Doctoral Dissertation

Towards Quality, Reliability, and Efficiency in Large Multilingual and Multimodal Models via Answer-Prefix Generation and Efficient Inference

Le, Khang Nguyen

Supervisor NGUYEN, Minh Le

Graduate School of Advanced Science and Technology
Japan Advanced Institute of Science and Technology
(Information Science)

March, 2026

Abstract

Large Language Models (LLMs) have achieved remarkable success in natural language processing tasks, particularly in question-answering (QA). However, their integration into real-world systems is often hindered by two key challenges: first, the need for structured, concise answers and reliable confidence estimation to support aggregation across multiple reasoning paths; and second, the large model size and computational cost, which limit deployment efficiency.

To address these challenges, this thesis presents two complementary contributions. First, we introduce ANSPRE, a structured answer generation framework that guides LLMs to produce concise answers with reliable confidence scores. ANSPRE improves aggregation across retrievals and reasoning chains, enhances answer quality, and generalizes across multilingual and vision-language QA tasks. Extensive experiments on open-domain, multilingual, and visual QA benchmarks demonstrate that ANSPRE significantly improves Exact Match (EM) and F1 scores while providing well-calibrated confidence estimates.

Second, we investigate the behavior of pruning techniques across varying sparsity levels and identify key findings that inform optimal pruning strategies. Building on these insights, we propose OPTIPRUNE, a method that dynamically selects the most suitable pruning approach for each sparsity regime. Empirical evaluation shows that OPTIPRUNE consistently outperforms state-of-the-art pruning methods across multiple architectures, benchmarks, and language-specific calibrations, enabling efficient deployment without significant performance degradation.

Together, ANSPRE and OPTIPRUNE advance LLMs toward high-quality, reliable, and deployment-efficient systems, addressing critical gaps in structured reasoning, confidence estimation, and model compression.

[**Keywords**] Model Reliability, Model Efficiency, Retrieval-Augmented Generation, Model Pruning, Question-Answering, Large-Language-Model

Acknowledgement

I would like to express my deepest gratitude to my primary supervisor, Professor NGUYEN Minh Le, for his exceptional guidance, intellectual insight, and unwavering support throughout the course of my doctoral research. His profound expertise and scholarly rigor have greatly shaped my academic development and been indispensable to the completion of this dissertation. I am particularly grateful for his constructive feedback, inspiring discussions, and constant encouragement, which have enriched both my research and personal growth.

My sincere appreciation also goes to my second supervisor, Professor Kiyoaki Shirai, whose thoughtful feedback and expert advice have profoundly influenced the theoretical and methodological foundations of this work. His analytical perspective and commitment to academic excellence have been invaluable throughout my studies. I am equally grateful to Professor Shinobu Hasegawa, my supervisor for minor research, for his kind guidance and insightful comments, which have contributed meaningfully to the refinement of this dissertation.

I wish to acknowledge the Japan Advanced Institute of Science and Technology (JAIST) for providing an excellent academic environment and the necessary infrastructure that enabled me to pursue my research objectives effectively. I am thankful to the faculty members, administrative staff, and technical personnel whose professionalism and support have greatly facilitated my academic journey.

I would like to extend my heartfelt gratitude to the Ministry of Education, Culture, Sports, Science, and Technology (MEXT) of Japan for awarding me the prestigious MEXT Scholarship. This generous financial and institutional support made it possible for me to undertake my doctoral studies in Japan. Beyond its academic value, this opportunity has allowed me to engage with Japan's vibrant research community and experience its rich cultural heritage—an experience that has profoundly shaped my perspective as a scholar.

I am deeply thankful to my family, friends, and colleagues for their encouragement, patience, and understanding, which have sustained me through the challenges of this academic endeavor. Their unwavering belief in my abilities has been a constant source of motivation and strength.

Finally, I would like to express my sincere appreciation to all those who have contributed, directly or indirectly, to the completion of this dissertation. Their intellectual, moral, and practical support has been indispensable, and I remain profoundly grateful for their presence throughout this long and rewarding journey.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

