paradigm shift by targeting the essence of the Transformer: query–key (QK) attention distributions and value–value (VV) correlations. Instead of matching entire hidden states, MiniLM distills these relational structures via KL divergence, enabling the student to internalize token-level dependencies critical for language understanding without incurring the computational cost of full-layer replication [1]. A key advantage of this approach lies in its flexibility: students can adopt arbitrary hidden dimensions and head configurations, breaking free from the restrictive one-to-one mappings of prior methods. This architectural freedom is vital for real-world edge deployment scenarios, where hardware constraints dictate custom layer depths and head reductions. MiniLMv2 extends these ideas by generalizing the relation transfer across multi-head attention, distilling Q–Q, K–K, and V–V pairwise similarities from the teacher to the student. Importantly, it eliminates the alignment barrier of requiring equal head counts, unlocking even more aggressive compression opportunities without sacrificing semantic fidelity [5].

2.4 Recent Innovations in Knowledge Distillation

Recent research has continued to refine relational KD by addressing alignment bottlenecks and token redundancy. Techniques such as gradient-guided token pruning incorporate causal attention signals to dynamically prune less influential tokens during distillation, thus reducing computational complexity while preserving contextually critical information [6]. Other approaches, such as ARC (Attention Replacement Compression), selectively replace heavy layers with lightweight modules and use fine-grained attention distillation to ensure functional equivalence across layers [7]. Multi-level frameworks like MLKD-BERT combine feature-level and relation-level supervision, transferring both localized representations and global interaction patterns, resulting in students that are more robust under severe parameter reduction [8]. Complementary innovations—such as response-priming prompting during KD [9] and preference-based distillation for alignment (PLaD) [10]—demonstrate that KD is evolving beyond structural compression toward preserving reasoning behavior and instruction-following capabilities in compact models.

2.5 Hybrid Compression Techniques and Practical Deployment

Surveys on Transformer compression [11], [12] consistently highlight that attention-centric KD outperforms simple hidden-state regression because it retains the structural priors governing token-to-token interactions. Case studies on compressing BERT-scale models further reveal that compression must be hardware-aware, aligning tensor dimensions and precision formats with device-specific compute kernels to avoid inefficiencies [13]. This observation motivates hybrid pipelines that integrate KD with numeric optimizations like quantization and structural pruning. For instance, quantization-aware training (QAT) simulates low-precision arithmetic (e.g., INT8) during fine-tuning, enabling students to remain resilient to quantization noise [11]. Structured pruning, in turn, removes entire attention heads or feed-forward blocks instead of individual weights, producing dense, hardware-friendly models that exploit SIMD parallelism without specialized sparse kernels [14].

Taken together, the literature suggests a two-stage compression workflow for edge deployment. First, apply MiniLMv2-style relational distillation to craft a compact student that preserves attention-driven reasoning while adopting an architecture tailored to resource constraints. Next, incorporate QAT and structured pruning to further reduce memory footprint and latency without introducing unstructured sparsity that harms hardware efficiency. Empirical evidence shows that these stages are complementary: distillation preserves accuracy under extreme downsizing, while precision and pruning adaptations translate these gains into tangible throughput and energy savings on NPUs, CPUs, and even microcontrollers [1], [5], [3], [8], [14], [2], [15].

3 Methodology

3.1 Baseline Methodology: MiniLM

As our baseline, we make use of **MiniLM**, a lightweight framework that helps compress large Transformer-based language models into smaller, faster versions. The idea is to keep the important knowledge from the large model (teacher) while making the smaller model (student) efficient enough to be used in real-world settings where speed and memory are limited. The strength of MiniLM lies in three main ideas:

1. **Mimicking the Teacher's Last Self-Attention Layer**
   Instead of copying knowledge layer by layer, MiniLM teaches the student model to learn directly from the self-attention mechanism of the teacher's last layer. Since this layer contains the most meaningful semantic and dependency information, the student can capture what really matters without the overhead of complex mappings.

