Dynamic Gated Distillation: Dual-Teacher Framework for Jailbreak Prevention in LLMs with Preserved Reasoning

By

Rwittik Sarker
22101634
Nabil Walid Rafat
22101613
Anika Tabassum Omi
22101794
Ferdous Ahmed Auvi
22101583
Zabid Rahman
24141085

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
BRAC University
January 2026.

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at BRAC University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Rwittik Sarker

22101634

Ferdous Ahmed Auvi

22101583

Nabil Walid Rafat

22101613

Anika Tabassum Omi

22101794

Zabid Rahman

24141085

Approval

The thesis/project titled "Dynamic Gated Distillation: Dual-Teacher Framework for Jailbreak Prevention in LLMs with Preserved Reasoning" submitted by

1. Rwittik Sarker (22101634)
2. Nabil Walid Rafat (22101613)
3. Anika Tabassum Omi (22101794)
4. Ferdous Ahmed Auvi (22101583)
5. Zabid Rahman (24141085)

Of Fall, 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science in Fall 2025.

Examining Committee:

Supervisor:
Dr. Amitabha Chakrabarty

Dr. Amitabha Chakrabarty

Professor
Department of Computer Science and Engineering
BRAC University

Co-Supervisor:
Mr. Md. Tanzim Reza

Mr. Md. Tanzim Reza

Senior Lecturer
Department of Computer Science and Engineering
BRAC University

Program Coordinator:
Dr. Md. Golam Rabiul Alam

Dr. Md. Golam Rabiul Alam

Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
Dr. Sadia Hamid Kazi

Dr. Sadia Hamid Kazi

Professos
Department of Computer Science and Engineering
BRAC University

Abstract

Knowledge Distillation (KD) can be used to reduce large language models into smaller and more efficient models, although traditional knowledge distillation can preserve unsafe guidance on behavioral instructions, so small language models (SLMs) are vulnerable to jailbreak attacks. Moreover, using traditional knowledge distillation to finetune in terms of jailbreak robustness often degrades general reasoning performance. This creates a tough challenge to preserve jailbreak robustness as well as general reasoning at the same time. So often traditional knowledge distillation falls short to tackle two downstream tasks in one SFT. We introduce Dynamic Gated Distillation (DGD), a novel framework to which harm-score prediction has been added to enhance safety-utility alignment in the process of a dual teacher distillation. Comparative analysis shows that our DGD framework manages to retain almost all of its general alignment (MMLU score 42.13%) after distillation, compared with the base student model (MMLU score 42.81%), while reducing the attack success rate (ASR) below 5%, demonstrating the framework's effectiveness in achieving both safety and utility in small language models (SLMs).

Keywords: Knowledge Distillation, Gated weight, Jailbreak Prompts, LLM, SLM, MMLU, ASR

Acknowledgement

We would like to first express our gratitude to The Almighty God, without His blessing and guidance, we would not have managed to finish this project. We are extremely thankful to our esteemed supervisor, Dr Amitabha Chakraborty, whose exceptional guidance, thought provoking advice and unwavering support has been invaluable for the success of this research. We also thank and appreciate our co-supervisor, Md Tanzim Reza, whose thoughtful suggestions and inputs cannot be overstated. Finally, we would like to thank our RA, Azwad Aziz, for his continuous support and encouragement.

Table of Contents

Declaration i

Approval ii

Abstract iv

Acknowledgment v

Table of Contents vi

List of Figures viii

List of Tables ix

Nomenclature ix

1 Introduction 1
- 1.1 Background . . . 1
- 1.2 Rational of the Study or Motivation . . . 1
- 1.3 Problem Statement . . . 2
- 1.4 Objective . . . 3
- 1.5 Methodology in Brief . . . 3
- 1.6 Scopes and Challenges . . . 4

2 Literature Review 6
- 2.1 Preliminaries . . . 6
- 2.2 Review of Existing Research . . . 7
- 2.3 Summary of Key Findings . . . 14
- 2.4 Research Gap . . . 16

3 Requirements, Impacts and Constraints 18
- 3.1 Final Specifications and Requirements . . . 18
- 3.2 Societal Impact . . . 19
- 3.3 Environmental Impact . . . 19
- 3.4 Ethical Issues . . . 20
- 3.5 Project Management Plan . . . 21
- 3.6 Risk Management . . . 21
- 3.7 Economic Analysis . . . 21

4 Proposed Methodology **23**
4.1 Design Process or Methodology Overview 23
4.2 Preliminary Design or Design (Model) Specification 24
4.3 Dataset Creation 26
4.3.1 Training Dataset: 26
4.3.2 Test Dataset: 28
4.4 Implementation of Selected Design 28

5 Result Analysis **32**
5.1 Performance Evaluation 32
5.2 Analysis of Design Solutions 34
5.3 Final Design Adjustments 36
5.4 Statistical Analysis 36
5.5 Comparisons and Relationships 37
5.6 Discussions 37

6 Conclusion **39**
6.1 Summary of Findings 39
6.2 Contributions to the Field 39
6.3 Recommendations for Future Work 39

Bibliography **40**

List of Figures

4.1 Training architecture of the Prediction model 26
4.2 Training Pipeline of Dynamic Gated Distillation (DGD) 29

5.1 Training and validation loss curves for the prediction model across epochs 34
5.2 Training loss comparison: Dynamic KD vs Baseline KD 35

List of Tables

2.1 Summary of papers 14

5.1 Regression metrics of the prediction model across training epochs 34
5.2 Attack Success Rate (ASR) and MMLU Average Accuracy for Llama Based Models 36

Chapter 1

Introduction

1.1 Background

The rapid advancement of Large Language Models (LLMs) has fundamentally transformed the landscape of Natural Language Processing, yielding substantial improvements in reasoning, program synthesis, and generative creativity. Despite these advances, the practical deployment of state-of-the-art LLMs is severely constrained by their extensive parameterization, which incurs substantial computational cost and latency during inference. Knowledge Distillation (KD) has consequently emerged as a prominent model compression strategy, wherein a compact student model is trained to approximate the output probability distributions (logits) produced by a high-capacity teacher model. While conventional KD methods are effective in transferring linguistic competence and task-specific knowledge, they also risk propagating undesirable or harmful capabilities embedded within the teacher model. Specifically, the student may inherit latent behaviors related to the generation of toxic, illegal, or ethically problematic content, thereby introducing critical challenges for AI safety, alignment, and responsible deployment.

1.2 Rational of the Study or Motivation

Large language models (LLMs) are still susceptible to jailbreaks wherein the system attempts to evade rejection by taking advantage of its teaching-following behavior [60]-[20], despite days of fine-tuning and Human feedback-based reinforcement learning making them safer. Quick engineering, no-go optimization, and multi-turn coercion demonstrated that they are the only methods to predict unsafe behaviors in even model-aligned models, suggesting systemic weaknesses in contemporary alignment methods and not individual failures [21], [30], [34]. The risks are still greater because the application of LLMs in practice in application areas of safety-critical importance is becoming more and more common.

Small Language Models (SLMs) are even less efficient or economical on the size of models but even less resilient to jailbreak compared to larger models, which are preferable for efficiency and portability [73], [70]. Empirical research reveals that SLMs have poor representational ability and truncated alignment pipelines that limits representational ability and leads to brittle refusal properties and bad generalization in the presence of adversarial prompting [70], [65]. In addition, a large number of SLMs are trained through knowledge compressions or distillations, which may tend to enhance all safety vulnerabilities and elevate the likelihood of success in jailbreaks in comparison to their

teachers [65], [59].

Knowledge distillation (KD) has thus become a viable process of imparting closure and safety acts of large teacher models to small students [52], [55]. Although it is proved before that refusal patterns and safety considerate responses can be inherited to some extent by using KD, the vast majority of strategies are supported by using static loss weighting, where the risk is assumed to be the same to all prompts [55]-[14]. New dynamic KD algorithms have a better robustness as they vary the supervision according to the training dynamics or measurements of confidence [16], [39]-[19], and altering the supervision with additional or multiple teachers enhances the generalization further by using complementary sources of supervision [6]-[10]. Nonetheless, these approaches are prompt-agnostic because adaptation is functional of loss statistics or teacher uncertainty and not the semantic risk of loss posed by the input.

Simultaneously, harm-aware scoring models and prompt-level risk classifiers have demonstrated good performance in identifying unsafe or malicious inputs before generation [42], [29], also based on known language-model risks taxonomies [18]. However, these scores are mostly filtered or rejected or used as an evaluator, but these scores are not put in KD goals or employed to demand and adapt supervision in the training process [37], [64].

The reason why this study took place is because there was a void in the risk-sensitive, timely-conditioned distillation framework. We suggest that a learned HarmScore can be used as a continuous control signal replacing the predetermined KD weighting factor, and allow dynamically switching and balancing between supervision sources, such as teacher models, in a black-box distillation context. This design can directly overcome the shortcomings of existing KD that are static, prompt-agnostic adaptive systems, as well as existent dual-teacher models, and provides a principled approach to jailbreak alignment in SLMs.

1.3 Problem Statement

The conventional Knowledge Distillation (KD) techniques do not take into account safety during training. They consider the responses of all the teachers as being equal regardless of the input being a normal question or a harmful prompt. This brings about two issues.

Capability Leakage. The student model is informed by the responses of the teacher to the regular prompts [30]. Consequently, they imitate the teacher in unsafe behaviors that the teacher was not trained to perform. Binary Rigidity. The existing safety systems employ predetermined rules to prevent dangerous outputs [34]. This creates issues when the model is faced with ambiguous or out of the border prompts. The model tends to generate redundant refusals, incoherent text or unstable responses [34], which are painful to the user experience and training quality.

The crux of the matter is that the current static KD systems are not flexible to strike a balance between two objectives, being helpful and being safe during supervised finetuning (SFT), depending on the query student model gets. Thats why after SFT using KD still the student model gets high ASR in jailbreak attacks. We should have a way whereby the student model would be able to adjust its behavior based on whether the input is benign or potentially harmful. This issue is crucial in the development of supervised finetuned models which are competent and safe for downstream tasks.

1.4 Objective

The general aim of this research is to answer the question of core trade-off between model compression and safety alignment in distilled Large Language Models (LLMs). Since high-capacity models are squeezed into smaller, deployment-friendly variants to edge and resource-constrained environments, they often acquire the undesirable instruction-following properties of their teacher models with no proportional internalization of safety properties. Our goal is to develop such knowledge distillation paradigm, also known as Dynamic Gated Distillation (DGD), a dual teacher framework, where during training the student model will dynamically decide whether to learn from safety teacher or utility teacher. In methodology part we discussed a brief explanation about utility teacher and safety teacher. To operationalize this objective, the study aims at the following specific objectives:

Constructing a non-linear knowledge distillation framework:
This paper seeks to go beyond traditional hard-threshold safety enforcement mechanisms to propose an alternative differentiable, sigmoid-based gating mechanism in the distillation process. The end goal is to facilitate the student model to acquire a continuous and context-dependent profile of safety response, thus facilitating subtle behavior in the face of borderline, ambiguous, or sensitive cues.

General reasoning handling with dynamic knowledge distillation:
One of the main goals is to find and inhibit the probabilistic channels by which student models learn the ability to perform malicious or adversarial instructions during distillation. This paper will attempt to decouple general reasoning capability and harmful instruction-following behavior by dynamically manipulating teacher logits conditioned on prompt intent, without transferring unsafe capabilities, and without compromising on task-relevant intelligence.

Balancing the jailbreak vs general reasoning trade-off:
This study focuses on providing a formal maximization of the trade-off between safety and utility of alignment-sensitive training. This should be done by ensuring that the Attack Success rate (ASR) is reduced significantly when tested under adversarial jailbreak conditions, whilst ensuring that the performance on general-purpose reasoning and language understanding, as assessed by the Massive Multitask Language Understanding (MMLU) benchmark, does not differ significantly.

1.5 Methodology in Brief

This research employs a novel approach to Jailbreak alignment within the framework of Knowledge Distillation (KD). Our framework is structured around the development and evaluation of a context-aware distillation pipeline designed to unfold model capability in terms of reasoning and harmful intent.

Research approach: The approach started by developing a Teacher-Student architecture, employing a Llama-3.1 8B model as the source of teacher model and a Llama-3.2 1B as the target student model. The core of the framework lies in the implementation of a Dynamic Soft-Gated mechanism based on harms-core of each prompts that modulates the learning objective during training period.

Our training dataset consists of total 3100 prompts. Every prompts comes with instruction-response pair. The dataset includes an utility set of 1550 benign prompts curated from UltraChat[24] and Open-Orca[27] to preserve reasoning capabilities, and of 1550 jailbreak prompts from JailbreakV-28k[45] to preserve jailbreak understanding. To maintain a better generalization capability during training, we ensured to have a 50-50 distribution of benign and jailbreak prompts. During distillation training, we generated the utility responses by querying Llama3.1 8B for the benign prompts and for the safety set we queried Llama 3.1 8B with a refusal system prompt to generate refusal responses for the jailbreak part of the dataset.

Dynamic loss developing: The student model is optimized via a dynamic loss function. To obtain this, first we measured harmscore for each prompt of our training data using a Prediction Model which was a ROBERTA based language model. After that, we developed a Sigmoid Gating Function ($w(h)$) centered at a harmscore threshold ($h0 = 0.7$). For each training step, the pipeline calculates a weighted combination between the raw teacher logits (Utility) and refusal biased teacher logits (Safety). Also, Instruction-Aware Masking is applied to the Cross-Entropy loss, ensuring the model's parameters are updated exclusively based on the response tokens, rather than the prompts.

For high-risk inputs, the pipeline strongly favors refusal responses by applying a bias of +8.0 to each refusal token. At the same time, a per-sample dynamic weighting factor reduces the influence of helpful responses as the predicted harm level increases.

Analysis: We evaluated our framework accross two major benchmark metrices. The General Intelligence was evaluated on Massive Multitask Language Understanding (MMLU) accross 57 general subjects. The MMLU score has become a major indicator to showcase the general reasoning for any advanced language mode. For safety alignment benchmarking we attacked our distilled model with two jailbreak attacks, Pair[21], DSN[60] and calculated the ASR. Empirical results show that our dynamic gated distillation framework, reduces the ASR rate massively comparing with base student model.

1.6 Scopes and Challenges

The limitation of this study is the aspect of safety alignment in the context of the. certain architecture of transformer-based Knowledge Distillation (KD). This study is interested in optimizing Small Language Models (SLMs) that have 1B parameters. range, in particular based on the Llama-3 architecture. The enquiry is limited. to text-to-text models and measures adversarial jailbreak resistance. prompts in English. Moreover, the area of capability evaluation is limited. to the Massive Multitask Language Understanding (MMLU) benchmark, which gives. a surrogate of general-purpose reasoning and facts. Although this study is focused, there are a number of technical and theoretical issues were encountered.