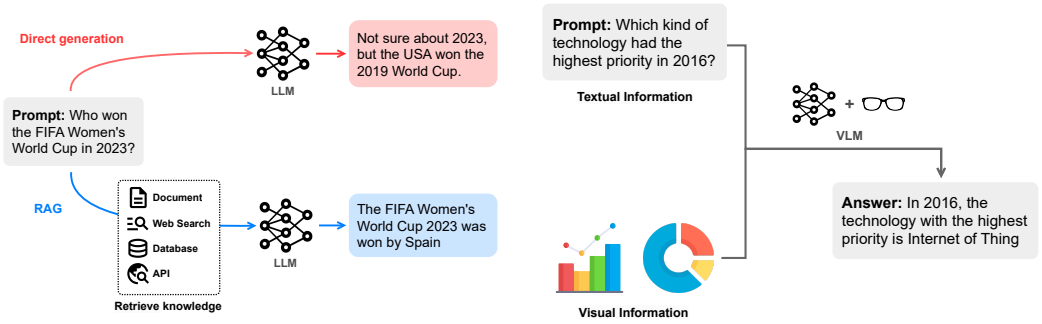
1.1 Background

1.1.1 Answer Quality and Reliability in LLM

Recent progress in Large Language Models (LLMs) has significantly advanced the field of Open-domain Question Answering (ODQA). Despite their remarkable capabilities, most LLMs rely solely on static pre-trained knowledge, which naturally becomes outdated as new information emerges. This limitation restricts their ability to answer time-sensitive or factual queries accurately. To mitigate this issue, retrieval-augmented approaches such as Retrieval-Augmented Generation (RAG) [43, 21] were proposed. RAG enhances question answering by retrieving relevant documents from an external knowledge source and conditioning the generation process on both the query and retrieved content. For example, when asked, "Who won the FIFA Women's World Cup in 2023?", an LLM trained prior to that year might respond incorrectly or acknowledge the lack of data. In contrast, a retrieval-augmented model can dynamically access updated information, enabling it to produce a correct and up-to-date answer. Figure 1.1a illustrates this difference, where a RAG-enabled system successfully retrieves and integrates relevant knowledge to identify the correct answer.

Beyond textual retrieval, LLMs have also been extended to Visual Question Answering (VQA), a task that combines linguistic understanding with visual perception. Similar in spirit to RAG—which retrieves textual evidence from a knowledge base—VQA integrates visual models with LLMs to extract information directly from images. Given an input question and an image, the model can reason over visual cues and textual context to produce an appropriate response. For instance, in Figure 1.1b, when asked, "Which kind of technology had the highest priority in 2016?", the model interprets

the visual chart to determine the correct answer. This demonstrates the growing potential of multimodal reasoning systems that unify textual and visual comprehension for more informed and context-aware answers.



(a) Retrieval-Augmented Generation (RAG) example.

(b) Visual Question Answering (VQA) example.

Figure 1.1: Examples of RAG and VQA systems.

Although such methods enrich the knowledge accessible to LLMs, they often lead to verbose outputs that obscure the essential answer. As shown in Figure 1.1, while both responses are factually correct, they can be more concisely expressed as "Spain" and "Internet of Things (IoT)." Consider the example in Figure 1.2, where the question "What gambling game, requiring two coins to play, was popular in World War I?" expects the concise answer "Two-Up." A conventional RAG pipeline, however, may generate an extended explanation around this answer, complicating its extraction. While instruction-tuned [79] or reinforcement-trained [62] LLMs can be prompted to produce shorter responses, their consistency varies considerably depending on the underlying model, prompt format, and instruction style.

Another important challenge in ODQA is the calibration of confidence scores. Reliable confidence estimation is critical in high-stakes applications such as healthcare, law, and finance, where incorrect predictions can have serious consequences. LLMs typically compute confidence based on token probabilities; however, prior studies [20, 30] have shown that these probabilities are often misaligned with actual answer correctness. This miscalibration undermines the dependability of model outputs, especially when integrating automated reasoning systems with structured databases or decision-making pipelines [15, 9].

To address these challenges, this study introduces **Answer-Prefix Generation** (ANSPRE), a novel framework designed to produce concise, well-calibrated answers from LLMs. The core idea of ANSPRE is to use explicit *answer prefixes* that guide the model to generate direct and precise responses.

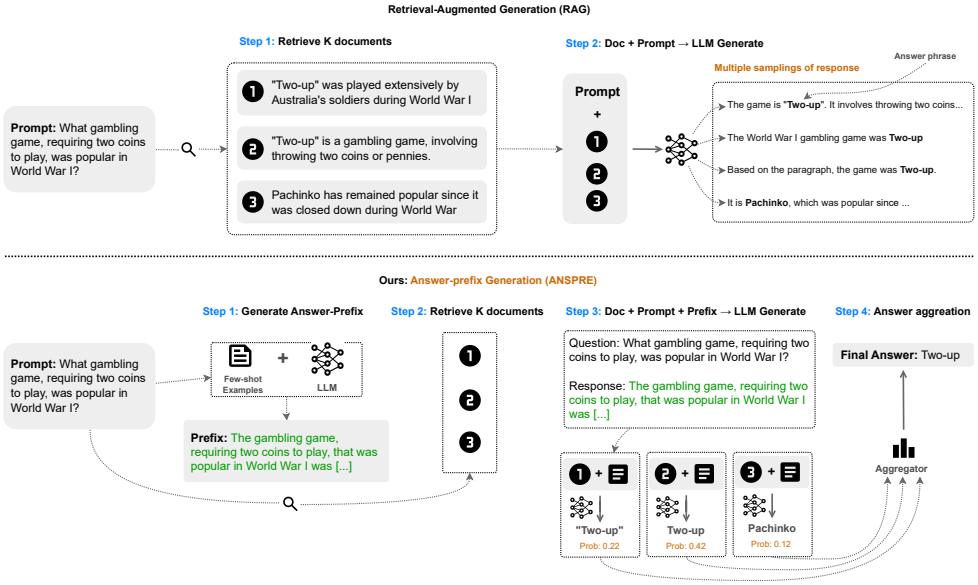


Figure 1.2: Overview of the proposed ANSPRE framework compared with the standard RAG pipeline.