2. **Attention and Value-Relation Transfer**

   - **Attention Transfer** - The student learns by aligning its self-attention distributions with those of the teacher, using KL-divergence minimization.
   - **Value-Relation Transfer** - Beyond attention distributions, MiniLM also transfers the relations between values in the self-attention mechanism. This allows the student to capture richer word dependency information and removes the need for additional parameter mappings, enabling flexibility in hidden dimensions.

3. **Teacher Assistant for Smaller Students**
   When the final student model is much smaller, MiniLM uses a teacher assistant—an intermediate model that first learns from the teacher and then guides the smaller student. This step-by-step approach makes it easier for very compact models to still perform well.

3.2 Proposed Methodology

3.2.1 Overview

Deploying large language models (LLMs) on resource-limited edge devices is challenging due to their heavy memory and compute requirements. Our proposed methodology tackles this problem using a two-stage approach. First, we apply knowledge distillation to compress the teacher model into a smaller, faster MiniLM-based student model. Then, we refine this student model further through quantization-aware training (QAT) and structured pruning, ensuring that it runs efficiently on real-world edge hardware while maintaining high accuracy.

This layered approach allows us to balance performance with efficiency. Instead of relying on one technique alone, we combine distillation with precision and sparsity optimizations, leading to a model that is smaller, faster, and still accurate enough for real-time edge applications.

3.2.2 Comparative Analysis of Transformer Distillation

Several methods have been developed to distill large transformer models into smaller ones.

- **DistilBERT**: Reduced BERT's size by 40% with faster inference, but relied on a fixed layer-to-layer mapping.
- **TinyBERT**: Introduced multi-stage distillation (pre-training + fine-tuning), but required rigid alignment between teacher and student layers.
- **PKD (Patient Knowledge Distillation)**: Allowed flexible layer selection (e.g., last k layers), improving training efficiency.
- **MiniLM**: Focused on transferring self-attention knowledge from the last transformer layer, avoiding strict layer mappings and capturing richer dependencies.
- **MiniLMv2**: Generalized MiniLM by transferring relations across attention heads, allowing the student to have fewer heads than the teacher, enabling further compression.

From the comparison shown in Table 1 the key insight that can be taken is distilling *relational knowledge* (how tokens interact) is more effective than copying layer outputs directly. Methods like MiniLM and MiniLMv2 provide more flexibility, making them highly suitable for edge deployment.

3.2.3 Proposed Methodology

As shown in Figure 1, this methodology contains two main stages to progressively compress and optimize the model for edge devices.

Stage 1: Foundational Task-Agnostic Distillation with MiniLMv2
In the first stage, we distill a large pre-trained teacher model (e.g., BERT-Large or RoBERTa-Large) into a lightweight MiniLMv2-based student. Instead of enforcing strict layer-to-layer mapping, the process leverages *self-attention relation transfer*, where relational knowledge from a high-impact teacher layer (e.g., the 21st layer of BERT-Large or the 19th of RoBERTa-Large) is distilled into the student [5]. This approach enables the student model to have significantly fewer layers (e.g., 6 layers) and a smaller hidden dimension (e.g., 384), while maintaining strong reasoning and semantic understanding. Compared to earlier methods such as