Hardware and Memory Constraints: Distilling a single 8B-parameter teacher model to a student model on a single 24 GB GPU required substantial memory optimization. In particular, maintaining multiple logs simultaneously and handling VRAM allocation for long sequence lengths posed significant challenges. Careful memory management was therefore necessary to prevent out-of-memory (OOM) errors.

The Safety alignment vs General reasoning: The Trade-off between the Safety and the Alignment Tax was an optimization problem that was constant. Aggressive safety enforcement by hard logit-wiping would frequently lead to a degradation of language or mode collapse (repetitive stuttering). The attainment of soft-bias configuration that increased the Success Rate (ASR) of the Attack. General reasoning (MMLU) of model had to be carefully tuned with hyperparameters.

Chapter 2

Literature Review

2.1 Preliminaries

This part establishes the central concepts that are needed in the interpretation of the reviewed literature in Section 2.2 and putting the framework of proposed jailbreak-aware distillation into perspective. The alignment of modern Large Language Models (LLMs) is usually performed through instruction tuning and reinforcement learning with human feedback (RLHF) to decrease harmful behavior and remain helpful [30], [20]. Even with deployed systems, alignment is frequently operationalised with refusal policies and lightweight safety heuristics (e.g. intent detection and refusal templates) which might be effective with common harmful requests, but fail in the face of adaptive adversarial prompting [25], [54], [56]. The intrinsic conflict between the helpfulness optimization and safety enforcement is a common motivational point in recent research: well-constructed prompts can tip the system towards compliance despite internal policy to reject them [30], [20].

Jailbreak is an expediency-driven attack that causes an aligned model to breach its safety restrictions and produce disallowed content [60], [21], [30]. Jailbreaks take advantage of hierarchy manipulation of instructions (e.g. instruction override), semantic obfuscation, role-play framing, and multi-turn conversational pressure to compromise safety limits [28], [43], [56], [71]. In addition to the manual prompt engineering, there are also automated approaches like black-box iterative refinement of prompt and optimization-based adversarial suffixes that are also model-independent and allow jailbreaks to be made scalable and transferable [21], [38], [34]. Due to the possibility that the same model can be safe on direct harmful queries but be vulnerable on reformulated or indirect prompts, results in evaluation usually present attack success rate (ASR) and associated metrics of robustness, typical with standardized red-teaming benchmarks [46], [38], [54].

Small Language Models (SLMs) are small models with high efficiency in terms of latency, compute and cloud native deployability, yet are often not as robust to safety as larger LLMs [73], [70]. Such a weakness is explained by a smaller representational ability to encode a finer safety circumference and by costly alignment induction pipeline, which tend to be less profound than those employed to model frontiers [73], [70], [59]. Moreover, most SLMs are optimized through compression mechanisms like distillation and pruning; although they are utility-preserving, these mechanisms achieve refusal consistency and adversarial brittle behavior at the cost of an efficiency-safety trade-off, the fundamental cause of jailbreak resistance in practice [70], [65].

Knowledge distillation KD is also becoming popular in knowledge transfer not only in capability transfer but also in alignment behavior, such as calibrated refusals and harmless completions [52]-[55]. Distillation may be white-box (teacher access to logits/representations) or black-box (teacher only) and black-box KD could also be both practical with proprietary teachers but limited by the

absence of internal confidence signals [52], [49], [55]. KD objectives are normally a combination of a term matching the teacher (not necessarily divergence-based), and an ordinary supervised objective with a trade-off between them regulated by a weighting component globally [16], [14]. Reasoning fixed weighting is common but may be sub-optimal due to the fact that the optimum supervision balance depends on inputs and the training levels and as such, alternative divergence formulations and adaptive weighting programs were suggested [68], [22].
Dynamic distillation approaches also aim to learn supervision based on sample-wise reweighting, curriculum learning, or uncertainty/confidence gating to minimize the impact of noisy teacher signals along with enhance the robustness [16], [39]-[19]. Nonetheless, majority of the adaptation processes are carriers of optimization signals (loss, entropy, uncertainty), as opposed to direct safety risk that is posed by the semantic information of prompts. Simultaneously, harm-aware scoring has been created using prompt classifiers, safeguard datasets, and red-team based systems of toxicity based on structured taxonomies of the risk posed by LLM; such methods support filtering, rejection and evaluation pipelines, and can also estimate risk on the prompt level itself [42]-[18], [46]. Nevertheless, damages scores are seldom as continuous control variables within KD goals. The last proof is that multi-teacher and dual-teacher distillation demonstrate that teacher combination through weighting, multi-level guidance or selective transfer can enhance generalization and stability, which encourages multi-source supervision design [6]-[57]. Current multi-teacher models, but in general fail to explicitly compute jailbreak conditioned, prompt-risk conditioned teacher selection or weighting, which prompts the need to risk-conditioned distillation mechanisms in adversarial environment.

2.2 Review of Existing Research

Jailbreak
Large Language Models (LLMs) are often safely fine tuned and supervised by reinforcement learning with human inputs to limit harmful or forbidden outputs. Nevertheless, there is widespread evidence that such alignment can be circumvented using jailbreak attacks, in which malicious prompts can be used to coerce models to circumvent refusal behavior and agree to restricted requests [60], [21]. These attacks demonstrate the structural weakness in existing alignment mechanisms and are serious hazards in themselves, leading to the creation of illegal or unethical content [30].

Definition of Jailbreak Attack and Main Causes

Jailbreak attack plays off of LLMs' ability to follow instructions and make decisions in context and put more emphasis on adversarial instructions over internal safety constraints. Empirical studies have attributed this vulnerability to conflicting goals that models learn during training: models are optimized to be both helpful and safe, but this balance can be changed to one of compliance through adversarial prompts. [20]. Furthermore, many alignment systems are based on shallow heuristics, such as calling out the intent on the surface layer or the fixed template of decline, which can be bypassed by semantic reformulation or manipulation of the context [46], [25]. Another aspect of why jailbreaks succeed is when we analyze why the language models struggle to apply consistent boundaries in safety, in cases involving indirect form of phrasing, long context reasoning or conversational progression [28], [43]. According to these analyses, jail breaks are likely examples of constrained safety generalization, reflecting that the issue has been systemic rather than isolated in these safety generalization situations.

Prompt-Based Jailbreak Attacks

Initial jailbreaks were mostly based on hand-written prompts, such as instruction-override attacks, which directly request the model to understand rules as requested with additional role-play attacks, which incorporate malicious requests within a fictional or hypothetical context [38]. A very notable example is the "Do Anything Now" (DAN) family of prompts, which prompt an unrestrained persona, and are effectively tripping over refusal mechanisms [54]. Beyond single turns you have multi-turn coercion i.e. the attackers introduce malicious intent bit by bit, turn by turn in the dialogue. Even in the case of a model who initially rejects, by frequent rephrasing or by pressure of the situation [56] can yield compliance. Studies looking at how jailbreaks are constructed in real life also indicate that successful prompts will often use semantic obfuscation or indirect reasoning tasks or progressive disclosure of malicious intent, making them hard to detect for defenses based on isolated prompts (rather than conversational context) [72], [23].

Automated and Optimization-Based Jailbreaks

So far to overcome the limitations of the human-crafted prompts, automated generation of jailbreaks has been explored. These approaches make jailbreaking an adversary optimization problem such that they alter input prompts to maximize the chances of outputs that violate policies [34]. Gradient-based methods build adversarial suffixes that when syntaxed into user queries, probably draw unsafe behavior across models in alignment [71]. Although these attacks at the token level show a good transferability, they are often syntactically unnatural and will be computationally expensive. Zhou and Wang propose Don't Say No (DSN) which explicitly targets refusal behavior by jointly encouraging affirmative responses and suppressing refusal-related tokens and achieves higher attack success rates than previous methods [60]. In parallel, PAIR (Prompt Automatic Iterative Refinement) presents a black-box framework where an attacker LLM relies on refinement of the given prompts through feedback from the model acting as the victim yielding a high success rate with few number of queries, and generates semantically coherent jailbreak prompts [21].

Jailbreak Evaluation and Defense Context

Jailbreak success is not simple to evaluate. Simple refusal keyword matching is widely-used but vulnerable to false-positives and false-negatives [25], while more complicated evaluation systems involve semantic consistency tests and external LLM-based judges, and frequently ensemble tests in order to more faithfully estimate jailbreaks [60]. Defensive strategies typically belong to the following categories: prompt-level defenses (e.g., filtering and detection) and model-level defenses (e.g., refusal training and alignment tuning) [30], [46]. However, several defence techniques are empirically proven to be brittle and can be bypassed with either a multiturn or optimization-based attack [20], [56]. In general, jailbreak attacks are a constant and evolving threat in the literature, which motivates a desire for more dynamic and risk-aware alignment mechanisms.

Small Language Models (SLMs) and Jailbreak

Small Language Models (SLMs) have garnered much attention as efficient alternatives to the large language models, due to reduced computational cost, lower latencies and deployability on edge and constrained resource devices. Despite these benefits, it is now clear from recent empirical work that SLMs are especially prone to vulnerability against target jailbreak attacks, in fact they are often found to provide weaker resistance than their larger counterparts [73]. This kind of vulnerability is a critical challenge, as SLMs are increasingly used in actual applications with real users, where failing the safety testing can have immediate effects.

Large-scale evaluations demonstrate that a lot of SLMs are not robust in enforcing alignment constraints in adversarial prompting. Zhang et al. say that almost half of the evaluated SLMs have

high jailbreak success rates, and a significant percent pass even in direct harmful queries [73]. Compared with the behavior of large models, which, generally, have careful optimizations placed on the prompts that interfere with the safety mechanisms, SLMs often display unsafe behavior under comparatively simple attacks. This discrepancy is mainly attributed to the fact that SLMs have limited representational capacity in encoding the safety boundaries in various contexts [70].

A second contributing factor is inferior training of alignment. Unlike large language models, which are able to make the most of massive reinforcement learning with human or AI feedback, SLMs are frequently trained with light or truncated alignment pipelines for cost-related reasons [73], [59]. As a result, refusal behavior in SLMs tends to be brittle and inconsistent making it easier to override safety constraints using adversarial prompts (e.g. role-play or indirect reasoning). Moreover, SLMs are often running without supplementary moderation systems pre-integrated or post-generation filters especially in on-device environments thus lacking a secondary defense layer after the internal alignment failure [73].

Knowledge distillation goes even further to make SLM vulnerable. Many SLMs are obtained by applying distillation, pruning, or compression to larger models trained by their teacher to maintain its performance while decreasing its size [62]. While successful for efficiency, distillation can often compromise the safety reality with respect to the task efficiency, resulting into distillation induced vulnerability. Empirical analyses indicate that breakdown SLMs (distilled SLMs) may show surface-level imitation of their teachers, while losing strong refusal behavior, so that the jailbreak success rate for those is higher than that of their teachers [70], [52]. In some instances, unsafe behaviors are enhanced by distillation, in that decision boundaries that are relevant to safety collapse when optimization is applied.

These problems represent a more general efficiency-safety trade-off in the design of SLMs. Techniques such as aggressive pruning and parameter sharing reduce the redundancy which often is crucial to useful safety generalization [65]. While quantization has been demonstrated to give robustness in certain situations, other forms of compression decrease the reliability of the alignment significantly [70]. Additionally, defensive strategies that have been suggested for the defense of large models, like adversarial training or filtering via ensembles, are usually out-of-practice for SLMs because of memory and latency constraints [59].

Collectively, older attempts establish that SLMs represent a unique and weaker kind of threat surface in the jailbreak situation. Their small capacity, weaker alignment and vulnerability to distillation artifacts mean they are especially vulnerable to adversarial prompting [73], [44]. These findings provide motivation for SLM-specific alignment strategies that increase the resistance to jailbreak without compromising the efficiency for directly justifying the focus of this work.

Jailbreak Alignment Using Knowledge Distillation

Knowledge distillation (KD) has become a feasible way to transfer the capabilities of one large and well-aligned language model to another smaller student model, which is especially useful in resource-constrained environments. Beyond the compression of performance, there is recent work that also pursues KD as a tool of alignment and safety transfer, where students inherit refusal behavior, instruction-following discipline and ethical constraints from a teacher [52], [55], an aligned teacher. In such settings, the teacher model is usually trained with reinforcement learning and human or AI feedback defines the reference of behavior and the student is being optimized against the target of mimicking the outputs or preferences of the teacher, not just the supervision by ground truth.

Previous work shows that alignment relevant behaviors can be implicitly transferred with KD. Surveyed methodologies reveal that teacher generated responses are immersed with safety signals like

calibrated refusals and harmless completions which enable students to superimpose approximated aligned behavior even without the straight human annotation [52]. Safety-aware distillation frameworks provide an even stronger association bias that favors aligned responses or preference-ranked outputs which hurts the generation of harmful outputs, compared with naive KD [44], [55]. These ranked KD one of the potential KD-key elements of which is that it is a scalable alternative to expensive alignment pipelines, especially for small language models.

Alignment distillation techniques can be classified as white-box and black-box settings in general. White-box KD has access to the internal representations or logits of the teacher and enables the construction of a detailed alignment between token-level distributions or intermediate features. [55] Of course this is effective but is often impractical when distilling for proprietary LLMs. In contrast, black-box KD only uses teacher outputs which are accessed using APIs, so it is more realistic for safety transfer in deployed systems [52]. Black-box approaches often take the form of distilling instruction-following demos, explanations or ranked responses. However, insufficient knowledge of the teacher's internal confidence and doubt limits the fidelity of safety transfer, especially in the face of adversarial [49]. Nevertheless, current black-box safety distillation cannot condition the input-risk since only outputs (not logits) are accessible, which restricts the adaptive supervision due to jailbreak prompts.

Even with encouraging results, the current KD-based alignment approaches have significant limitations. First, most methods use static loss weighting which uses a static coefficient to balance the distillation objective and task loss during training [55]. This rigid formulation is based on assumption of equal importance for all prompts and do not take into consideration some variation of difficulty or safety risk. Second, most previous research takes a prompt-agnostic viewpoint, treating the benign and adversarial inputs equally in the distillation process. As a result, students can overfit on the common benign patterns, while being vulnerable to the jailbreak prompts which take advantage of the semantic ambiguity or edge cases [49]. Third, strong KL-based distillation objectives can lead to over-regularization or safety collapse where students fail to get generative diversity, become too conservative, or blindly inherit teacher biases [69], [55]. In some cases, improvements in alignment as measured by average benchmarks despair catastrophic failures faced by targeted attacks.