As shown in Figure 1.2, the method proceeds in four stages: (1) generating a compact set of answer-prefix examples, (2) retrieving relevant evidence using standard RAG techniques, (3) composing an augmented prompt that integrates the question, retrieved passages, and answer prefix to guide the generation process, and (4) aggregating predictions and associated confidence scores across multiple retrievals. This structured pipeline improves both answer accuracy and confidence interpretability, making it suitable for practical, reliability-critical applications.

Furthermore, we extend this approach by integrating ANSPRE with Self-Reflective Retrieval-Augmented Generation (SELF-RAG) [4]. SELF-RAG introduces reflection tokens to assess the quality and relevance of retrieved information during generation. Our combined framework, termed SELF-ANSPRE, fuses the calibrated confidence mechanism from ANSPRE with the reflective retrieval strategy of SELF-RAG, enhancing both retrieval precision and response reliability. Figure 3.2 illustrates this integration, where SELF-ANSPRE unifies structured answer generation and retrieval-based reflection for improved performance across ODQA benchmarks.

1.1.2 Model Pruning

Large Language Models (LLMs) have achieved remarkable performance across a wide range of natural language processing tasks. However, their immense parameter counts and computational demands pose challenges for efficient deployment, especially in resource-constrained environments. To mitigate these issues, model compression techniques such as pruning have been extensively studied [38, 24, 59, 67, 18, 87, 84]. Model pruning aims to reduce model size and inference cost by removing redundant or less significant parameters. Depending on the granularity of the removed components, pruning methods can be categorized into three types: unstructured pruning (individual parameter removal), semi-structured pruning (removal under partial structural constraints), and structured pruning (removal of entire neurons, filters, or layers). Among these, *post-training pruning*—a one-shot approach that eliminates the need for retraining—has proven particularly practical for large-scale models, especially in unstructured and semi-structured forms. Given a target sparsity ratio, these methods iteratively remove parameters until the desired sparsity level is achieved (e.g., pruning 70% of parameters for 70% sparsity).

Recent state-of-the-art (SOTA) pruning methods generally adopt a layerwise strategy, where each layer is pruned independently to minimize the discrepancy between the outputs of the pruned and original models. Most of these approaches employ a uniform sparsity pattern, applying the same sparsity ratio across all layers. Pruning can be accomplished either by constructing a binary sparsity mask or by reconstructing weights based on second-order information such as the Hessian matrix. Mask-based methods typically rely on parameter magnitude [22] or relative importance estimation [87], whereas reconstruction-based approaches [45, 17, 18] optimize the remaining weights using Hessian-guided updates. In contrast, several studies have explored non-uniform sparsity, allowing the sparsity ratio to vary across layers while maintaining a fixed overall sparsity budget [16, 41, 76, 39, 49, 84]. Despite their success, our findings indicate that these methods exhibit performance trade-offs depending on the sparsity regime: non-uniform sparsity tends to be advantageous at higher compression ratios, while uniform sparsity performs better when pruning is moderate. Moreover, little prior work has examined how pruning affects multilingual generalization, leaving this an open area for further investigation.

1.2 Research Objectives

The overarching goal of this research is to advance the development of **high-quality, reliable, and deployment-efficient large language models (LLMs)**. While existing LLMs exhibit impressive capabilities, they often face challenges in generating concise and structured outputs, providing reliable confidence estimates, generalizing across modalities and languages, and maintaining efficiency under pruning. To address these limitations, this thesis pursues two major research directions, each corresponding to a core objective.

1.2.1 Objective 1: Toward High-Quality and Reliable Answer Generation

The first objective is to design a mechanism that enhances the answer quality, reliability, and adaptability of LLMs. Specifically, this research proposes a framework, termed ANSPRE, which aims to:

- Guide LLMs to produce **concise and structured answers**, improving interpretability and enabling aggregation of reasoning paths to enhance overall accuracy.
- Develop a method for **reliable confidence estimation**, ensuring that the model's self-assessed confidence aligns closely with its true performance.
- Evaluate the proposed mechanism's **generalizability** across multilingual and vision-language settings to confirm its robustness and cross-domain applicability.

1.2.2 Objective 2: Toward Deployment-Efficient Large Language Models

The second objective focuses on optimizing LLM efficiency through structured pruning analysis and improvement. This research investigates how various pruning strategies behave under different sparsity levels and derives actionable insights for optimal compression. Based on these findings, this thesis introduces OPTIPRUNE, an approach designed to:

- Analyze the effects of pruning across multiple sparsity regimes to understand performance trade-offs.
- **Leverage sparsity-dependent characteristics** to determine the most effective pruning method for each sparsity range.

- Achieve an optimal balance between model performance, efficiency, and deployability across diverse computational settings.