Table 1: Comparative Analysis of Transformer Distillation Methods

Method	Target Knowledge	Loss Functions	Layer Mapping Strategy	Attention Head Flexibility	Key Contribution
DistilBERT	Final logits, intermediate hidden states	Language Modeling, Distillation, Cosine-Distance	Fixed, one-to-one	None (Same as Teacher)	Pioneered task-agnostic distillation; established a simple, effective triple loss objective.
TinyBERT	Embeddings, Hidden States, Attention Matrices, Logits	Multi-loss for each representation type	Fixed, explicit layer-to-layer mapping	None (Same as Teacher)	Introduced a two-stage learning framework for general and task-specific knowledge transfer.
PKD	Intermediate hidden states	Logit-based and hidden-layer-based	Flexible, selects layers via PKD-Last or PKD-Skip	None (Same as Teacher)	Demonstrated the value of "patiently" learning from multiple intermediate layers for incremental knowledge extraction.
MiniLM	Last layer's Self-Attention Relations	KL Divergence (Q·K and V·V)	Flexible, single last-layer transfer	None (Same as Teacher)	Shifted focus to a single, high-impact layer; introduced Value-Value (V·V) relation distillation.
MiniLMv2	Multi-Head Self-Attention Relations	Distills Q-Q, K-K, and V-V relations	Flexible, selects best teacher layer for transfer	Yes, student can have fewer heads	Generalized MiniLM; broke the same-head-count constraint for greater architectural freedom.
MLKD-BERT	Multi-level (Feature and Relation-level)	Six loss functions across layers	Flexible, maps multiple teacher heads to one student head	Yes, student can have fewer heads	Explored relation-level knowledge among tokens and samples; enabled flexible head number reduction.

TinyBERT [3] and PKD [4], MiniLMv2 offers greater architectural flexibility and supports compression across different numbers of attention heads. This creates a compact yet robust student model that preserves most of the teacher's accuracy.

Stage 2: Precision and Sparsity Optimization

With the distilled model as the foundation, the second stage further compresses and accelerates the network. First, *Quantization-Aware Training (QAT) [16]* is applied to fine-tune the model under low-precision arithmetic (e.g., INT8 or INT4), ensuring resilience to quantization effects and avoiding the significant accuracy drops commonly seen in Post-Training Quantization (PTQ) [17]. Next, *structured pruning* is performed to eliminate entire attention heads or neurons that contribute least to model performance, following saliency-driven criteria. Unlike unstructured pruning, structured pruning produces a dense, hardware-friendly architecture that can be executed efficiently on standard CPUs and NPUs.

By first distilling full-precision knowledge and then applying QAT and structured pruning sequentially, the methodology ensures that the compressed model minimizes information loss. Together, these techniques yield a highly efficient model with reduced latency, memory footprint, and power consumption, making it suitable for real-world edge deployment.