Collectively, previous studies form the basis of understanding the usefulness of KD as a powerful but imperfect tool for safety and alignment transfer. Static goals, homogenous prompt treatment and brittle generalization under unfriendly environments restrict the effectiveness of current methods. These shortcomings provide motivation for dynamic and prompt-aware distillation mechanisms that adjust the magnitude of supervision based on input characteristics and avoid over-regularizing the student–and provide the basis for the approach in this work.

Knowledge Distillation and Loss Optimization

Knowledge distillation (KD) traditionally trains a student model with both supervisions from ground-truth labels and from a teacher model. This guidance is usually achieved by encouraging the student to match the divergence of the output of the teacher using a divergence based on the divergence, most commonly Kullback-Leibler (KL) divergence [16], [14]. To reveal richer relational information about output classes, predictions of the teacher are often softened via temperature scaling, which approaches smooth the probability distribution and deliver more training signals besides hard labels [14]. In practice, the actual training objective is a combination of the distillation loss and the conventional cross-entropy loss expressed as a weighting factor, commonly denoted as a, which regulates the importance of teacher guidance during training [16].

The selection of the loss formulation and its weighting plays an important part in the effectiveness

of distillation. Early studies show that the performance of distillation is very sensitive to the trade-off between teacher supervision and ground truth learning, since too much reliance on one or the other signal can degrade student generalization [14]. A low distillation weight may lead the student to ignore useful knowledge provided by the teacher while an overly large weight may compel the student to overfit the teacher's outputs, inheriting its biases or errors [68]. Temperature scaling has additional interactions with this depicted balance, and higher temperatures drive the system to match the teacher's full distribution, while lower temperatures are becoming more like hard label supervision [14].

Despite such intricacies, most of the existing methods for determining the KD are dependent on a constant value which is found from some validation and remains constant over the training [16], [22]. This static weighting is based on an implicit assumption that all the training samples will be equally significant in the learning process and that there will be no optimum balance difference between losses according to the inputs or the stages of training. However, the empirical evidence is that this assumption is rarely true. Different samples may need different levels of teacher guidance, depending on their difficulty, ambiguity or novelty of distribution [22]. As a result, the choice of a (a singular parameter globally) often represents a sort of compromise choice that is not optimal for many samples.

Recent studies have exposed a number of limitations that result from this one size fits all optimization. First, static loss weighting is incapable of adjusting to input-specific risk or uncertainty, therefore static loss weighting is inappropriate for a safety-critical environment which requires adversarial or out-of-distribution prompts to be treated differently from benign inputs [68]. Second, the practice of uniform weighting does not address the changing needs of the student throughout training; early on there may be more valuable teacher guidance, whereas later on task-specific learning may take charge [22]. Third, fixed values are potentially worse with over-regularization, resulting in lack of diversity in output, over-conservatism, or a lack of robustness when the student needs to blindly imitate teacher behavior in all situations [66].

These problems have provided incentives for the study of alternative designs of loss and adaptive weighting schemes. Some works have been done to browse through the replacement of KL divergence by other objectives, which can be logit matching or modified divergence functions, to enhance optimization stability and robustness [14], [68]. Others suggest learning and/or schedule the distillation weight over time, to better balance supervision signals [22], [66]. While these approaches show improvements on static baselines, generally for most cases they are still prompt-agnostic because they treat them in a uniform way regardless of their semantic content or safety implications.

In summary, the literature has already determined that how loss is formulated and weighted plays a key role in the success of KD, but many existing methods have tended to use static, global optimization approaches. TThese limitations are particularly noteworthy in adversarial situations or jail break situations where the uniform weighting of losses doesn't represent the input specific risk. This observation is the motivation for dynamic and context-aware loss optimization, forming the groundwork to this work on the adaptive distillation framework.

Dynamic Defense Distillation

A rising boom of research has been conducted exploring dynamic and adaptive knowledge distillation (KD) strategies to improve robustness and generalization, instead of invariable supervision schemes. Unlike conventional methods of KD that utilize a fixed balance between cross-entropy and KL-divergence loss, dynamic distillation methods vary the strength or form of the teacher supervision based on the training dynamics, input characteristics or teacher confidence. These approaches arise from observation that the level of teacher guidance is useful at different samples and different

training stages [66].
One well-known direction is input-dependent loss weighting - that is the relative importance of the distillation loss will be adjusted at the sample level. Lu et al., RW-KD, which uses a meta-learning framework to learn sample-wise weights for the KD and task losses, which allows each input to be given a customized supervision balance [66]. This approach provides evidence that treating all samples equally is not the best approach, and that adaptive weighting invariably leads to better performance for students. Similarly, Ganguly et al. introduce AdaKD that dynamically scales the kl-divergence loss using the loss from the teacher to each input [39]. In this formulation, certain (confident) teacher predictions have more distillation emphasis and uncertain or more challenging samples have more reliance on ground truth supervision, thereby effectively implementing a curriculum-like strategy.
Another line of work involves applying a process of curriculum distillation in which the difficulty of the distillation signal changes over the course of training. Li et al. presents Curriculum Temperature Knowledge Distillation [17], where the temperature for softening the temperature of the teacher outputs is dynamically adjusted as the student gets better. By gradually reducing the difficulty of teacher supervision this method matches the process of distillation and the stage of learning of the student and provides improved convergence and generalization but fixed temperature baselines [17]. These curriculum-based approaches make it clear that the timing and strength of teacher guidance are important variables in strong student training.
Beyond scheduling and weighting, there are a number of studies that incorporate confidence aware or uncertainty aware supervision. Wen et al. provide some mechanisms to lessen the impact of weak teacher predictions by adjusting the distilling mechanism under low teacher confidence [9]. Their approach dynamically sharpens or suppresses teacher outputs depending on the prediction entropy which mitigates the noisy supervision. Guo et al. further introduce uncertainty awareness of KD, in which low confidence predictions from the teacher will be selectively masked when computing the KL divergence to emphasize the reliable supervision signals [41]. In multi-teacher settings, a Confidence-Aware teacher weighting is proposed by Zhang et al. where the contribution of each teacher is scaled depending on its reliability for a given input so that poor supervision is not allowed to dominate student learning [19].
Collectively, these methods prove dynamic KL-based supervision (such as adaptive weighting, scheduling the schedule of different curricula, or confidence gating) can make a significant difference to the robustness when compared to static KD. However, existing approaches have an important shortcoming - adaptation mechanism manufactures driven by training loss, uncertainty or confidence measures, not semantic or safety features of the input remain unexplored. None explicitly consider whether a prompt is potentially a jailbreak or adversarial query for the purpose of changing of supervision strength. As a result, dynamic distillation is prompt agnostic without such mechanisms to switch or modulate teacher influence depending upon input threat.
This gap motivates the need for jailbreak-aware dynamic distillation where strategies about how to supervise act not just based on how much things are learning, but also based on the level of danger in terms of prompts. Using this limitation directly bridges the dynamic Defence distillation to harm-aware alignment making the foundation for the proposed approach in this work.

Jailbreak Alignment Using HarmScore

The recent developments in the domain of LLM safety are more and more dependent on the harm-aware and risk-aware scoring systems to detect and prevent the unsafe behavior at the timely level. These techniques are used to approximate the probability that a particular input will produce harmful, toxic or policy violating outputs so that preventative safety measures can be taken before

generation. Conceptual foundations have provided a complete taxonomy of risks brought about by language models, such as toxicity, self-harm, facilitation of criminality, and misuse, which then drive the creation of automated systems to detect harm to apply alignment constraints [18].

One of the usual methods is to train harm classifiers which can directly be applied to user input. Gupta et al. present a supervised classification system, which classifies prompts into fine grained harm, including hate speech, violence and illegal activities [42]. Through red-teaming data and harmful question benchmarks, the models they build achieve high results on harmful vs. benign inputs, indicating that prompt-level risk can be predictably computed before the generation of responses. Equally, Wang et al. introduce the Do-Not-Answer dataset which includes high-risk user instructions that aligned models are expected to reject [29]. They provide experimental results showing that miniature classifiers trained on such data may detect unsafe prompts as well as large LLMs, which proves the accuracy of a low score on the scalability engine to verify safety measures.

In addition to using categorical classification, other works use toxicity or safety scores as continuous measures of risk. Corbo et al. produce EvoTox, an automated red-teaming system that involves the application of a toxicity classifier that quantifies the model outputs with numerical risk scores [64]. These scores are used in an evolutionary search process to produce successively more harmful jailbreak prompts in EvoTox and these results reveal how toxicity measures can be used to quantify and maximise unsafe behaviour. These types of scores (typically obtained via applications such as the Perspective API) are generally applied in moderation and evaluation pipelines to rank, filter, or block unsafe content [64].

During training as well as fine-tuning, risk-aware scoring has been investigated. Choi et al. suggest a safety-conscious fine-tuning architecture that learns harmfulness scores of training samples and removes or weighs up samples that are very unsafe during model adaptation [37]. Their findings indicate that the negative outputs are indeed decreased, which proves that the harm scores may be one of the useful indicators to use in order to align the data curation. In these techniques, however, harm scores are used merely as extraneous heuristics, to either provide a filter on the data, or to provide a refusal at inference time.

Regardless of these improvements, the current harm-conscious approaches have significant drawbacks in common. First, the harm scores are mainly applied on the filtering, rejection, or evaluation, as opposed to the harm scores as a part of the optimization objective of the model [42], [29], [64]. Second, harm scores affect data selection instead of the learning process of the model itself even when they are implemented in training pipelines [37]. Importantly, no previous literature uses harm or risk scores in the loss term of knowledge distillation, or operate them to balance dynamically supervision clues with timely risk. Consequently, the current distillation-based alignment models are prompt-agnostic, which means that they prepare benign and adversarial inputs equally when training.

This gap is the incentive of our approach. Directly incorporating a HarmScore into the distillation process will not only be moving beyond the mere static filtering, but also allow timely risk-conditioned supervision. This enables the student model to adaptively modify the manner in which it learns in response to various teachers or goals when faced with the threat of jailbreak prompts, which gives a principled method of balancing safety and utility in adversarial environments. As none of the previous papers treat harm scores as a continuous control signal within KD loss so as to dynamically balance supervision in black-box distillation.

2.3 Summary of Key Findings

Table 2.1: Summary of papers

Paper	Model	Target	Strategy	Metrics	Result
[60]	LLaMA-2-Chat-7B; Vicuna-7B-v1.3	Safety alignment Break refusal behavior and coerce harmful compliance	A jailbreak that is based on optimization and boosts affirmative responses and inhibits refusal-related tokens	ASR (%)	LLaMA-2-Chat-7B, the Mean ASR is 47.7% and best run 74% (baseline GCG best 43%) and in Vicuna-7B-v1.3, the Mean ASR is 57.1% and best run 83% (baseline GCG best 52%)
[21]	Victims: Vicuna-13B, LLaMA-2-7B-Chat, GPT-3.5, GPT-4, Claude, Gemini	Black-box jailbreak of aligned LLMs	Iterative prompt refinement using attacker feedback from target responses (query-efficient)	Jailbreak (%), Success	In Vicuna-13B, GPT-3.5, GPT-4 and Gemini-Pro jailbreak percentage is respectively 88% , 51% , 48% and 73%
[30]	GPT-3.5-Turbo, GPT-4, Claude (early)	RLHF safety failure modes	Systematic jailbreak prompt design + empirical probing	ASR (%), refusal rate	In GPT-3.5 the score of ASR is 70% and in GPT-4 the ASR is between 30–40%
[46]	GPT-3.5-Turbo, GPT-4, Claude-2, LLaMA-2-Chat-7B, Vicuna-13B	Standardized evaluation of jail-break/harm robustness	HarmBench benchmark + automated red-teaming protocols	ASR (%), harm coverage	In Vicuna-13B, LLaMA-2-Chat-7B, GPT-4 the score of ASR is respectively 90% , 70–80% and 20–30%
[72]	GPT-3.5-Turbo, GPT-4	Defend against jailbreaks at inference	JBShield: activated concept analy-sis/manipulation to suppress jailbreak concepts	ASR (%)	In GPT-3.5 the score of ASR reduces from 55% to 25% and in GPT-4 the ASR is reduces from 30% to 12-15%
[73]	Phi-2, TinyLLaMA-1.1B, Qwen-1.5-4B, LLaMA-2-7B-Chat	Jailbreak robustness of SLMs	Large-scale evaluation across prompt/multi-turn/optimization attacks	ASR (%)	In Phi-2 the score of ASR is 55–65% and in TinyLLaMA the ASR is between 60–70% and lastly in LLaMA-2-7B it is around 35–45%

Paper	Model	Target	Strategy	Metrics	Result
[70]	Phi-2, TinyLLaMA-1.1B, Qwen-1.5-4B, LLaMA-2-7B-Chat	Hidden jailbreak risk from efficiency-oriented SLM design	Empirical analysis vs capacity/alignment depth/compression	ASR (%)	In Phi-2 the score of ASR is 60–70% and in TinyLLaMA the ASR is between 65–75%
[65]	Teacher: LLaMA-2-Chat-7B; Students: Phi-2, TinyLLaMA-1.1B, Qwen-1.5-4B	Transfer stealthy jailbreaks from LLMs to SLMs	Adversarial Prompt Distillation (prompt transfer + stealth)	ASR (%), detectability	In Phi-2 the score of ASR is 70–80% and in TinyLLaMA the ASR is between 75–85%
[62]	GPT-3.5-Turbo, GPT-4	Low-effort jailbreak via natural interaction	SPEAK EASY: polite/incremental conversational elicitation	ASR (%)	In GPT-3.5 the score of ASR is 65–75% and in GPT-4 the ASR is between 30–40%
[49]	LLaMA-2-Chat-7B, Vicuna-7B	Improve teacher–student distribution matching in KD	Distribution alignment constraints during distillation	Accuracy (%), KL-divergence	The accuracy gain increased from 3-6%
[16]	ResNet-32, ResNet-110	Sample-wise adaptive KD	RW-KD meta-learning reweighting of KD vs task loss	Top-1 Accuracy (%)	The Top-1 accuracy gain increased from 1.5–3.0%
[66]	ResNet-18, ResNet-50, ViT-B	Context-aware adaptive KD weighting	Adjust KD loss weight by sample difficulty/context	Top-1 Accuracy (%)	The Top-1 Accuracy gain increased from 1.2–2.8%
[42]	RoBERTa-base, DeBERTa-v3-base (classifiers); eval: GPT-3.5-Turbo, GPT-4	Detect harmful prompts pre-generation	Supervised prompt-level harm classification	Accuracy (%), F1 (%)	F1-score is 88–92% on harmful-prompt detection

Paper	Model	Target	Strategy	Metrics	Result
[29]	GPT-3.5-Turbo, GPT-4; RoBERTa-base (classifier)	Evaluate refusal safeguards on unsafe prompts (DNA)	Do-Not-Answer dataset + refusal/detection evaluation	Refusal Accuracy (%), Detection Accuracy (%)	Refusal Accuracy is 85–90% and 65–75% respectively in GPT-4 and GPT-3.5 and Detection accuracy is 80–85% in RoBERTa
[6]	CNN student + multiple CNN teachers	Multi-teacher supervision to improve generalization	Weighted aggregation of soft targets from multiple teachers	Accuracy (%)	Accuracy increased by 1–3%
[12]	CNN student + hierarchical multi-teacher supervision	Multi-level feature/output distillation with multiple teachers	Adaptive weighting across teachers and network layers	Accuracy (%)	Adaptive multi-teacher KD improves robustness by increasing resulting accuracy to 2–4%
[10]	CNN/RNN students + collaborative multi-teacher setup	Improve learning via cooperative staged supervision	Collaborative teaching across different learning phases	Accuracy (%), convergence	Accuracy increased by 2%, faster convergence
[57]	Vision–Language student + dual teachers	Continual learning with reduced forgetting	Selective dual-teacher distillation by task relevance	Accuracy (%), forgetting rate	Accuracy increased by 3–5%, so it reduced forgetting

2.4 Research Gap

- **The KD optimization objective generally does not involve harm, or risk scores.** Most of the previous safety studies use harm classifiers to filter, trigger refusal, or even to evaluate, as opposed to risk estimates as continuous control measures in the distillation loss
- **Safety transfer in a globally adaptive or static KD weighting case is predominant.** The majority of alignment-based KD models follow the assumption that prompts have the same degree of importance, which is less than optimal in adversarial environments where the risky prompts have to be supervised differently.
- **The Black box safety distillation does not have conditioning of input-risk.** Although black-box KD can be implemented in practice with proprietary teachers, recent KD methods solely use teacher outputs and do not scale strength of supervision with the timeliness of risk of a safety incident.