1.3 Research Contributions

This thesis makes several contributions toward advancing the quality, reliability, and efficiency of large language models (LLMs). The contributions are organized around two main research objectives: (1) improving answer generation and confidence estimation, and (2) enhancing model pruning for deployment efficiency.

1.3.1 High-Quality and Reliable Answer Generation

- **Structured Answer Generation:** We propose ANSPRE, a structured generation framework that guides LLMs to produce concise and well-structured answers. This approach improves interpretability and enables effective aggregation across multiple retrievals and reasoning paths, enhancing overall answer accuracy.
- **Reliable Confidence Estimation:** ANSPRE introduces a mechanism for deriving reliable confidence scores alongside generated answers. These scores align closely with model correctness, improving the trustworthiness of LLM predictions in high-stakes applications.
- **Integration into Retrieval-Augmented Generation:** We extend the self-reflective RAG framework by proposing SELF-ANSPRE, which integrates reflection token scores and answer-prefix supervision. This design enhances both the quality of generated answers and the precision of the retrieval process.
- **Multilingual and Multimodal Adaptation:** We adapt ANSPRE to multilingual and multimodal question answering tasks, achieving substantial improvements on Vietnamese and Japanese QA benchmarks, as well as on visual question answering (VQA) datasets. This demonstrates the generality and adaptability of the approach.
- **Comprehensive Evaluation:** Extensive experiments across open-domain QA, multilingual QA, and VQA tasks using diverse LLM families show that ANSPRE significantly improves Exact Match (EM) and F1 scores. Moreover, it provides better-calibrated confidence scores, making it suitable for applications requiring reliable reasoning and factual consistency.

1.3.2 Deployment-Efficient LLMs via Pruning

- **Empirical Findings on Pruning Behavior:** We present and validate two key findings through extensive empirical studies on state-of-the-art pruning techniques. These findings reveal how different pruning strategies behave across varying sparsity levels and guide the design of more effective pruning mechanisms.

- **Proposed Method – OptiPrune:** Building on these insights, we introduce OptiPrune, a method designed to optimally prune LLMs across a wide range of target sparsity levels. The approach dynamically selects the most suitable pruning strategy based on observed sparsity-dependent characteristics.

- **Superior Empirical Performance:** Experimental results demonstrate that OptiPrune consistently outperforms state-of-the-art pruning methods across multiple model architectures, sparsity regimes, and evaluation benchmarks. Furthermore, it maintains strong multilingual performance after language-specific calibration, highlighting its robustness and practical deployability.

Chapter 2

Related Work

2.1 Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) has emerged as a key framework for addressing knowledge-intensive tasks by integrating external information sources into the response generation process [43]. The central idea is to enable large language models (LLMs) to retrieve relevant evidence before or during text generation, thereby enhancing factuality and grounding the model's outputs.

Several studies have proposed different strategies for incorporating retrieval into LLMs. One approach focuses on instruction-tuning with a fixed number of retrieved passages to guide the model toward better-informed responses [50]. Another line of research jointly trains models to perform both retrieval and response generation, allowing them to learn the relationship between retrieved context and output quality in an end-to-end manner [28]. More recent advances explore dynamic retrieval mechanisms, where the model actively queries external knowledge during the decoding process to refine its reasoning and answer accuracy [31]. Similarly, systems such as Toolformer train models to decide when to issue API calls for external data sources, enabling more context-aware information retrieval [65].

A notable advancement is the Self-RAG framework, which introduces a self-reflective mechanism allowing models to determine when retrieval is necessary and to evaluate both retrieved content and generated answers. This method has achieved state-of-the-art results on multiple open-domain question-answering (ODQA) benchmarks, demonstrating the potential of self-assessment for enhancing RAG performance.

RAG-based methods have also been tailored for specialized domains such as medicine and fact-checking. For instance, TrumorGPT [23] addresses the

problem of health-related misinformation by integrating LLMs with graph-based RAG over dynamic health knowledge graphs, thus enabling real-time, factual public health verification. Likewise, [6] proposed a clinical decision-support system that combines open-source LLMs with RAG in the Obstetrics and Gynecology (OBGYN) domain, leveraging Bio-Mistral-7B [36] and medical embeddings to improve factual accuracy and relevance. Despite these achievements, most RAG systems still overlook the importance of explicitly identifying concise answer phrases within verbose model outputs and have not adequately addressed the calibration of model confidence scores.

2.2 Question Answering

Question Answering (QA) remains one of the central challenges in Natural Language Processing (NLP), with broad applications in knowledge retrieval, dialogue systems, and digital assistants. The rise of LLMs has shifted research attention toward improving the factuality, interpretability, and faithfulness of QA systems. Recent studies have introduced methods to mitigate common issues such as retrieval noise and hallucination. For example, the retriever–generator–verifier framework proposed in [68] improves QA accuracy by verifying the consistency between retrieved evidence and generated answers. Other work has focused on enhancing robustness by masking unstable knowledge during machine reading comprehension [44].