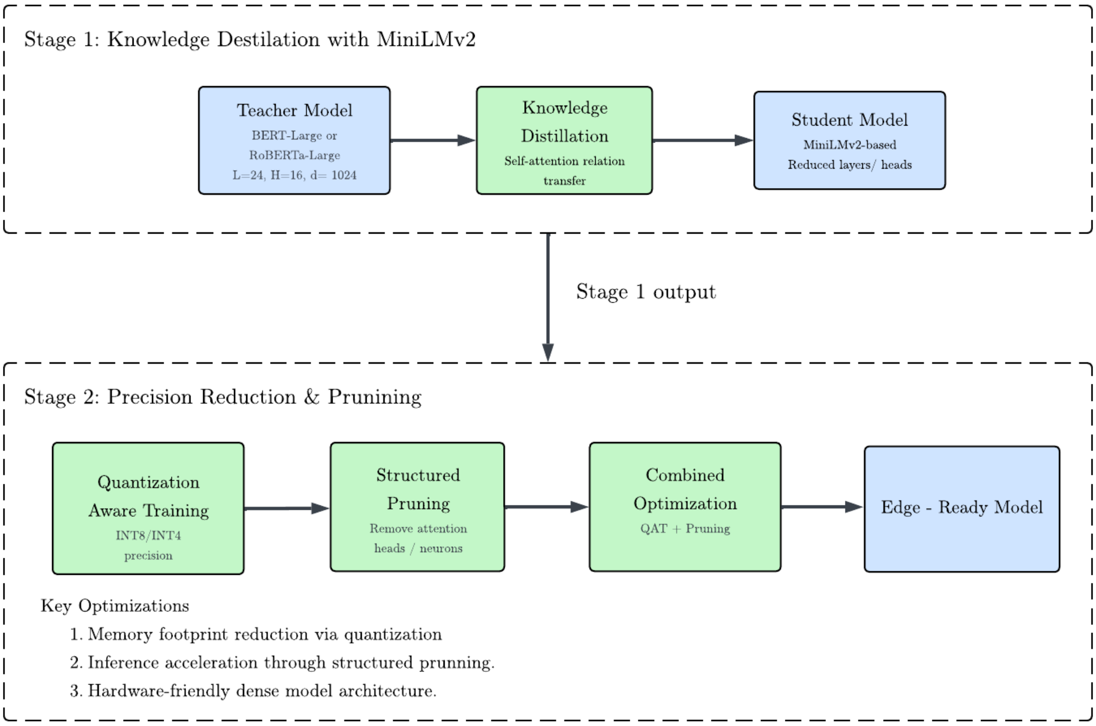


Figure 1: Overview of the proposed two-stage methodology.

3.3 Expected Outcomes

The proposed two-stage methodology comprising MiniLMv2-based [1] , [5] knowledge distillation followed by precision optimization and structured pruning is expected to yield the following outcomes,

1. **Compact and Efficient Model:** The distilled student model, derived from a large teacher model (e.g., BERT-Large or RoBERTa-Large), will feature fewer layers and attention heads, significantly reducing both memory footprint and storage requirements. Overall, the model size is expected to be reduced by 5–10× while retaining most of the teacher's performance.

2. **Preserved Accuracy and Language Understanding:** By emphasizing self-attention relation transfer, the student model preserves the teacher's reasoning capabilities and semantic understanding. Minimal degradation in downstream task performance is anticipated, with accuracy expected to remain above 95–99% of the teacher model.

3. **Faster Inference:** The combination of quantization and structured pruning is expected to accelerate inference, enabling low-latency responses on standard CPUs and NPUs, which is critical for real-time edge applications.

4. **Lower Power and Resource Consumption:** Reduced numerical precision (INT8 / INT4) and pruned network structures will lead to lower energy consumption, making the model suitable for mobile, embedded, and IoT devices.

5. **Flexible Deployment:** MiniLMv2 provides architectural flexibility, allowing the student model to have fewer layers or attention heads than the teacher. This enables deployment across a diverse set of edge devices with varying hardware capabilities.

6. **Robustness to Edge Constraints:** Structured pruning ensures that the model remains dense and hardware-friendly, compatible with standard inference engines. Furthermore, Quantization-Aware Training helps maintain model accuracy even under reduced precision, ensuring reliability on resource-constrained devices.

3.4 Datasets Used

Model / Method	**Pretraining Data**	**Distillation / Training Data**	**Evaluation Benchmarks**
MobileBERT [15]	BooksCorpus, Wikipedia	GLUE dev data (for ablation)	GLUE, SQuAD v1.1, SQuAD v2.0
DeCoT (Long CoT) [18]	– (LLMs pretrained separately)	NuminaMath (6K), R1 (16K self, 17K Bespoke-Stratos), QwQ (33K generated)	AIME2024, MATH500, GSM8K, OMNIMATH
DistilBERT [2]	Wikipedia, Toronto BookCorpus	–	GLUE, IMDb, SQuAD v1.1
Response-Priming Distillation [9]	–	GSM8K train split (1.3k–2.6k samples)	GSM8K test split
Internal Repr. (BERT6) [19]	– (BERTbase teacher)	–	GLUE (CoLA, QQP, MRPC, RTE)
Causal Attention Distillation [6]	– (LLMs pretrained separately)	NuminaMath-CoT (30k subsets), AceCode-87K (120k)	GSM8K, MATH, OlympiadBench, HumanEval+, LeetCode, LivecodeBench
MiniLMv2 [5]	Wikipedia, BookCorpus; RoBERTa/XLM-R sources	–	GLUE, SQuAD 2.0, XNLI, MLQA
MLKD-BERT [8]	Wikipedia, BookCorpus (via BERT-base)	–	GLUE, SQuAD v1.1, v2.0
SHD (Squeezing-Heads) [20]	BabyLM (LLM pretraining), ImageNet-1K (vision)	MiniLLM (DollyEval, SelfInst, VicunaEval, S-NI, UnNI)	SuperGLUE, ImageNet-1K
PLaD [10]	–	Anthropic-HH (58k), Reddit TL;DR (56k)	Anthropic-HH, TL;DR
ROSITA [14]	Wikipedia, BookCorpus (via BERT-base)	Augmented GLUE datasets (up to 8M+)	GLUE (CoLA, SST-2, QNLI, QQP, MNLI)
TinyBERT [3]	Wikipedia (2,500M words)	Large-scale text corpus, augmented task-specific data	GLUE, SQuAD v1.1, v2.0