- **A large number of jailbreak defenses are external or prompt-time instead of distillation-time safety transfer.** Most of the defenses are based on filtering, refusal templates, or detectors at inference time. Adaptive prompting bypasses these and therefore do not make the student model inherently robust.
- **Jailbreak resilience to SLM through KD has not been studied extensively.** Small language models are empirically proven to be more susceptible to jailbreak attacks, but it is not explicitly stated within the KD and multi-teacher methods that to what extent they pursue safety generalization per SLMs efficiency criteria.
- **Multi-teacher KD does not generally model itself into risk-conditioned safety-utility balancing.** Multi-teacher and dual-teacher KD enhance generalization, although they do not offer a mechanism to alternate or combine teachers based on immediate danger (e.g. more intense safety guidance on high-risk prompts and more advantagous guidance on benign prompts).
- **Usefulness vs over-blocking is not a concern that is easily managed in a risk-based manner.** There are those defenses that drive models to conservative refusal. There are few methods that uphold such a continuous mechanism that borderline prompts are not undertaken in the same way as a prompt that is obviously malicious as training goes on.

Chapter 3

Requirements, Impacts and Constraints

3.1 Final Specifications and Requirements

Teacher-Student Model Pair:
The methodology of this framework required two-teacher distillation approach requiring a high-parameter Llama-3.1 8B model to serve as the utility teacher and a safety teacher. Moreover, this framework needed a Llama 3.2 1B to act as a student model. Llama 3.1 8B is an effective teacher model with a large number of parameters, performance on a variety of tasks and high-quality reasoning abilities in terms of benchmark scores, whereas Llama 3.2 1B is the student model, being small enough to be deployed under resource-constrained conditions and with reasonable reasoning skills [47]. The ratio between sizes (8:1) is sufficiently large to offer a capacity difference worth training on knowledge transfer, specifically, in transferring reasoning patterns between the teacher at higher performance level. In other words these two are strong state of the art models that are open sourced and can be deployed in edge devices [47]. That's why we choosed this two model for our research.

Judge Model:
We have chosen to use the Llama-3.3-70B-Instruct model loaded from Ollama interface as our judge model for the entire evaluation process. This model has been found to be a robust judge due to its scale and instruction tuning, which helps in the precise interpretation of the evaluation rubrics for both regular and jailbreak prompts[48]. Previous research has shown that large instruction aligned language models have high level of understanding with human evaluation when used as judges. For the cases of jailbreak benchmarks, the use of 70B paramaeters llama models has low false-negative rates and are generally stable compared to smaller models[67]. This helps in the reliability of the evaluation of subtle policy violations which is very crucial for us while evaluating the jailbreak prompt's responses. The open-weight characteristic of this model also helped us in the reproducibility of the evaluation in the context our research.

Dataset:
One of the most important requirement we had to ensure that the dataset maintain a 50:50 ratio between benign prompts and harmful prompts to maintain generalization capability between general reasoning abilities and safety alignment. A pre-trained RoBERTa-base classifier, which has been trained with a two-stage transfer learning task (Jigsaw toxicity classification followed by spectrum regression) is also necessary to give the high-fidelity harmscore signals to each and every training prompts that drive the gating logic.

Hardwares and Libraries:
The computational and resource demands required a HPC environment with an RTX 4090 of 24 GB VRAM, to allow concurrent memory usage of the teacher logits, training and backpropagate gradients. The numerical stability and the memory efficiency of all the models need to be achieved with the bfloat16 precision. Other necessary libraries such as, PyTorch, Huggingface, Transformer, Pandas, lm-evaluation-harness was needed to train and to benchmark. Lastly, a local Ollama server with Llama-3.3 70B needed to serve as an instruction-following deterministic judge to validate the Attack Success Rate (ASR) at the evaluation phase.

Summary of Requirements:

- **Model:** Llama 3.3 70B Instruct, Llama 3.1 8B Instruct, Llama 3.2 1B Instruct
- **Server:** Local Ollama Server
- **Libraries:** PyTorch, CUDA, Huggingface Transformers, Pandas, and `lm-evaluation-harness`
- **Environment:** Virtual Environment (venv)
- **OS:** Windows 11
- **CPU:** i9-14900K
- **GPU:** RTX 4090 Super 24GB
- **RAM:** 64GB DDR5
- **SSD:** 1 TB

3.2 Societal Impact

According to [58], Jailbreak attacks on large language models (LLMs) pose a growing societal threat as they enable the circumvention of safety mechanisms, allowing models to produce harmful, unethical contents. These attacks have become increasingly sophisticated, enabling adversaries to generate malicious content including misinformation, cyber-attack tools, and explicit instructions for criminal activity by exploiting weaknesses in even the most advanced LLMs. Moreover, jailbreak attacks can manipulate LLMs to produce content that promotes misinformation, hate speech, or instructions for illegal activities, such as malware distribution or fraud. This vulnerability threatens public safety and trust in AI systems [33]. So the research of defending jailbreak attempts in LLMs in a robust way marks great societal impact. Crafting a proper supervised finetuning method in such way that any smaller model could easily be implemented in any small device that could effectively ignore any jailbreak attemps while maintaining general intelligence [33]

3.3 Environmental Impact

Large language models are increasingly causing issues with their environmental impact because of the high computational cost of training, activating, and deploying them, this contemporary approach to safety alignment usually is based on reinforcement learning involving human feedback, massive

red-teaming, and frequent steps of adversarial evaluation, which highly increase energy usage and carbon emissions [20]. Such expenses are escalated further when the safety measures have to be re-implemented again and again through various models or revised in keeping with the changed jailbreak patterns [56].

The suggested concept of Dynamic Gated Distillation (DGD) framework has a positive contribution to environmental sustainability since it allows knowledge to be transferred effectively in terms of safety transfer. Rather than using continuous inference-time moderation, extrinsic filters or repeated alignment steps, it uses DGD to move the safety and refusal behavior of a large capacity teacher model to a small student model in a single training stage. It designs the repeated inference-time computation to the training time reducing the inference cost and enhancing long-term efficiency of the environment [52], [55].

The framework also saves further environmental overhead by concentrating on Small Language Models (SLMs). The existing literature indicates that SLMs need significantly lower memory, computing, and energy costs of inference than large language models, thus they could be deployed on edge devices and resources intensive settings [70], [73]. When safety-aligned through distillation, they can work without cloud-scale infrastructure and reduce operation carbon footprint, as well as energy demand.

Besides that, making jailbreak resistance central to the distilled student model enables jailbreak robustness with less cave-in adversarial retraining or patching during emergencies, which are invariably induced by an identified jailbreak method [71]. The cost of maintaining a safe model in terms of lifecycle energy cost is majorly cut by reducing occurrences of such costly retraining cycles.

In general, the suggested framework indicates the same aspect of safety goals achieving environmental sustainability through the efficient transfer of safety, reduced dependence on inference infrastructure on large scale, and the ability to deploy robustly on-devices. This design promotes the creation of responsible AI systems which are secure, efficient and energy efficient, besides being environmentally friendly.

3.4 Ethical Issues

The primary ethics threat in this project is dual-use as this project collaborates directly with harmful and jailbreak-style prompts. The knowledge that can be used to minimize jailbreak success can also be utilized by an individual to create better jailbreaks. To minimise that risk, we consider any jailbreak prompts as restricted research data, we do not publish prompt lists that are ready-to-attack, and only publish results in aggregate (such as overall Attack Success Rate and not prompt-by-prompt failures).

Second, there is exposure to harmful content. The training and testing sets can contain some prompts, which can be violent, hate, unlawful, or explicit. Our three techniques to reduce exposure are (1) maintaining a separation of the harmful datasets and general datasets, (2) ensuring that the data to which the team has access are only the resources necessary to aid evaluation, (3) scoring fully and using automated scoring and refusal-oriented teachers such that the human does not need to read full harmful generations to assist with debugging.

Thirdly, there is a fairness and over-blocking danger. A safety system can be too conservative and deny benign requests that contain sensitive keywords (such as hack in an education on cybersecurity situation). Our design takes this into consideration with a continuous harm score between 0 and 1, and based on smooth gating, such that borderline prompts are not handled similarly to evidently malicious ones. We also require manual spot-checking of benign topics in order to make the model useful.

Fourth, there are issues of privacy and data handling. Personal data should not be included in any of the prompts obtained. Storage of logs and any results of the evaluation should be done in a secure manner and any output should be filtered prior to sharing. Lastly, there must be accountability: even when the distilled student is put on-device, it must have usage policies and monitoring options (where feasible), and a clear user message when it does not.

3.5 Project Management Plan

The thesis project has been implemented in the duration of 4 months or so, using the iterative process of research-and-engineering. The work breakdown schedule of the plan below gives a summary of work breakdown, milestones, tooling and allocation of resources to deliver a reproducible training and evaluation pipeline.

Work Breakdown and Schedule

- **Phase 1 - Problem formulation and literature review (Weeks 1–2):** Establish the threat model and assessment criteria (e.g., attack success rate), examine what has been done previously on jailbreak attacks and defenses, and complete the experimental hypotheses.
- **Phase 2 - Data collection and preprocessing (Weeks 3–4):** This is by curating benign and harmful prompts, establishing a target benign ratio: harmful ratio, cleaning/deduplication, and labeling/scoring validation.
- **Phase 3 - Baselines and infrastructure (Weeks 5–6):** Set up the training code base, model checkpoints and experiment tracking. Also Set up a template code base for reproducibility. Construct the reproduction of baseline student performance and of baseline safety metrics.
- **Phase 4 - Method implementation (Weeks 7–10):** Implementing the dual-teacher distillation framework, logits level manipulation for adding bias, developing harm score gating mechanism, creating the dynamic KD loss and final total loss.
- **Phase 5 - Training, tuning & benchmarking (Weeks 11–14):** Full-fledged training experiments, changing the hyperparameters and benchmarking the distilled student model.
- **Phase 6 - Analysis and thesis writing (Weeks 15–16):** Summarize findings and produce numbers/tables, record constraints and ethical implications and complete thesis and supporting writing.

3.6 Risk Management

While our framework significantly reduces jailbreak success rates, absolute robustness against evolving adversarial techniques cannot be guaranteed. Continuous updates and monitoring are needed to stay effective against new attacks.

3.7 Economic Analysis

The practical value of this work is that it enables the creation of jailbreak-resistant language models in small scale. A smaller model of student (approximately 1B parameters) costs less to operate than

an 8B model of teacher, and can be implemented on cheaper servers (or even on locally available machines). This cuts down on the inference cost incurred repeatedly, minimizes the latency and does not need to pay per-request API fees when a proprietary teacher is not needed at runtime.
The training cost is a fixed cost. We trained a 24GB single-gpu and a comparatively small training corpus (3100 prompts to be distilled and 6315 to predict the harm-score) in our arrangement. This makes compute requirement of a student project more affordable than large-scale safety tuning which can need a number of GPUs and much bigger datasets. Costs associated with the principal cost items are: (1) GPU time, (2) datasets and logs storage and (3) engineering time to construct and test the pipeline.
The advantage can be quantified in two numbers: the student records low jailbreak and the general reasoning only slightly higher than the base student. This minimizes the "safety incidents" (which have actual operational cost) but does not make significant decline in utility, which would reduce product value. In addition, since the framework is black-box friendly, it can utilise the results of the teacher via APIs during training but needs not the teacher during deployment. That converts the continuous inference cost to a constrained training cost which in most cases is cheaper in the long run when the system is used by many people.
In general, the cost-benefit trade is positive when (1) the application requires numerous inferences, (2) the latency is important, and (3) the failure of safety is costly (both, reputationally and operationally).

Chapter 4

Proposed Methodology

4.1 Design Process or Methodology Overview

Usually we use Knowledge Distillation for compressing a larger language model into a smaller language model. Firstly, the large model here is called the teacher model and the smaller model here is called the student model, which learns from the teacher model. But when the student model learns, it also learns the adversarial vulnerabilities of the teacher model. We have used Knowledge Distillation to make our student model, Llama 3.2 1B more robust and efficient against jailbreak prompts. We propose a modified distillation pipeline that produces a Llama 3.2 1B student model with improved computational efficiency and enhanced robustness to harmful prompts.

Our work aims to transfer the reasoning and linguistic capabilities of the Llama-3 8B teacher model to a compact Llama-3.2 1B student model through knowledge distillation. In addition to preserving core performance, we focus on improving robustness against jailbreak attacks by reducing the Attack Success Rate (ASR) to below 5–10%. At the same time, we seek to minimize the alignment tax, ensuring that the imposed safety constraints do not substantially degrade the student model's general intelligence or reasoning ability.

We have implemented a dynamic gated distillation with dual teachers (Utility teacher and Safety teacher) to accomplish the above mentioned objectives. To choose between the dual teachers, we calculated a harm score of each input in the training dataset, based on which our pipeline dynamically decides which teacher to learn from statistically. This dynamic distillation was chosen over a binary approach so that the pipeline can perform a continuous, smooth, optimizable balance between safety and utility rather than abrupt binary filtering.