In domain-specific QA, approaches such as fine-tuning with hard-sample-aware optimization have been shown to effectively reduce hallucinations in legal QA systems [27]. Additionally, evaluation metrics have been refined to better align with human judgment, emphasizing correctness and factuality in instruction-following models [1]. For low-resource languages, notable progress has been made in developing efficient QA systems that achieve strong performance with minimal supervision; a Vietnamese open-domain QA model, for example, demonstrated impressive results using compact training data [60]. Beyond textual QA, advancements in Visual Question Answering (VQA) have also emerged, such as the use of bias-guided margin loss to mitigate language priors and improve visual reasoning accuracy [69].

2.3 Re-ranking and Aggregation

Re-ranking and aggregation are crucial components for improving the precision and reliability of model-generated responses. Re-ranking involves reordering candidate outputs to select the most relevant or accurate answer,

while aggregation combines multiple generated hypotheses or reasoning paths into a unified prediction. With the rise of LLMs, several new aggregation techniques have been proposed to exploit the diversity of model outputs.

A prominent example is the Self-Consistency approach [78], which generates multiple reasoning paths using the Chain-of-Thought (CoT) paradigm [80] and then marginalizes over the final answers to produce a consensus result. Although effective in structured reasoning tasks such as mathematics, Self-Consistency often struggles in open-ended QA tasks, where the reasoning process is less deterministic.

Building upon this foundation, our proposed aggregation strategy enhances the reliability and interpretability of QA results by leveraging the answer-prefix generation capability of ANSPRE. Instead of merely voting across multiple sampled generations, our approach identifies and aggregates precise answer phrases extracted from diverse retrieval contexts. Furthermore, this method enables multi-level aggregation — both within the most relevant retrieved sources and across all available external documents — leading to more robust, consistent, and trustworthy responses in RAG-based QA tasks.

Chapter 3

ANSPRE Generation

3.1 Problem Formulation and Overview

3.1.1 Problem Formulation

Given a natural language question, such as the one shown in Figure 1.2, the objective is to employ a RAG pipeline integrated with a LLM to produce both the correct answer phrase (e.g., "Two-up") and an associated confidence score that quantifies the model's certainty in that answer. The key goal is to ensure that the generated output is not only accurate but also well-calibrated in terms of the model's self-assessed reliability.

Formally, let a question be denoted as Q, and let $\mathcal{M}$ represent a pre-trained large language model. A document retriever first identifies a collection of k relevant documents $\mathcal{D} = \{D_1, D_2, \ldots, D_k\}$ that may contain supporting evidence for answering Q. The model $\mathcal{M}$ then takes Q and $\mathcal{D}$ as input and generates a textual response Y, along with a confidence estimate reflecting how certain the model is about its output.

The generation process of $\mathcal{M}$ can be mathematically formulated as a sequence of conditional next-token predictions:

$$P_{\mathcal{M}}(y_t \mid X, y_{<t}),$$

where X denotes the input prompt (which may include both Q and the retrieved documents $\mathcal{D}$), and y_t is the token predicted at time step t. Consequently, the probability of generating the full sequence $Y = [y_1, y_2, \ldots, y_T]$ is given by:

$$P_{\mathcal{M}}(Y \mid X) = \prod_{t=1}^{|Y|} P_{\mathcal{M}}(y_t \mid X, y_{<t}).$$

In the context of RAG, this conditional probability becomes $P_{\mathcal{M}}(Y \mid Q, D)$, where the model conditions its next-token predictions not only on the question but also on the retrieved evidence documents. In practical implementations, the cumulative *log-probability* of the generated sequence is typically used to improve numerical stability during computation.

However, a notable limitation of conventional RAG approaches is the weak correlation between the sequence probability $P_{\mathcal{M}}(Y \mid Q, D)$ and the model's actual confidence in the correctness of the answer phrase. This discrepancy leads to situations where a model might assign a high probability to an incorrect or hallucinated answer. The ANSPRE framework is designed to mitigate this issue by introducing mechanisms for aligning model confidence with prediction accuracy.

3.1.2 Background on Calibration

In probabilistic modeling, calibration measures the alignment between predicted confidence and empirical accuracy. For example, a model with a confidence score of 0.6 should be correct approximately 60% of the time. A perfectly calibrated model satisfies:

$$P(\hat{Y} = Y \mid f_{\text{con}}(\hat{Y}) = p) = p, \quad \forall p \in [0, 1],$$

where $f_{\text{con}}(\hat{Y})$ denotes the predicted confidence of answer $\hat{Y}$. To quantitatively evaluate calibration, the *Expected Calibration Error (ECE)* [20] is widely used. ECE partitions predictions into M discrete confidence intervals (or "bins") and computes the average absolute difference between accuracy and confidence across all bins:

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} \left|\text{acc}(B_m) - \text{con}(B_m)\right|,$$

where B_m represents the m-th bin, $\text{acc}(B_m)$ is the empirical accuracy of predictions in that bin, $\text{con}(B_m)$ is the average confidence score, and n is the total number of predictions. Lower ECE values indicate better calibration.