Table 2: Summary of datasets used for pretraining, distillation/training, and evaluation across different knowledge distillation models and methods.

4 Limitations

The proposed research acknowledges several potential limitations and considerations,

4.0.1 Technical Challenges

The two-stage compression approach may introduce cumulative accuracy degradation, where losses from distillation compound with those from quantization and pruning. Careful hyperparameter tuning and validation will be essential to minimize these effects. Additionally, the optimal balance between compression ratio and performance may vary significantly across different downstream tasks and hardware platforms.

4.0.2 Hardware Dependency and Resource Constraints

Edge devices exhibit considerable heterogeneity in terms of computational capabilities, memory constraints, and specialized accelerators. Moreover, limited availability of GPUs and other high-performance hardware may restrict experimentation and evaluation, potentially impacting the depth and scale of the study.

4.0.3 Evaluation Complexity

Comprehensive evaluation across multiple downstream tasks, hardware platforms, and real-world deployment scenarios presents significant complexity. Ensuring fair comparison with existing methods while accounting for task-specific requirements will require careful experimental design.

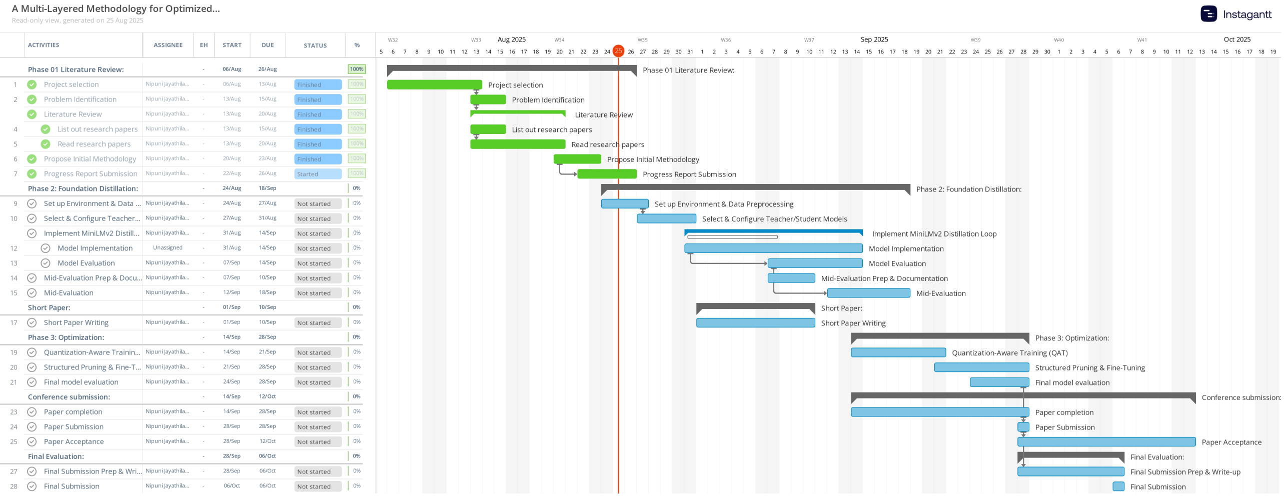
4.0.4 Ethical Considerations

The deployment of compressed language models on edge devices raises privacy and security concerns. While edge deployment can enhance data privacy by reducing reliance on cloud services, the reduced model capacity may impact fairness and bias mitigation capabilities. Additionally, ensuring that compressed models maintain appropriate safety guardrails and do not exhibit harmful behaviors requires careful validation.