The development of this pipeline was guided by three key considerations:

1. **Hardware Limitations:** The distillation needed to remain feasible on consumer-grade hardware, specifically under a 24GB VRAM constraint.

2. **Alignment Trade-off:** A careful balance was required between enforcing safety constraints and preserving general-purpose intelligence, acknowledging the unavoidable alignment tax.

3. **Linguistic Coherence:** It was essential to maintain fluent and natural language generation while avoiding over-refusal, where the model becomes overly cautious and declines benign requests.

System Overview
The proposed methodology is implemented through three-stages:

1. **Harm Scoring Phase:** Development of a Prediction model to generate a continuous safety score for every training sample.

2. **Target Distribution Construction:** Curating the teacher's logits to create a virtual "Safety Teacher" that prioritizes professional refusals.

3. **Gated Distillation Loop:** A unified training process where the student model optimizes against a weighted interpolation of both teachers, conditioned on the safety signal.

4.2 Preliminary Design or Design (Model) Specification

Multiple candidate models were explored for this research. At first, a binary switching between dual teachers were considered, but because it causes stuttering and sudden gradient jumps, we discarded this idea. Instead we switched to this dynamic dual teacher method. For this, we first need to build a robust Prediction model that can assign a continuous harm score between 0 to 1 range.

For the Prediction model, we used RoBERTa(Robustly optimized BERT approach) pre-trained model as the semantic backbone because RoBERTa model has better downstream task performance than BERT model [8]. Adversarial jailbreak attacks often rely on subtle linguistic manipulation, which is why a model that can provide a more nuanced understanding of linguistic context is better for our purpose, which is why RoBERTa was chosen.

Stage 1: To train the chosen RoBERTa model specifically on jailbreak harm scoring, first we defined the range of harm score for prompts, in which we defined that 0-0.3 range is allocated for normal prompts, 0.4-0.6 range is allocated for prompts that can appear harmful/dangerous but are not harmful/jailbreak prompts in context, and finally 0.7-1 range is allocated for jailbreak prompts. Now to train our model on this specific task of predicting the harmscore of a prompt, we first initialize it through a multi-label toxicity classification using the Jigsaw dataset [40]. The purpose of doing this is to teach the pre-trained RoBERTa model to identify surface level toxicity features on multi-labels to employ transfer learning, bypassing the need for a large supervised dataset.

Stage 2: In this second stage, the classification is head is replaced with a regression head to perform semantic intent calibration. We implemented Mean Pooling to transform the 12 layer transformer hidden states into a single intent vector. Mean Pooling is used over [CLS] pooling because Mean Pooling aggregates semantic information from all the tokens, making it aware of harmful instructions located anywhere in the prompt.

The model maps the intent vector to a scalar h using a Sigmoid activation:

$$h = \sigma(\mathbf{W}^T\mathbf{u} + b) \tag{4.1}$$

The pooled intent vector u is converted into a scalar harm score h using a linear transformation and a Sigmoid activation as expressed in Equation (4.1). The weights W and bias b are learned during training. To improve generalization, we apply a dropout of 0.4 to the pooled vector before the regression head, which randomly zeroes a fraction of the vector elements during training to prevent

Algorithm 1 Two-Stage Training of the Harm Controller

Require: Toxicity dataset D_{tox}, Harm spectrum dataset D_{harm}, Pretrained RoBERTa encoder R_ϕ
Ensure: Calibrated harm scoring model H_ϕ

Stage 1: Linguistic Toxicity Foundation
Attach a multi-label toxicity classification head to R_ϕ
for each training epoch **do**
    Encode input text using RoBERTa
    Predict multiple toxicity categories (e.g., insult, threat)
    Update model weights using binary cross-entropy loss
end for
Stage 2: Semantic Intent Calibration
Remove the toxicity classifier and attach a single scoring head
for each training epoch **do**
    Encode the prompt using RoBERTa
    Aggregate token representations into a sentence-level vector
    Predict a harm intensity score between 0 and 1
    Update model weights using binary cross-entropy loss with a small learning rate
end for
return Trained harm scoring model H_ϕ

overfitting. The model is trained for 3 epochs with a batch size of 12, using a small learning rate of 1×10^{-5} and binary cross-entropy (BCE) loss as the regression objective. These settings ensure stable training and high precision in harmful regions of the output.
The overall training procedure for the Prediction model is summarized in Algorithm 1.

Figure 4.1 illustrates the two-stage training pipeline used for the prediction model.

After designing the Prediction model, we construct a black box knowledge distillation pipeline to define the baseline distillation. The baseline distillation process optimizes a joint training objective that combines supervised instruction learning with probabilistic imitation of the teacher model. Given a teacher model T and a student model S_θ, the goal is to construct a weighted sum of the Cross-Entropy loss (L_{CE}) and the Kullback–Leibler divergence (L_{KL}).

The student model learns to from the teacher model via minimizing the KL Divergence between their smoothed output distributions. In this standard configuration, the student model receives a uniform signal from the teacher regardless of the prompt's intent. The total loss $\mathcal{L}_{total}$ is weighted by a static learning rate α:

$$\mathcal{L}_{total} = \alpha\mathcal{L}_{CE} + (1 - \alpha)\mathcal{L}_{KL} \tag{4.2}$$

Equation (4.2) denotes the general knowledge distillation loss. This generic loss function is constructed from summing up with CE loss and KL divergence. Over time student model tries to learn from only one bigger teacher model. This baseline acts as a Safety-Critical Control, transferring the teacher's reasoning ability without any mechanism to minimize the teacher's adversarial vulnerabilities transferring to the student.

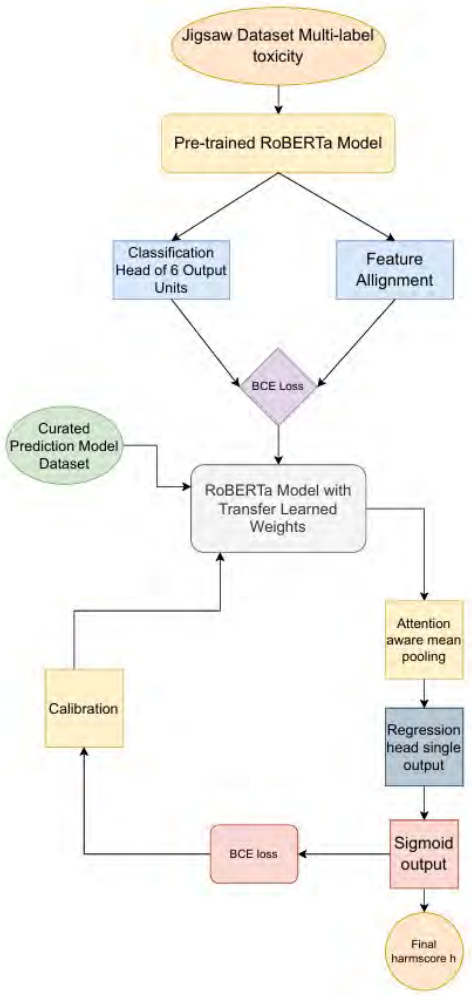

Figure 4.1: Training architecture of the Prediction model

4.3 Dataset Creation

We have used multiple datasets from Hugging Face and divided them into two main parts for our entire pipeline such as training dataset and test dataset. The training datasets are used in our prediction model pipeline and knowledge distillation pipeline whereas the test datasets are used to benchmark our teacher and different versions of student model accordingly.

4.3.1 Training Dataset:

Prediction Model: We availed the Jigsaw toxic comment classification challenge dataset[40] from Hugging Face which consists of 150k harmful Wikipedia comments of six categories which are toxic, severe toxic, obscene, threat, insult and identity hate for the transfer learning process of our prediction model. Additionally, we have used multiple datasets to set the score range of input prompts from 0 to 1 where 0.0 is the most safest and normal prompts and 1 being the most harmful prompts.

Harm Scoring Range and Dataset:
Score 0.0 : Our scoring range for the input prompts starts from 0.0 for which we used the pref-

erence data math stack exchange dataset[50] that has 18726 Q/A on mathematical problems from the Stack Exchange Data Dump. We scored this Q/A as the least harmful as these are used only for mathematical and academic purposes.

Score 0.1: Moving on to our 0.1 range prompts, we used the SQUAD dataset[5]which is a reading comprehension dataset that contains 100k questions on more than 500 Wikipedia articles and their corresponding answers hence these are not harmful prompts as well.

Score 0.2-0.3: For our 0.2-0.3 range we used SQUAD v2 [7] dataset that contains the SQUAD dataset Q/A with 50000 more answerable questions and in total 130k questions and answers. To score these, we used some dual use keywords as mentioned in [26] such as "hack", "breach", "toxic", "break" etc and safety context keywords such as "history", "security", "prevention", "policy" etc.. If any input prompt has only dual use keywords or safety context keywords we score them as 0.3 and if any prompt has both then we score them as 0.3 range prompts which is not harmful but consist of some harmful context and keywords.

Score 0.4-0.5: Afterwards, for 0.4-0.5 range prompts we generated 2000 middle ranged prompts by rule generated script where we have used templates as described in [53], keywords and modifiers to implement the scenarios, characters and actions of the prompts which eventually creates not normal not much harmful rather middle ranged prompts.

Score 0.6-1: For crafting the slightly harmful to most harmful prompts we used the SpeakEasy[62] pipeline on our datasets. In this dataset, we used GPTFuzz[32], in the wild jailbreak[51],AdvBench[35] and JBB-Behavior(hamrful)[36] datasets' prompts with their PAIR [21] and DSN [60]attack based variants and generate their 3 sub-query or variant of the input prompts, later on we measured the actionability and informativeness of these prompts and got scores. These scores are the added to have combined scores of the variant and whichever variant has highest combined score we generated its response by our teacher model and evaluated this response by our judge model to check whether it is a refusal or not. In the case of not refusal, we calculate its harmscore by multiplying its actionability and informativeness score and later on find the square root of the score to have its final harmscore. For refusal response cases,its harmscore is set to 0 which is later scaled linearly from 0.6 to 1.

Contrastive Dataset: We used the JBB-Behavior[36] dataset's 100 beingn prompts and generated its hamrful versions using our teacher model. Then the benign prompts are set with a score of 0.15 and hamrful version of the benign prompts are set with the score of 0.85. Eventually, this helped our prodection model to a great extent to accuarately score the prompts harmscore later on.

Therefore, in total our training dataset of the prediction model consists of 6315 prompts of different categories.

Knowledge Distillation: For training our models in knowledge distillation, we used in total 3100 prompts among which 1550 are jailbreak prompts from the JailbreakV[45] dataset and 1550 are normal prompts from Ultrachat_200k[24] and OpenOrca[27] to have a balanced learning process.

4.3.2 Test Dataset:

For testing our teacher model, base student, distilled student model after normal KD and dynamic student model after dynamic KD we used 520 prompts from AdvBench [35] and 100 from JBB-Behavior hamrful prompts. Then on this 620 base prompts we applied two popular jailbreak attack which are DSN [60] and PAIR [21].

DSN: This DSN code inspired from [60],generates adversarial jailbreak prompts by applying single-step, gradient-based token perturbations to the base dataset using Qwen-2.5-7B-Instruct as the perturbation model as its high efficiency in terms of reasoning [63]. For each prompt, the code computes gradients of the language modeling loss with respect to input embeddings to identify the most sensitive token position, replaces that token with a small set of gradient-aligned candidate tokens and filters candidates using a sentence-embedding semantic judge (MiniLM-L6-v2) to ensure the perturbed prompt remains semantically similar to the original. Among valid candidates, the version with the lowest loss is selected to preserve fluency. This process produces a fast, semantics preserving adversarial dataset expansion,by minimally modified but more attack-effective jailbreak prompts that are later used to attack our target model in a black-box setting.

PAIR: This PAIR code is a simpler version of the [21] that implements a fully black-box, iterative jailbreak attack pipeline where Qwen-2.5-7B-Instruct [63] acts as the attacker that progressively refines adversarial base prompts,teacher model, student model, distilled student model and dynamic student model are the target models and a judge model (via Ollama) scores jailbreak success. Starting from a base jailbreak dataset, the attack model generates an initial adversarial prompt using one of several persuasion strategies (roleplaying, logical appeal, or authority endorsement) combined with controlled light obfuscation (symbols, leetspeak, homoglyphs under a 20 percent limit). The perturbed prompt is then fed to the target model to obtain a response, which is evaluated by the judge on a strict 1–10 jailbreak scale. If the score is low, the judge's score and the target's response are fed back to the attack model, which iteratively improves the prompt over 3 rounds, rotating strategies and refining wording until the response crosses the jailbreak threshold or iterations are exhausted.
So, overall to test our models we have in total 1860 prompts in our test dataset.

4.4 Implementation of Selected Design

Figure 4.2 illustrates the overall training pipeline for our proposed DGD framework.

In the preliminary design section, we discussed about our Prediction Model that intelligently assigns harmscore (h) for any given prompt according their threat level. This harmscore is basically the heart of our proposed framework, that will further control the gating mechanism between utility teacher and safety teacher. Depending on the harmscore for any input prompt, the information will pass down to each and every step of the DGD framework. A step by step algorithmic view of our proposed DGD framework is explained below:

The Control Gate
The harmscore gets passed into the logistic sigmoid function, which is defined by $w(h)$ coefficient.

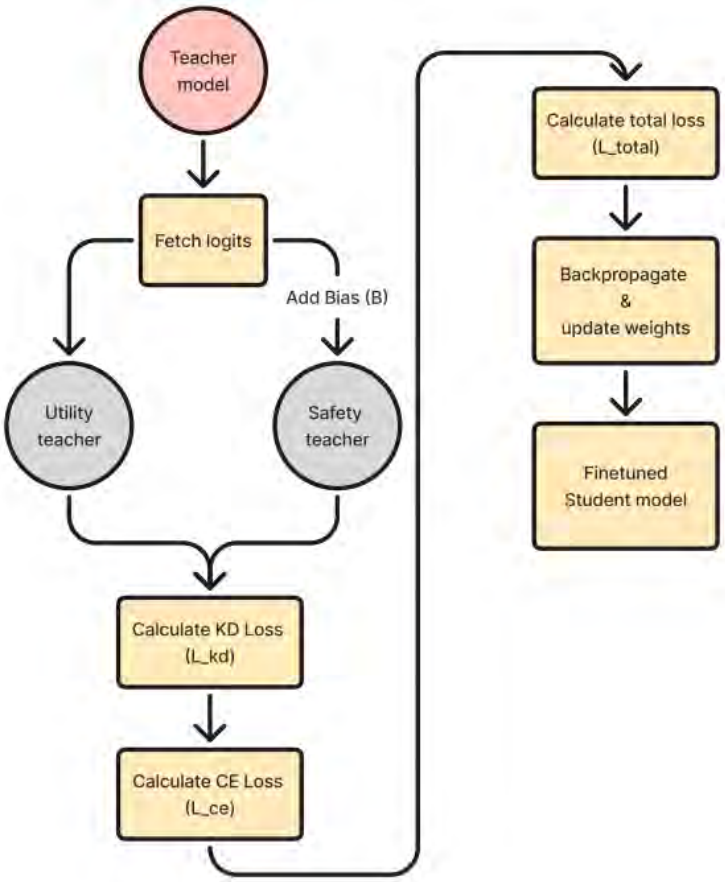

Figure 4.2: Training Pipeline of Dynamic Gated Distillation (DGD)

Notably the harmscore range is $h \in [0, 1]$.

$$w(h) = \frac{1}{1 + e^{-k(h-h_0)}} \tag{4.3}$$

Equation (4.3) explains the overall control gate construction. Here, h_0 is a constant, which we are calling decision threshold and the value of it is set set to 0.7. The transition steepness ($k = 15$) determines how much steep the sigmoid curve will be. Higher steepness causes the sigmoid value to jump rigorously. Setting it to 15 maintains a decent steepness. $w(h)$ function determines the dynamic weight between the utility and safety teachers.