3.1.3 Instruction-tuning and Few-shot Approaches

Various strategies have been developed for improving the answer-generation capability of LLMs. Few-shot prompting [63] leverages example-based guidance directly within the prompt to help the model infer task patterns. Meanwhile, instruction-tuning [79] and reinforcement learning with human feedback (RLHF) [62] adapt LLMs through large-scale supervised fine-tuning.

However, these approaches face practical limitations in the RAG setting. Few-shot prompting becomes less feasible as input length increases, particularly when multiple retrieved passages must be included. Similarly, instruction-tuning or RLHF requires substantial amounts of labeled data, which are often unavailable for specialized domains or low-resource languages.

3.1.4 Overview

The proposed ANSPRE framework, illustrated in Figure 1.2 (Bottom), enhances LLM-based answer generation by introducing an *answer prefix* mechanism. Since causal LLMs are inherently trained to predict the next token in a sequence, appending an answer prefix to the input prompt effectively guides the model to anticipate the correct answer phrase as the immediate continuation. The cumulative log-probability of the generated tokens can then be interpreted as a natural measure of confidence for that answer.

Importantly, ANSPRE requires no additional fine-tuning or retraining, allowing it to be seamlessly integrated with existing pre-trained LLMs—whether monolingual, multilingual, or multimodal (e.g., vision-language models). This design makes it both scalable and adaptable. The subsequent subsection details how the answer prefix is constructed and how the method is incorporated into the Self-Reflective RAG system to further enhance both accuracy and calibration.

3.2 Generating Answer Prefix

3.2.1 Generation Method

Consider a question such as *"What is the capital of France?"*. Its declarative counterpart would be *"The capital of France is [ANSWER]"*, where the token *[ANSWER]* denotes the expected response. The segment preceding this placeholder, i.e., *"The capital of France is"*, is referred to as the *answer prefix*. Generating such prefixes enables the model to begin an answer in a natural, declarative form that aligns with grammatical conventions in English.

To produce these prefixes automatically, we employ a large language model (LLM) that is prompted to convert interrogative questions into their corresponding declarative forms. The prompting process follows a few-shot strategy, where the LLM is conditioned on a small set of manually curated examples that cover diverse *wh*-types (e.g., *who*, *what*, *where*, *when*, *why*, *which*, *whose*, and *how*). Each example demonstrates how an interrogative

question can be rewritten into a declarative sentence containing the placeholder *[ANSWER]* at the answer position.

For English, we construct three representative examples per question type, yielding a total of 24 few-shot examples. These examples are reused during inference to maintain stylistic consistency and guide the model toward generating fluent prefixes. During deployment, for each input question, the same few-shot context is concatenated with the new question, and the LLM is instructed to output a declarative reformulation ending with the token *[ANSWER]*. For instance, when given the question *"What is the largest ocean on Earth?"*, the model produces the prefix *"The largest ocean on Earth is "*, which subsequently serves as the first part of the final answer.

The transformation is prompted using the following instruction: *"Given an interrogative sentence: "{question}", and assuming the answer to this question is [ANSWER], convert the sentence into its declarative form."* Once the LLM $\mathcal{M}$ outputs the placeholder *[ANSWER]*, the process terminates, and the preceding text is extracted as the answer prefix E. Details of the few-shot examples used for English are provided in Table 3.1.

To verify the quality of the generated prefixes, we randomly sampled 100 questions from the TriviaQA dataset [32] and generated corresponding prefixes using the proposed method. The results were evaluated by GPT-4o, which judged **96.23%** of the generated prefixes as syntactically and semantically correct, confirming the reliability of our prefix generation process.

3.2.2 Interrogative/Declarative Transformations

To effectively train the model to generate appropriate *answer prefixes*, it is essential to provide high-quality few-shot examples that illustrate the transformation from interrogative to declarative sentence structures. These examples enable the LLM to learn the linguistic patterns necessary to generate declarative statements that naturally precede an answer. Tables 3.1 and 4.2 present representative samples of these transformations in English and Vietnamese, respectively.

English Examples. Table 3.1 demonstrates the process of converting English interrogative sentences into their declarative counterparts across various *wh*-question types, including *who*, *what*, *where*, *when*, *why*, *which*, *whose*, and *how*. Each example is paired with its corresponding declarative form, in which the placeholder *[ANSWER]* indicates the position of the anticipated response. For instance, the question *"Who wrote the 1975 book 'Superwoman'?"* is reformulated as *"The author of the 1975 book 'Superwoman' was*

[ANSWER]." This explicit reformulation allows the model to learn the grammatical and syntactic alignment between question types and their declarative realizations.

These examples serve as the few-shot context provided to the language model during inference. By generalizing from these patterns, the model can automatically produce grammatically coherent answer prefixes for unseen questions. This process supports cross-question generalization while maintaining fluency and syntactic correctness.