5 Discussion

Despite the aforementioned limitations, the proposed methodology is designed to maximize the achievable outcomes within the available resources. Rigorous cross-validation procedures, comprehensive hardware profiling, and systematic evaluation protocols will be employed to extract the best possible performance from the compression approach. Collaboration with industry partners and access to diverse edge hardware, even if limited, will be leveraged strategically. Through these measures, the research aims to balance efficiency, performance, and ethical considerations while achieving meaningful insights from the methodology.

6 Timeline



A Multi-Layered Methodology for Optimized...

Read-only view, generated on 25 Aug 2025

	ACTIVITIES	ASSIGNEE	EH	START	DUE	STATUS	%
	Phase 01 Literature Review:		-	06/Aug	26/Aug		100%
1	Project selection	Nipuni Jayathila...	-	06/Aug	13/Aug	Finished	100%
2	Problem Identification	Nipuni Jayathila...	-	13/Aug	15/Aug	Finished	100%
	Literature Review	Nipuni Jayathila...	-	13/Aug	20/Aug	Finished	100%
4	List out research papers	Nipuni Jayathila...	-	13/Aug	15/Aug	Finished	100%
5	Read research papers	Nipuni Jayathila...	-	13/Aug	20/Aug	Finished	100%
6	Propose Initial Methodology	Nipuni Jayathila...	-	20/Aug	23/Aug	Finished	100%
7	Progress Report Submission	Nipuni Jayathila...	-	22/Aug	26/Aug	Started	100%
	Phase 2: Foundation Distillation:		-	24/Aug	18/Sep		0%
9	Set up Environment & Data ...	Nipuni Jayathila...	-	24/Aug	27/Aug	Not started	0%
10	Select & Configure Teacher...	Nipuni Jayathila...	-	27/Aug	31/Aug	Not started	0%
	Implement MiniLMv2 Distill...	Nipuni Jayathila...	-	31/Aug	14/Sep	Not started	0%
12	Model Implementation	Unassigned	-	31/Aug	14/Sep	Not started	0%
13	Model Evaluation	Nipuni Jayathila...	-	07/Sep	14/Sep	Not started	0%
14	Mid-Evaluation Prep & Docu...	Nipuni Jayathila...	-	07/Sep	10/Sep	Not started	0%
15	Mid-Evaluation	Nipuni Jayathila...	-	12/Sep	18/Sep	Not started	0%
	Short Paper:		-	01/Sep	10/Sep		0%
17	Short Paper Writing	Nipuni Jayathila...	-	01/Sep	10/Sep	Not started	0%
	Phase 3: Optimization:		-	14/Sep	28/Sep		0%
19	Quantization-Aware Trainin...	Nipuni Jayathila...	-	14/Sep	21/Sep	Not started	0%
20	Structured Pruning & Fine-T...	Nipuni Jayathila...	-	21/Sep	28/Sep	Not started	0%
21	Final model evaluation	Nipuni Jayathila...	-	24/Sep	28/Sep	Not started	0%
	Conference submission:		-	14/Sep	12/Oct		0%
23	Paper completion	Nipuni Jayathila...	-	14/Sep	28/Sep	Not started	0%
24	Paper Submission	Nipuni Jayathila...	-	28/Sep	28/Sep	Not started	0%
25	Paper Acceptance	Nipuni Jayathila...	-	28/Sep	12/Oct	Not started	0%
	Final Evaluation:		-	28/Sep	06/Oct		0%
27	Final Submission Prep & Wri...	Nipuni Jayathila...	-	28/Sep	06/Oct	Not started	0%
28	Final Submission	Nipuni Jayathila...	-	06/Oct	06/Oct	Not started	0%

Figure 2: Project Timeline and Key Activities

References

[1] 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,” *Advances in neural information processing systems*, vol. 33, pp. 5776–5788, 2020.