Constructing the Utlity and Safety Teacher

The utlity teacher and safety teacher both acts as two different virtual teacher. Both has their own contribution in the distillation process. We simply fetch all the logits from the teacher model and made two copies of it.

$$Z_H = Z_U + B, \quad B_i = \begin{cases} \beta & \text{if } i \in \mathcal{V}_{safe} \\ 0 & \text{else} \end{cases} \tag{4.4}$$

In Equation (4.4), Z_U denotes the logits of utility teacher and Z_H denotes the logits of safety teacher. Given a vocabulary set of refusal tokens, that we constructed from common refusal responses [61], $\mathcal{V}_{refusal} \subset \mathcal{V}$, we constructed the safety teacher by adding a safety bias score B with logits of the

corresponding tokens belonging in the refusal vocabulary set. So basically, the logic is that if there are refusal tokens, increase the bias score of those tokens by adding the safety bias score, if not just add 0 to keep the logits as it is. This soft biasing towards refusal tokens, ensures during distillation if the response is harmful, the student model will tend to align with the safety teacher accordingly.

Dynamic Distillation Loss
The Knowledge Distillation loss $\mathcal{L}_{KD}$ was formulated using a combination of utility teacher and safety teacher output probability distribution. Let, $P_S = \text{Softmax}(Z_S/\tau)$ be the output probability distribution from the student model, $P_U = \text{Softmax}(Z_U/\tau)$ be the output distribution from the Utility Teacher and $P_H = \text{Softmax}(Z_H/\tau)$ be the output distribution from the Safety Teacher.
Summing up the unified distillation loss that we get:

$$\mathcal{L}_{KD} = (1 - w(h))\mathcal{D}_{KL}(P_S||P_U) + w(h)\mathcal{D}_{KL}(P_S||P_H) \tag{4.5}$$

In Equation (4.5), $\mathcal{D}_{KL}$ denotes the kullback-leibler divergence. The $w(h)$ control the gating mechanism of distillation learning process. If the value of $w(h)$ is high, the student model will try to align and learn from safety teacher. If the value of $w(h)$ is low it will lean towards utility teacher.

Hard Loss
In the distillation process, the reponse column acts as hard label to calculate Cross-Entropy (CE) loss. This CE loss ensures that the student model doesn't totally forget about the ground truth label or the response column.

$$\mathcal{L}_{CE} = -\sum_{t \in y} \log P(y_t|y_{<t}, x) \tag{4.6}$$

Equation (4.6) CE loss keeps the check that the model doesnt generate gibberish answers if any corner case comes up.

Global Distillation Loss
In the final loss function we added the CE loss $\mathcal{L}_{CE}$ and Knowledge Distillation loss $\mathcal{L}_{KD}$, multiplicating with a learning parameter α. This α will regulate the final loss between the ground truth learning and the teacher model learning.

$$\alpha(h) = \text{clamp}(1 - w(h), 0.1, \alpha_{base}) \tag{4.7}$$

$$\mathcal{L}_{total} = \alpha(h)\mathcal{L}_{CE} + (1 - \alpha(h))\mathcal{L}_{KD} \tag{4.8}$$

In Equation (4.7), the base value of $\alpha_{base} = 0.5$. In other words $\alpha(h)$ is a trust regulator, which determines when to trust the ground truth and when to trust the KD loss. By clamping the $\alpha(h) \in [1 - w(h), 0.1, 0.5]$, we ensured that our student model doesn't fully nullify one part of the loss. while blindly trusting on the other part of the loss. Equation (4.8) denotes the total loss including CE loss and KD loss. Based on this loss the student model will update its weights and gets learn over the time during backpropagation.

Proposed DGD Framework Algorithm

Algorithm 2 Dynamic Gated Distillation

Input: Teacher T_U, Student T_S, Dataset $\mathcal{D}$, Threshold h_0, Epoch E
Output: Total loss $\mathcal{L}_{total}$
Initialize: $\alpha_{base} = 0.5, \tau = 4.0, k = 15$
for epoch in $1 \dots E$ **do**
    for each batch $(x, y, h) \in \mathcal{D}$ **do**
        $w \leftarrow \sigma(k(h - h_0))$ ▷ Control gate
        $Z_U \leftarrow T_U(x)$ ▷ Forward pass Teacher
        $Z_H \leftarrow Z_U + B$ ▷ Adding bias to refusal tokens
        $Z_S \leftarrow T_S(x)$ ▷ Forward pass Student
        $\mathcal{L}_{CE} \leftarrow \text{CrossEntropy}(Z_S, y)$
        $\mathcal{L}_{KD_U} \leftarrow \text{KL}(Z_S, Z_U)$
        $\mathcal{L}_{KD_H} \leftarrow \text{KL}(Z_S, Z_H)$
        $\mathcal{L}_{KD} \leftarrow (1 - w(h)\mathcal{L}_{KD_U} + w(h)\mathcal{L}_{KD_H}$ ▷ Dynamic KD loss
        $\alpha(h) \leftarrow \text{clamp}(1 - w, 0.1, \alpha_{base})$
        $\mathcal{L}_{total} \leftarrow \alpha(h)\mathcal{L}_{CE} + (1 - \alpha(h))\mathcal{L}_{KD}$ ▷ Total loss
        $\mathcal{L}_{total}$.backward()
        optimizer.step()
    end for
end for

The proposed Algorithm 2 of DGD, illustrates the overall algorithmic view of our proposed DGD framework. The whole algorithm will run batch wise in every epoch. In every batch there will be three component prompt(x), response(y) and corresponding harmscore(h). First calculating the value of w. After that the algorithm follows standard forward pass of teacher and student model. Fetching those logits and keeping a two copies of teacher logits. After that adding bias B of the refusal tokens in the logits. Calculating the CE loss L_{CE} from the student logits. Then the algorithm calculates KL Divergence of utility teacher (Z_U) vs student model (Z_S) and safety teacher (Z_H) vs student model (Z_S). Here is where the main learning happens, because while calculating KL divergence with utility teacher and safety teacher, the student model tries allign with the output probability distribution from both of these teacher. After that the framework calculates the KD loss (L_{KD}) and the final total loss (L_{total}) by adding up the CE loss and KD loss. After one successfull epoch it will backpropagate and update weights.

Chapter 5

Result Analysis

5.1 Performance Evaluation

Our main purpose of this research is to enhance the robustness of smaller language models that can handle jailbreak cases efficiently while maintaining general reasoning even with smaller parameters using our dynamic gated distillation (DGD) pipeline. To ensure this, we evaluated our teacher and student models based on some evaluation metrics such as Attack Success Rate(ASR) and MMLU [11] benchmark. For the prediction model, we calculated the Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Spearman's Rank Correlation and R^2 as mentioned in [13] to evaluate the model's prediction accuracy.

Attack Success Rate (ASR) : This metric measures the proportion of the number of successful jailbreak prompts and number of total prompts used in benchmark dataset.
For our evaluation ASR is calculated as following :

$$\text{ASR} = \frac{\text{Number of successful jailbreak prompts}}{\text{Total number of prompts}} \times 100\% \tag{5.1}$$

Using Equation (5.1), we have evaluated the successful jailbreak responses to calculate different models' adversarial resistance.

Massive Multitask Language Understanding: MMLU [11] is a measuring benchmark that measures the performance of large language models (LLMs) in understanding and reasoning across many academic and professional domains. The measuring benchmark consists of 15,908 multiple choice questions in 57 domains including math, physics, computer science, law, medicine, history, economics and ethics. The questions in the measuring benchmark range from high school to professional levels and the models are often tested in zero-shot or few-shot settings which explains that they must comply with their general alignment rather than specific task based fine tuning.

Mean Absolute Error (MAE): MAE is the absolute average of distance of the predicted results values and actual values [3]. For the regression task of RoBERTa based models e.g.the prediction of eye-tracking features [15], the MAE measures the proximity of the predicted numbers and the actual numbers. A smaller MAE implies a higher accuracy of the model.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{5.2}$$

Accoding to Equation (5.2), y_i denotes the ground truth value and $\hat{y}_i$ the predicted value and n the total number of samples.

Root Mean Squared Error (RMSE): RMSE is the square root of the average of the squared differences between the predicted and actual values [3]. This means that larger errors are given more weight. Although less frequently used than MAE, this is also used for evaluating the accuracy of prediction models in regression tasks particularly in NLP models.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{5.3}$$

In Equation (5.3), where y_i and $\hat{y}_i$ represent the true and predicted values respectively and n is the number of observations.

Spearman's Rank Correlation: This metric checks if the predicted and actual values have a monotonic relationship. For the semantic similarity and complexity prediction of the RoBERTa-based models [13], Spearman's correlation is widely used. This checks if the model has the correct order of the data even if the actual values are different[1].

$$\rho = \frac{\text{cov}\left(\text{rank}(X), \text{rank}(Y)\right)}{\sigma_{\text{rank}(X)}\,\sigma_{\text{rank}(Y)}} \tag{5.4}$$

In Equation (5.4),rank(X) and rank(Y) denote the ranked variables,σ represents the standard deviation of ranks and n is the total number of samples.

R^2: R^2 measures the proportion of the variance of the actual values explained by the predicted values of the model. For the regression and lexical complexity task of the RoBERTa based models, R^2 is used widely. It is used to determine the accuracy of the model in capturing the data of the target variable, not simply minimizing the errors [2].

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{5.5}$$

Here, in Equation (5.5),$\bar{y}$ is the mean of the ground truth values, y_i is the true value,$\hat{y}_i$ is the predicted value and n denotes the total number of samples.

Training Loss and Validation Loss: The training loss measures the error of the model on the training data and how well the model is fitting the data. On the other hand, the validation loss measures the error of the model on the validation data and how well the model is able to generalize. These metrics are useful for detecting overfitting for RoBERTa based prediction models.

5.2 Analysis of Design Solutions

In this section, we have assessed the performance and robustness of our RoBERTa based prediction model and our teacher, student models in terms of prediction and adversarial attacks. The performance metrics considered are regression metrics such as MAE, RMSE, R^2, Spearman and Attack Success Rate (ASR) which represents the model's robustness against adversarial attacks. MMLU average accuracy is also reported as a performance metric to represent the overall performance of the model on general reasoning tasks.

Table 5.1: Regression metrics of the prediction model across training epochs

Epoch	MAE	RMSE	R^2	Spearman	Training Loss	Validation Loss
1	5.05	6.78	0.9540	0.9335	49.17	46.77
2	4.87	6.73	0.9547	0.9360	46.71	46.73
3	4.69	6.56	0.9569	0.9368	46.56	46.66

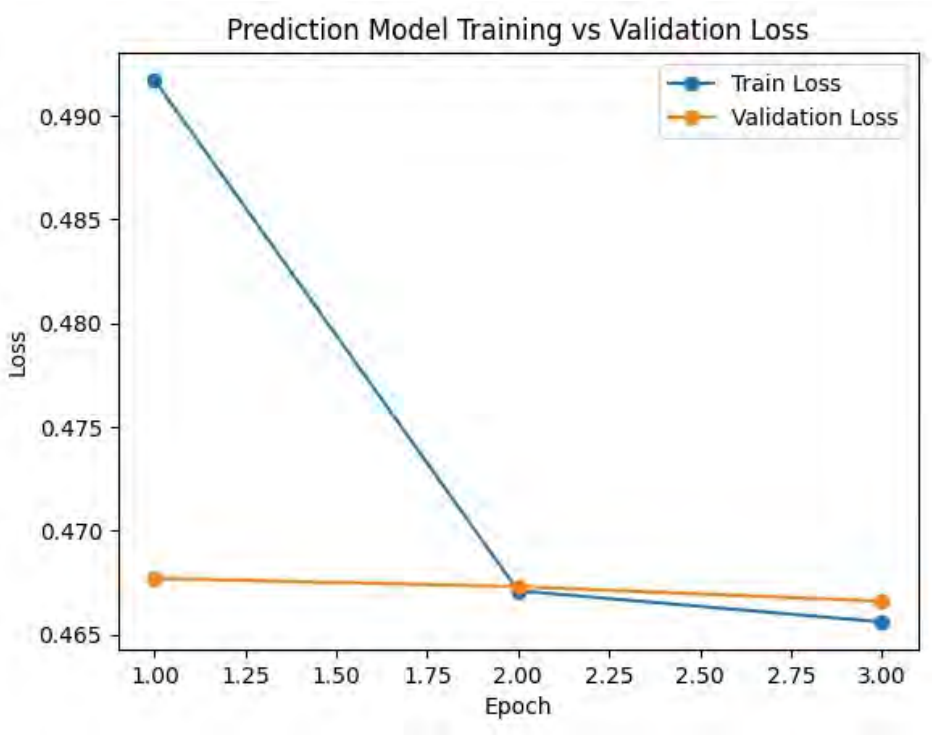

Figure 5.1: Training and validation loss curves for the prediction model across epochs

Our prediction model exhibits excellent convergence and generalization performance across three training epochs. At Table 5.1 we can see that,the prediction model improves consistently across all metrics with a reduction in MAE from 5.05 to 4.69, a decrease in RMSE from 6.78 to 6.56 and an increase in R^2 from 0.9540 to 0.9569, thereby showing that the model is successful in explaining more than 95% of the variance in the target variable. The training process exhibits healthy learning behavior with training loss reducing consistently from 49.17 to 46.56 while maintaining a stable validation loss at 46.66-46.77. As seen in Figure 5.1, training and validation losses converge by epoch 3, indicating minimal overfitting and strong generalization. High values of correlation coefficients are consistently maintained across all training epochs with Spearman correlation improving from

0.9335 to 0.9368. This indicates that the model is successful in maintaining rank order relationships. The consistent improvement across all metrics with no signs of performance degradation or instability suggests that the prediction model's architecture is suitable for prompt harmscore prediction task, efficiently extracting all complex relationships while maintaining excellent generalization performance.