3.3 Answer Phrase Generation

Given the answer prefix E produced in the previous step, the next objective is to generate the full answer phrase conditioned on both the prefix and the supporting documents. For each retrieved document D_i, the LLM predicts the next token according to the conditional probability distribution:

$$P_{\mathcal{M}}(y_t \mid Q \oplus E, D_i, y_{<t}),$$

where $\oplus$ denotes sequence concatenation. Generation proceeds until the model outputs a termination token or punctuation, indicating the end of the phrase. To ensure diversity and robustness, beam search with beam width B is applied to generate multiple candidate sequences, from which the top N are retained for aggregation in subsequent steps.

To effectively combine parametric knowledge encoded in the LLM and non-parametric information retrieved from external sources, we define two complementary scoring functions. The first,

$$\mathcal{S}_{\text{Phrase}} = P_{\mathcal{M}}(Y \mid Q \oplus E, D_i),$$

represents the likelihood of the generated phrase given the retrieved evidence, thereby capturing non-parametric knowledge. The second score,

$$\mathcal{S}_{\text{Sentence}} = P_{\mathcal{M}}(E \oplus Y),$$

measures the intrinsic probability of the complete declarative sentence within the model's internal distribution, reflecting its parametric knowledge.

The final ANSPRE score integrates these two components through a weighted combination:

$$\mathcal{S}_{\text{ANSPRE}} = w_{\text{Phrase}} \cdot \mathcal{S}_{\text{Phrase}} + w_{\text{Sentence}} \cdot \mathcal{S}_{\text{Sentence}},$$

where the weights w_{Phrase} and w_{Sentence} are predefined hyperparameters satisfying $w_{\text{Phrase}} + w_{\text{Sentence}} = 1$. This formulation balances factual grounding from retrieved evidence with the fluency and coherence inherent to the model's own linguistic knowledge.

Type	Interrogative Form	Declarative Form
Who	Who became Heavyweight world boxing champion after Riddick Bowe refused to fight him in 1992?	The Heavyweight world boxing champion after Riddick Bowe refused to fight him in 1992 was [ANSWER].
	Who in 1807 & 1808 was the first to discover and isolate the metals sodium, potassium and calcium?	In 1807 & 1808, the first to discover and isolate the metals sodium, potassium, and calcium was [ANSWER].
	Who wrote the 1975 book 'Superwoman'?	The author of the 1975 book 'Superwoman' was [ANSWER].
What	What is the moniker of the mustache growing month of November?	The moniker of the mustache-growing month of November is [ANSWER].
	What distance is an Olympic steeplechase?	An Olympic steeplechase covers a distance of [ANSWER].
	What is the English name for the Japanese Parliament?	The English name for the Japanese Parliament is [ANSWER].
Where	Where would you find the Machu Picchu mountain?	The Machu Picchu mountain is located [ANSWER].
	Where did Horatio Nelson lose his right arm?	Horatio Nelson lost his right arm at [ANSWER].
	Where did Malcolm Campbell set a land speed record of 245.7 m.p.h. in 1931?	Malcolm Campbell set a land speed record of 245.7 m.p.h. in 1931 at [ANSWER].
When	When did field hockey become an Olympic event for men?	Field hockey became an Olympic event for men in [ANSWER].
	When did people last walk on the moon?	People last walked on the moon in [ANSWER].
	When eating out, what French phrase is effectively the opposite of 'a la carte'?	When eating out, the French phrase effectively opposite of 'a la carte' is [ANSWER].
Why	Why did Canadian George Lyon refuse to accept his gold medal for golf at the 1908 Olympics?	Canadian George Lyon refused to accept his gold medal for golf at the 1908 Olympics because [ANSWER].
	Why did the chicken cross the road?	The reason why the chicken crossed the road was [ANSWER].
	Why was Carol Hersey once the most popular seen person on British TV?	Carol Hersey was once the most popular seen person on British TV because [ANSWER].
Which	Which British Prime Minister was a cousin of Rudyard Kipling?	The British Prime Minister who was a cousin of Rudyard Kipling was [ANSWER].
	Which bird appears as a supporter on the Australian coat of arms?	The bird that appears as a supporter on the Australian coat of arms is [ANSWER].
	Which group had a Top Ten hit in 1986 with 'Hunting High And Low'.	The group that had a Top Ten hit in 1986 with 'Hunting High And Low' was [ANSWER].
Whose	Whose first two hit albums were No Parlez and The Secret of Association?	The first two hit albums, No Parlez and The Secret of Association, belong to [ANSWER].
	Whose picture is on the 2 dollar bill?	The picture on the 2 dollar bill belongs to [ANSWER].
	Whose first novel was Buddenbrooks, published in 1901?	The first novel Buddenbrooks, published in 1901, belongs to [ANSWER].
How	How many times is the f-word used in the film Goodfellas?	The number of the f-word used in the film Goodfellas is [ANSWER].
	How many points does a snow flake have?	The number of points a snowflake has is [ANSWER].
	How was the bird Pinguinus impennis commonly known?	The bird Pinguinus impennis commonly known [ANSWER].

Table 3.1: English question-answer pairs showing transformations from interrogative to declarative form.