[2] 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.

[3] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, “Tinybert: Distilling bert for natural language understanding,” *arXiv preprint arXiv:1909.10351*, 2019.

[4] S. Sun, Y. Cheng, Z. Gan, and J. Liu, “Patient knowledge distillation for bert model compression,” *arXiv preprint arXiv:1908.09355*, 2019.

[5] W. Wang, H. Bao, S. Huang, L. Dong, and F. Wei, “Minilmv2: Multi-head self-attention relation distillation for compressing pretrained transformers,” *arXiv preprint arXiv:2012.15828*, 2020.

[6] Y. Guo, W. Yang, Z. Sun, N. Ding, Z. Liu, and Y. Lin, “Learning to focus: Causal attention distillation via gradient-guided token pruning,” *arXiv preprint arXiv:2506.07851*, 2025.

[7] D. Yu and L. Qiu, “Arc: A layer replacement compression method based on fine-grained self-attention distillation for compressing pre-trained language models,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2024.

[8] Y. Zhang, Z. Yang, and S. Ji, “Mlkd-bert: Multi-level knowledge distillation for pre-trained language models,” *arXiv preprint arXiv:2407.02775*, 2024.

[9] V. Goyal, M. Khan, A. Tirupati, H. Saini, M. Lam, and K. Zhu, “Enhancing knowledge distillation for llms with response-priming prompting,” *arXiv preprint arXiv:2412.17846*, 2024.

[10] R. Zhang, J. Shen, T. Liu, H. Wang, Z. Qin, F. Han, J. Liu, S. Baumgartner, M. Bendersky, and C. Zhang, “Plad: Preference-based large language model distillation with pseudo-preference pairs,” *arXiv preprint arXiv:2406.02886*, 2024.

[11] Y. Tang, Y. Wang, J. Guo, Z. Tu, K. Han, H. Hu, and D. Tao, “A survey on transformer compression,” *arXiv preprint arXiv:2402.05964*, 2024.

[12] C. Xu and J. McAuley, “A survey on model compression and acceleration for pretrained language models,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 9, 2023, pp. 10 566–10 575.

[13] P. Ganesh, Y. Chen, X. Lou, M. A. Khan, Y. Yang, H. Sajjad, P. Nakov, D. Chen, and M. Winslett, “Compressing large-scale transformer-based models: A case study on bert,” *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 1061–1080, 2021.

[14] Y. Liu, Z. Lin, and F. Yuan, “Rosita: Refined bert compression with integrated techniques,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 10, 2021, pp. 8715–8722.

[15] Z. Sun, H. Yu, X. Song, R. Liu, Y. Yang, and D. Zhou, “Mobilebert: Task-agnostic compression of bert by progressive knowledge transfer,” 2019.

[16] M. Chen, W. Shao, P. Xu, J. Wang, P. Gao, K. Zhang, and P. Luo, “Efficientqat: Efficient quantization-aware training for large language models,” *arXiv preprint arXiv:2407.11062*, 2024.

[17] H. Bai, L. Hou, L. Shang, X. Jiang, I. King, and M. R. Lyu, “Towards efficient post-training quantization of pre-trained language models,” *Advances in neural information processing systems*, vol. 35, pp. 1405–1418, 2022.

[18] Y. Luo, Y. Song, X. Zhang, J. Liu, W. Wang, G. Chen, W. Su, and B. Zheng, “Deconstructing long chain-of-thought: A structured reasoning optimization framework for long cot distillation,” *arXiv preprint arXiv:2503.16385*, 2025.

[19] G. Aguilar, Y. Ling, Y. Zhang, B. Yao, X. Fan, and C. Guo, “Knowledge distillation from internal representations,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 05, 2020, pp. 7350–7357.

[20] Z. Bing, L. Li, and J. Liang, “Optimizing knowledge distillation in transformers: Enabling multi-head attention without alignment barriers,” *arXiv preprint arXiv:2502.07436*, 2025.