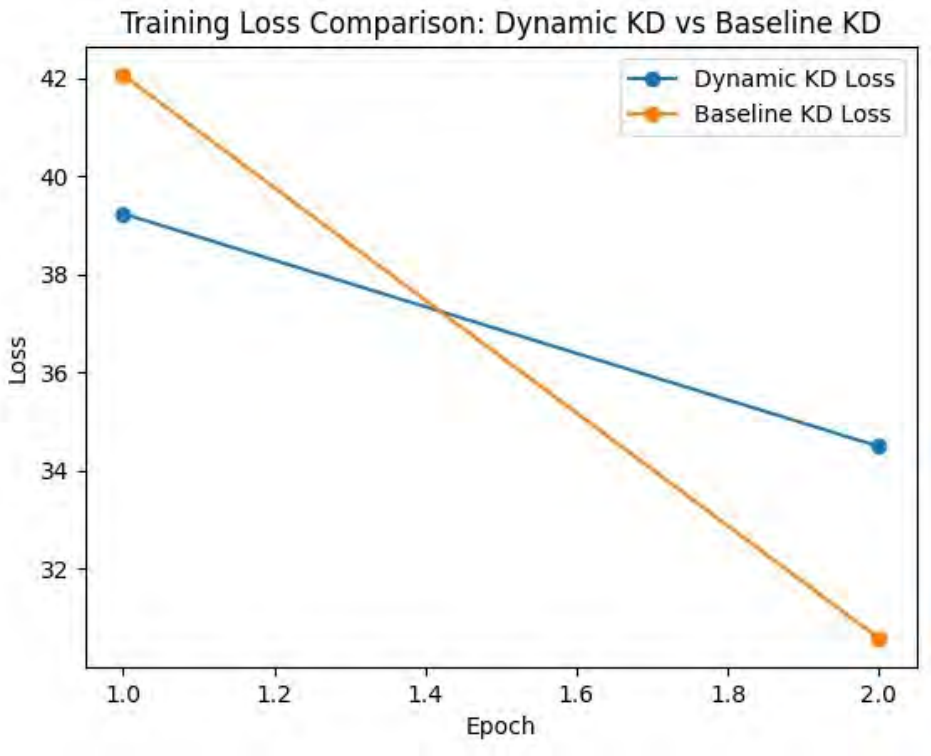

Figure 5.2: Training loss comparison: Dynamic KD vs Baseline KD

Looking at the comparison of the knowledge distillation loss indicates the training process between the dynamic gated KD and baseline KD methods over two epochs. It is noticeable that both methods show a trend towards decreasing loss values which confirms successful knowledge transfer from the teacher to the student model. As we see in Figure 5.2, baseline KD method initially has a higher loss value around 42, which decreases sharply to 30.5. On the other hand, the dynamic gated KD method initially has a lower loss value around 39, and decreases gradually to 34.5. The sharp decrease of the baseline KD method can be considered an aggressive training proces while the gradual decrease of the dynamic KD method can be considered a controlled training process. The lower loss value of the baseline KD method indicates better alignment with the teacher's output distribution while the higher loss value of the dynamic KD method indicates better performance retention in MMLU compared to the baseline KD method.

The table at 5.2, illustrates the ASR and MMLU average accuracy for four different Llama based models. The base student model is more susceptible to adversarial attacks, with an average ASR of 67.85%. The ASR for this model is high for all four attack types: BASE (69.19%), DSN (69.03%), PAIR (65.32%). The normal distilled student model has a much better ASR of 0.05%, which indicates high robustness to adversarial attacks. However, this model has relatively low accuracy in MMLU of 40.60%, indicating degraded reasoning performance compared to the base student model. The dynamic gated distilled student model has a good balance of ASR and MMLU accuracy. The

Table 5.2: Attack Success Rate (ASR) and MMLU Average Accuracy for Llama Based Models

Model	Avg ASR (%)	BASE (%)	DSN (%)	PAIR(%)	MMLU Avg Accuracy (%)
Llama-3.1-8B-Instruct(Teacher)	17.53	16.61	18.87	17.10	71.58
Llama-3.2-1B-Instruct(Base Student)	67.85	69.19	69.03	65.32	42.81
Llama-3.2-1B-Instruct (Distilled Student)	0.05	0.16	0.00	0.00	40.60
Llama-3.2-1B-Instruct(DGD Student)	3.39	3.06	3.55	3.55	42.13

ASR for this model is 3.39%, which is slightly higher than that of the normal distilled student model and significantly lower than the ASR of the teacher model (17.53%) and the base student model (67.85%). However, this model has a higher accuracy in MMLU of 42.13%, which is much better than the normal distilled student model and indicates better reasoning and generality. The teacher model has a much better accuracy in MMLU of 71.58%, but this model is more susceptible to adversarial attacks than the dynamic gated distilled student model.

Overall, it is evident that the dynamic gated distilled student model is a robust and reasoning model, effectively balancing ASR and MMLU accuracy and is a good candidate for applications requiring resistance to adversarial or jailbreak attacks and cognitive reasoning.

5.3 Final Design Adjustments

For the final stage of design refinements, various parameters and architectural choices have been tuned to improve the performance of the prediction model as a prompt based assistant. The dropout rate has been set to 0.4 which helps in the regularization of the model and reduces the chances of overfitting. The training process is done for 3 epochs with a batch size of 12 and a learning rate of 1e-5. In addition, a response column was added to the dataset, enabling the student model to better align its outputs with the prompt. The logits safe bias is reduced from 20 to 8, which reduces the number of refusals but still maintains the safety constraints. Overall, these changes helped in creating a more efficient, accurate and prompt aligned assistant.

5.4 Statistical Analysis

In order to critically examine the effectiveness of various Llama based models, we adopted descriptive and comparative statistical measures where the MMLU average accuracy and the Attack Success Rate (ASR) were used. ASR was determined for each model using three different types of attacks: BASE, DSN, and PAIR. The average ASR value was used to determine the general robustness of the models. On the other hand, MMLU scores were also determined to examine the general

reasoning and multitasking capabilities of the models on various domains. This method helped to determine the strengths and weaknesses of the models where the base student model showed a high ASR value indicating that the model was vulnerable to jailbreak attacks, while the normal distilled and dynamic gated distilled student models showed a low ASR value, indicating that the models were highly resistant to adversarial attacks.

5.5 Comparisons and Relationships

The trade offs between robustness and reasoning performance are further corroborated by existing literature on evaluation, distillation and safety [11] introduce MMLU as a comprehensive benchmark for evaluating multi-domain reasoning and knowledge retention, thereby explaining the reasons why a larger teacher model performs significantly better in terms of MMLU accuracy and why the student models experience a degradation in performance. [4] prove that conventional knowledge distillation where teacher and student logits are closely aligned, imposes undue constraints on the student representations. This is in line with the extremely low ASR and lower MMLU accuracy of the typically distilled student. However, existing adaptive knowledge distillation techniques such as the dynamic or gated knowledge transfer mechanisms in [31], prove that a controlled level of regulation is sufficient for the student models to retain reasoning capabilities and improve robustness. This is in direct alignment with why the dynamic gated distilled model performs better in MMLU accuracy compared to the typically distilled model despite a slightly higher ASR and is significantly more robust than the teacher model. These studies prove that dynamic gating provides a principled trade-off between safety and reasoning, rather than optimizing one at the expense of the other.

To shed a light on this section, our teacher model has depicted the highest accuracy in terms of MMLU (71.58%), it has a moderate level of vulnerability to adversarial attacks with an ASR of 17.53%. On the other hand, the base student model has a high level of vulnerability with an ASR of 67.85%, while at the same time having limited defense mechanisms in place. Conversely, the normally distilled student model minimizes model vulnerability with an ASR of 0.05%, but at the same time having a lower level of reasoning performance with an MMLU accuracy of 40.60%. However, the dynamic gated distilled student model provides a balanced outcome with a lower ASR of 3.39% having a higher level of reasoning performance compared to the normally distilled student model with an MMLU accuracy of 42.13%. This shows that the application of dynamic gating in knowledge distillation provides a viable option for designing LLMs based on size, robustness and generalization against adversarial attacks.

5.6 Discussions

The key findings of our research provide significant insights into the performance and limitations of the tested Llama models. For instance, the performance of the dynamic gated distilled (DGD) student model shows significant robustness, as indicated by the low ASR of 3.39% and also an improvement in reasoning compared to the normal distilled student model. However, the accuracy of MMLU does not show a significant improvement. This indicates that the DGD model has the capacity for robustness and also preserves the capacity for reasoning, although there is no significant improvement in reasoning beyond the level of the baseline model. Morever,the hardware limitations including the 24GB VRAM of the RTX 4090, limited the loading of models of larger parameter scale which was very crucial for us. Similarly, the 12 hour time slot of the pc also limited our training

of some of the epochs, which required more than 16 hours hence we had to settle for less and small trials for our research most of the time. Therefore, these limitations impacted our testing phases greatly but these obstacles also helped us to look for more optimal solutions at the same time.

Chapter 6

Conclusion

6.1 Summary of Findings

Our evaluation results show that different models based on the Llama architecture have different trade-offs in terms of robustness and reasoning capabilities. Llama-3.1-8B-Instruct, the teacher model achieves the best accuracy in the MMLU which indicates the best reasoning and retaining knowledge capabilities although it also shows moderate jailbreak adversarial attack vulnerability. Llama-3.2-1B-Instruct, the base student model shows the highest vulnerability to attacks while the normal distilled student model shows reduced vulnerability at the cost of reasoning capability. Interestingly, the dynamic gated distilled student model shows the best balance in terms of maintaining low ASR vulnerability and increasing the accuracy of the MMLU compared to the normal student model.

6.2 Contributions to the Field

Our research focuses on the significance of Dynamic Gated Knowledge Distillation (DGD) in the development of robust instruction following large language models (LLMs). We have tried to improve the limitations of the conventional knowledge distillation method which either compromises the reasoning ability of the LLM or fails to improve robustness. The student model of the DGD achieves an optimal performance with low Attack Success Rates (ASR) on adversarial jailbreak prompts and reasoning proficiency on the MMLU benchmark. Moreover, our work contributes to the discourse on the development of safe and robust LLMs by providing guidelines on the development of LLMs with an optimized balance between safety and generalization. Additionally, our proposed prediction model works efficiently for assessing the harmscore of adversarial prompts. This research provides useful strategies on the development of safe and reliable large language models by providing a clear path to the balance between safety and generalization and the prediction model for judging the harm of the adversarial prompts shows that dynamic gating can be incorporated in the usual distillation to obtain robust and effective student models.

6.3 Recommendations for Future Work

The future of this work should involve investigating ways to improve reasoning and robustness in distilled LLMs further. For instance, improving the model size and fine-tuning quality of teacher and judge models can potentially improve the knowledge base used in distillation, which in turn

improves the MMLU accuracy. Additionally,increasing the dataset used in training the prediction model and the DGD pipeline should be expanded to improve generalization and robustness against adversarial prompts. Also, from our observation we have encountered the fact that sometimes irrelevant or inconsistent responses are generated from input prompts calls for an investigation of methods to mitigate LLM hallucinations such as methods that improve context grounding and reinforcement learning from human feedback(RLHF). Overall, these prospective works are expected to result in LLMs that are not only secured from jailbreak attacks but also very reliable in reasoning, making them safer and more effective in practice.

Bibliography

[1] C. Spearman, "The proof and measurement of association between two things," *The American Journal of Psychology*, 1904.

[2] N. R. Draper and H. Smith, "Applied regression analysis," *Technometrics*, 1966.

[3] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd. Springer, 2009.

[4] G. Hinton, O. Vinyals, and J. Dean, *Distilling the knowledge in a neural network*, arXiv preprint arXiv:1503.02531, 2015. [Online]. Available: https://arxiv.org/abs/1503.02531

[5] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Nov. 2016, pp. 2383–2392. DOI: 10.18653/v1/D16-1264 arXiv: 1606.05250 [cs.CL]. [Online]. Available: https://aclanthology.org/D16-1264

[6] 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 (KDD '17)*, Aug. 2017, pp. 1285–1294. DOI: 10.1145/3097983.3098135

[7] P. Rajpurkar, R. Jia, and P. Liang, "Know what you don't know: Unanswerable questions for squad," *arXiv preprint arXiv:1806.03822*, 2018. [Online]. Available: https://arxiv.org/abs/1806.03822

[8] Y. Liu et al., *Roberta: A robustly optimized bert pretraining approach*, arXiv preprint arXiv:1907.11692, Jul. 2019. [Online]. Available: https://arxiv.org/abs/1907.11692

[9] T. Wen, S. Lai, and X. Qian, *Preparing lessons: Improve knowledge distillation with better supervision*, arXiv preprint arXiv:1911.07471, Nov. 2019. [Online]. Available: https://arxiv.org/abs/1911.07471

[10] H. Zhao, X. Sun, J. Dong, C. Chen, and Z. Dong, *Highlight every step: Knowledge distillation via collaborative teaching*, arXiv preprint arXiv:1907.09643, Jul. 2019. [Online]. Available: https://arxiv.org/abs/1907.09643

[11] D. Hendrycks et al., *Measuring massive multitask language understanding*, arXiv preprint arXiv:2009.03300, 2020. [Online]. Available: https://arxiv.org/abs/2009.03300

[12] Y. Liu, W. Zhang, and J. Wang, "Adaptive multi-teacher multi-level knowledge distillation," *Neurocomputing*, vol. 415, pp. 106–113, Jul. 2020. DOI: 10.1016/j.neucom.2020.07.048

[13] T. Bani Yaseen, Q. Ismail, S. Al-Omari, E. Al-Sobh, and M. Abdullah, "Just-blue at semeval-2021 task 1: Predicting lexical complexity using bert and roberta pre-trained language models," in *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, Association for Computational Linguistics, 2021, pp. 661–666. DOI: 10.18653/v1/2021.semeval-1.85 [Online]. Available: https://aclanthology.org/2021.semeval-1.85

[14] T. Kim, J. Oh, N. Y. Kim, S. Cho, and S.-Y. Yun, *Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation*, arXiv preprint arXiv:2105.08919, May 2021. [Online]. Available: https://arxiv.org/abs/2105.08919

[15] B. Li and F. Rudzicz, "Torontocl at cmcl 2021 shared task: Roberta with multi-stage fine-tuning for eye-tracking prediction," in *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics (CMCL 2021)*, Association for Computational Linguistics, 2021, pp. 85–89. DOI: 10.18653/v1/2021.cmcl-1.9 [Online]. Available: https://aclanthology.org/2021.cmcl-1.9

[16] P. Lu, A. Ghaddar, A. Rashid, M. Rezagholizadeh, A. Ghodsi, and P. Langlais, "RW-KD: Sample-wise loss terms re-weighting for knowledge distillation," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, Jan. 2021, pp. 3145–3152. DOI: 10.18653/v1/2021.findings-emnlp.270

[17] Z. Li et al., *Curriculum temperature for knowledge distillation*, arXiv preprint arXiv:2211.16231, Nov. 2022. [Online]. Available: https://arxiv.org/abs/2211.16231

[18] L. Weidinger et al., "Taxonomy of risks posed by language models," in *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency (FAccT '22)*, Jun. 2022, pp. 214–229. DOI: 10.1145/3531146.3533088

[19] H. Zhang, D. Chen, and C. Wang, *Confidence-aware multi-teacher knowledge distillation*, arXiv preprint arXiv:2201.00007, Jan. 2022. [Online]. Available: https://arxiv.org/abs/2201.00007

[20] N. Carlini et al., *Are aligned neural networks adversarially aligned?* arXiv preprint arXiv:2306.15447, Jun. 2023. [Online]. Available: https://arxiv.org/abs/2306.15447

[21] P. Chao, A. Robey, E. Dobriban, H. Hassani, G. J. Pappas, and E. Wong, *Jailbreaking black box large language models in twenty queries*, arXiv preprint arXiv:2310.08419, Oct. 2023. [Online]. Available: https://arxiv.org/abs/2310.08419

[22] J. Cui, Z. Tian, Z. Zhong, X. Qi, B. Yu, and H. Zhang, *Decoupled kullback-leibler divergence loss*, arXiv preprint arXiv:2305.13948, May 2023. [Online]. Available: https://arxiv.org/abs/2305.13948

[23] G. Deng et al., *Masterkey: Automated jailbreak across multiple large language model chatbots*, arXiv preprint arXiv:2307.08715, Jul. 2023. DOI: 10.48550/arXiv.2307.08715

[24] N. Ding et al., *Enhancing chat language models by scaling high-quality instructional conversations*, 2023. arXiv: 2305.14233 [cs.CL].

[25] N. Jain et al., *Baseline defenses for adversarial attacks against aligned language models*, arXiv preprint arXiv:2309.00614, Sep. 2023. [Online]. Available: https://arxiv.org/abs/2309.00614

[26] D. Kang, X. Li, I. Stoica, C. Guestrin, M. Zaharia, and T. Hashimoto, *Exploiting programmatic behavior of llms: Dual-use through standard security attacks*, arXiv preprint arXiv:2302.05733, 2023. [Online]. Available: https://arxiv.org/abs/2302.05733

[27] W. Lian, B. Goodson, E. Pentland, A. Cook, C. Vong, and "Teknium", *Openorca: An open dataset of gpt augmented flan reasoning traces*, https://https://huggingface.co/datasets/Open-Orca/OpenOrca, 2023.

[28] A. Rao, S. Vashistha, A. Naik, S. Aditya, and M. Choudhury, *Tricking LLMs into disobedience: Formalizing, analyzing, and detecting jailbreaks*, arXiv preprint arXiv:2305.14965, May 2023. [Online]. Available: https://arxiv.org/abs/2305.14965

[29] Y. Wang, H. Li, X. Han, P. Nakov, and T. Baldwin, *Do-not-answer: A dataset for evaluating safeguards in LLMs*, arXiv preprint arXiv:2308.13387, Aug. 2023. [Online]. Available: https://arxiv.org/abs/2308.13387

[30] A. Wei, N. Haghtalab, and J. Steinhardt, *Jailbroken: How does LLM safety training fail?* arXiv preprint arXiv:2307.02483, Jul. 2023. [Online]. Available: https://arxiv.org/abs/2307.02483

[31] X. Wu, Y. Li, H. Zhang, and H. Sun, *Selective knowledge distillation for efficient and robust language models*, arXiv preprint arXiv:2305.14081, 2023. [Online]. Available: https://arxiv.org/abs/2305.14081

[32] J. Yu, X. Lin, and X. Xing, *Gptfuzzer: Red teaming large language models with auto-generated jailbreak prompts*, arXiv preprint arXiv:2309.10253v1, 2023. [Online]. Available: https://arxiv.org/abs/2309.10253v1

[33] L. Zheng et al., *Judging llm-as-a-judge with mt-bench and chatbot arena*, arXiv preprint arXiv:2306.05685, 2023. [Online]. Available: https://arxiv.org/abs/2306.05685

[34] A. Zou, Z. Wang, N. Carlini, M. Nasr, J. Z. Kolter, and M. Fredrikson, *Universal and transferable adversarial attacks on aligned language models*, arXiv preprint arXiv:2307.15043, Jul. 2023. [Online]. Available: https://arxiv.org/abs/2307.15043

[35] A. Zou, Z. Wang, J. Z. Kolter, and M. Fredrikson, *Universal and transferable adversarial attacks on aligned language models*, 2023. arXiv: 2307.15043 [cs.CL].

[36] P. Chao et al., "Jailbreakbench: An open robustness benchmark for jailbreaking large language models," in *NeurIPS Datasets and Benchmarks Track*, 2024.

[37] H. K. Choi, X. Du, and Y. Li, *Safety-aware fine-tuning of large language models*, arXiv preprint arXiv:2410.10014, Oct. 2024. [Online]. Available: https://arxiv.org/abs/2410.10014

[38] J. Chu, Y. Liu, Z. Yang, X. Shen, M. Backes, and Y. Zhang, *Jailbreakradar: Comprehensive assessment of jailbreak attacks against LLMs*, arXiv preprint arXiv:2402.05668, Feb. 2024. [Online]. Available: https://arxiv.org/abs/2402.05668

[39] S. Ganguly, R. Nayak, R. Rao, U. Deb, and P. Ap, *AdaKD: Dynamic knowledge distillation of ASR models using adaptive loss weighting*, arXiv preprint arXiv:2405.08019, May 2024. [Online]. Available: https://arxiv.org/abs/2405.08019

[40] Google and Hugging Face Datasets Contributors, *Google/jigsaw_toxicity_pred: Toxic comment classification dataset*, Hugging Face Dataset, 2024. [Online]. Available: https://huggingface.co/datasets/google/jigsaw_toxicity_pred

[41] Z. Guo, D. Wang, Q. He, and P. Zhang, "Leveraging logit uncertainty for better knowledge distillation," *Scientific Reports*, vol. 14, no. 1, p. 31 249, Dec. 2024. DOI: 10.1038/s41598-024-82647-6

[42] O. Gupta, M. De La Cuadra Lozano, A. Busalim, R. R. Jaiswal, and K. Quille, "Harmful prompt classification for large language models," in *Proceedings of the 2024 Conference on Human Centred Artificial Intelligence - Education and Practice (HCAIep '24)*, Dec. 2024, pp. 8–14. DOI: 10.1145/3701268.3701271

[43] T. Li et al., *Revisiting jailbreaking for large language models: A representation engineering perspective*, arXiv preprint arXiv:2401.06824, Jan. 2024. [Online]. Available: https://arxiv.org/abs/2401.06824

[44] J. Liu et al., *DDK: Distilling domain knowledge for efficient large language models*, arXiv preprint arXiv:2407.16154, Jul. 2024. [Online]. Available: https://arxiv.org/abs/2407.16154

[45] W. Luo, S. Ma, X. Liu, X. Guo, and C. Xiao, *Jailbreakv-28k: A benchmark for assessing the robustness of multimodal large language models against jailbreak attacks*, 2024. arXiv: 2404.03027 [cs.CR].

[46] M. Mazeika et al., *Harmbench: A standardized evaluation framework for automated red teaming and robust refusal*, arXiv preprint arXiv:2402.04249, Feb. 2024. [Online]. Available: https://arxiv.org/abs/2402.04249

[47] Meta AI, "The llama 3 herd of models," Meta AI, Tech. Rep., 2024. [Online]. Available: https://ai.meta.com/llama/

[48] A. Olteanu and J. Waples. "What is meta's llama 3.3 70b? how it works, use cases & more." Accessed: 2026-01-30, DataCamp. [Online]. Available: https://www.datacamp.com/blog/llama-3-3-70b

[49] T. Peng and J. Zhang, *Enhancing knowledge distillation of large language models through efficient multi-modal distribution alignment*, arXiv preprint arXiv:2409.12545, Sep. 2024. [Online]. Available: https://arxiv.org/abs/2409.12545

[50] prhegde and H. F. D. Contributors, *Prhegde/preference-data-math-stack-exchange: Math stack exchange preference dataset*, Hugging Face Dataset, 2024. [Online]. Available: https://huggingface.co/datasets/prhegde/preference-data-math-stack-exchange

[51] X. Shen, Z. Chen, M. Backes, Y. Shen, and Y. Zhang, ""do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, ACM, 2024.

[52] X. Xu et al., *A survey on knowledge distillation of large language models*, arXiv preprint arXiv:2402.13116, Feb. 2024. [Online]. Available: https://arxiv.org/abs/2402.13116

[53] Y. Xu and W. Wang, "Linkprompt: Natural and universal adversarial attacks on prompt-based language models," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 2024, pp. 6473–6486. [Online]. Available: https://aclanthology.org/2024.naacl-long.360/

[54] Z. Xu, Y. Liu, G. Deng, Y. Li, and S. Picek, *A comprehensive study of jailbreak attack versus defense for large language models*, arXiv preprint arXiv:2402.13457, Feb. 2024. [Online]. Available: https://arxiv.org/abs/2402.13457

[55] C. Yang et al., *Survey on knowledge distillation for large language models: Methods, evaluation, and application*, arXiv preprint arXiv:2407.01885, Jul. 2024. [Online]. Available: https://arxiv.org/abs/2407.01885

[56] S. Yi et al., *Jailbreak attacks and defenses against large language models: A survey*, arXiv preprint arXiv:2407.04295, Jul. 2024. [Online]. Available: https://arxiv.org/abs/2407.04295

[57] Y.-C. Yu et al., *Select and distill: Selective dual-teacher knowledge transfer for continual learning on vision-language models*, arXiv preprint arXiv:2403.09296, Mar. 2024. [Online]. Available: https://arxiv.org/abs/2403.09296

[58] W. Zhao et al., *Defending large language models against jailbreak attacks via layer-specific editing*, arXiv preprint arXiv:2405.18166, Jun. 2024. [Online]. Available: https://arxiv.org/abs/2405.18166v2

[59] X. Zhao et al., *Weak-to-strong jailbreaking on large language models*, arXiv preprint arXiv:2401.17256, Jan. 2024. [Online]. Available: https://arxiv.org/abs/2401.17256

[60] Y. Zhou, J. Lou, Z. Huang, Z. Qin, Y. Yang, and W. Wang, *Don't say no: Jailbreaking LLM by suppressing refusal*, arXiv preprint arXiv:2404.16369, Apr. 2024. [Online]. Available: https://arxiv.org/abs/2404.16369

[61] K. Chae, H. Jin, and T. Kim, *From threat to tool: Leveraging refusal-aware injection attacks for safety alignment*, arXiv preprint arXiv:2506.10020, Jun. 2025. [Online]. Available: https://arxiv.org/abs/2506.10020

[62] Y. S. Chan, N. Ri, Y. Xiao, and M. Ghassemi, *Speak easy: Eliciting harmful jailbreaks from LLMs with simple interactions*, arXiv preprint arXiv:2502.04322, Feb. 2025. [Online]. Available: https://arxiv.org/abs/2502.04322

[63] E. Contributors, *Qwen-2.5-7b-instruct: Overview of an instruction-tuned large language model*, EmergentMind Webpage, 2025. [Online]. Available: https://www.emergentmind.com/topics/qwen-2-5-7b-instruct

[64] S. Corbo, L. Bancale, V. De Gennaro, L. Lestingi, V. Scotti, and M. Camilli, "How toxic can you get? search-based toxicity testing for large language models," *IEEE Transactions on Software Engineering*, vol. 51, no. 11, pp. 3056–3071, Oct. 2025. DOI: 10.1109/TSE.2025.3607625

[65] X. Li, C. Zhang, J. Wang, F. Wu, Y. Li, and X. Jin, *Efficient and stealthy jailbreak attacks via adversarial prompt distillation from LLMs to SLMs*, arXiv preprint arXiv:2506.17231, Jun. 2025. [Online]. Available: https://arxiv.org/abs/2506.17231

[66] Z. Li, *Context-aware knowledge distillation with adaptive weighting for image classification*, arXiv preprint arXiv:2509.05319, Aug. 2025. [Online]. Available: https://arxiv.org/abs/2509.05319

[67] S. Saha, X. Li, M. Ghazvininejad, J. Weston, and T. Wang, *Learning to plan & reason for evaluation with thinking-llm-as-a-judge*, 2025. arXiv: 2501.18099 [cs.AI]. [Online]. Available: https://arxiv.org/abs/2501.18099

[68] G. Wang, Z. Yang, Z. Wang, S. Wang, Q. Xu, and Q. Huang, *ABKD: Pursuing a proper allocation of the probability mass in knowledge distillation via α-β-divergence*, arXiv preprint arXiv:2505.04560, May 2025. [Online]. Available: https://arxiv.org/abs/2505.04560

[69] J. Yang et al., *Feature alignment and representation transfer in knowledge distillation for large language models*, arXiv preprint arXiv:2504.13825, Apr. 2025. [Online]. Available: https://arxiv.org/abs/2504.13825

[70] S. Yi, T. Cong, X. He, Q. Li, and J. Song, *Beyond the tip of efficiency: Uncovering the submerged threats of jailbreak attacks in small language models*, arXiv preprint arXiv:2502.19883, Feb. 2025. [Online]. Available: https://arxiv.org/abs/2502.19883

[71] Z. Ying et al., *Reasoning-augmented conversation for multi-turn jailbreak attacks on large language models*, arXiv preprint arXiv:2502.11054, Feb. 2025. [Online]. Available: https://arxiv.org/abs/2502.11054

[72] S. Zhang et al., *Jbshield: Defending large language models from jailbreak attacks through activated concept analysis and manipulation*, arXiv preprint arXiv:2502.07557, Feb. 2025. [Online]. Available: https://arxiv.org/abs/2502.07557

[73] W. Zhang, H. Xu, Z. Wang, Z. He, Z. Zhu, and K. Ren, *Can small language models reliably resist jailbreak attacks? a comprehensive evaluation*, arXiv preprint arXiv:2503.06519, Mar. 2025. [Online]. Available: https://arxiv.org/abs/2503.06519