3.4 Answer Aggregation

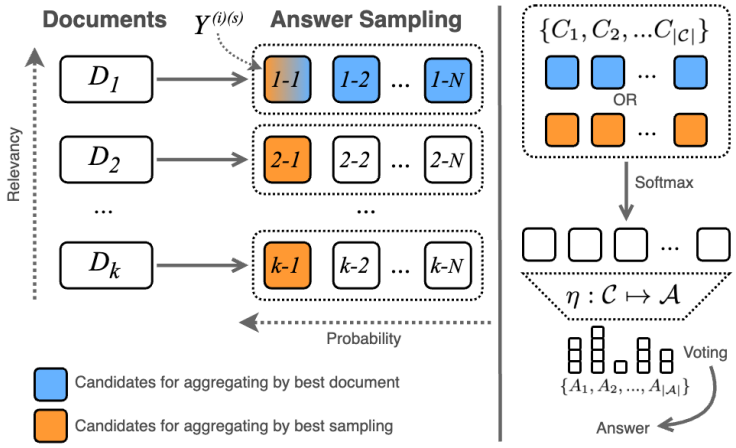


Figure 3.1: Overview of the proposed answer aggregation process.

To derive a single, coherent final answer from multiple candidate phrases, we design an *answer aggregation* mechanism. This module consolidates diverse answer candidates—produced during the answer phrase generation stage—by assigning each candidate a confidence score and then merging semantically equivalent ones. Figure 3.1 presents an overview of this process.

Each generated candidate is first associated with a final confidence score that reflects both parametric and non-parametric evidence (see Section 3.3). These scores are normalized using a softmax operation to ensure comparability across candidates. Subsequently, candidates expressing equivalent or paraphrased content are grouped together through a normalization function. Within each group, the cumulative probability mass serves as an aggregate measure of support for that answer variant. The group with the highest aggregate score is then selected as the final system output.

Formally, let $\mathcal{C} = \{C_1, C_2, \ldots, C_{|\mathcal{C}|}\}$ denote the complete set of candidate answer phrases, and let $\mathcal{A} = \{A_1, A_2, \ldots, A_{|\mathcal{A}|}\}$ represent the set of normalized (or canonical) answer forms. We define a surjective mapping $\eta : \mathcal{C} \to \mathcal{A}$ that standardizes candidate text by applying operations such as lowercasing, punctuation removal, and minimal text normalization. Given that $\mathcal{S}(C_j)$ denotes the unnormalized confidence score assigned to candidate C_j, the normalized aggregated score for each canonical answer A_i is computed as:

$$\text{Score}(A_i) = \sum_{\eta(C_j)=A_i} \frac{e^{\mathcal{S}(C_j)}}{\sum_{C \in \mathcal{C}} e^{\mathcal{S}(C)}}.$$

The canonical answer $\hat{A}$ corresponding to the highest aggregated score, i.e., $\hat{A} = \arg\max_{A_i} \text{Score}(A_i)$, is returned as the final prediction.

To explore different aggregation strategies, we propose two configurations for constructing the candidate set $\mathcal{C}$ based on the retrieval and decoding stages. Suppose the retrieved documents are denoted by $\mathcal{D} = \{D_1, D_2, \ldots, D_k\}$, ranked by retrieval relevance such that D_1 is the most relevant. During answer generation, beam search is employed with width B, producing N candidate samples per document. Let $Y^{(i)(s)}$ denote the s-th candidate generated from document D_i, ranked in descending order of likelihood (i.e., $Y^{(i)(1)}$ is the most probable).

- **Top-N from the most relevant document:** In the first configuration, we take the top N candidates solely from the highest-ranked document, forming the candidate set

  $$\mathcal{C} = \{Y^{(1)(s)} \mid 1 \leq s \leq N\}.$$

  This strategy prioritizes precision by relying exclusively on the most relevant context.

- **Top-1 from each retrieved document:** The second configuration aims to improve diversity by selecting only the best candidate from each retrieved document, constructing

  $$\mathcal{C} = \{Y^{(i)(1)} \mid 1 \leq i \leq k\}.$$

  This approach enables the model to aggregate evidence across multiple relevant documents and mitigates overreliance on a single retrieval result.

By combining these candidate-level strategies with score-based aggregation, our method effectively balances the trade-off between accuracy and diversity in open-domain answer generation.

3.5 Adaptation to Self-Reflective RAG

Self-Reflective Retrieval-Augmented Generation (Self-Rag) [4] has demonstrated state-of-the-art results across open-domain question-answering (ODQA) benchmarks. It enhances LLM-based retrieval systems by allowing dynamic document retrieval during text generation and by introducing self-critique tokens that guide reflection and evaluation. Building upon this paradigm, we

propose Self-Anspre, an extension that integrates our Anspre methodology into the Self-Rag framework (Figure 3.2, right). This integration introduces structured answer prefix generation and refined critique-based scoring mechanisms.

The general workflow of Self-Anspre follows the Self-Rag pipeline, with two principal modifications: (1) an additional step for generating an *answer prefix* before answer generation, and (2) the incorporation of the Anspre score into the reflective critique phase.

During the initial retrieval stage, the model determines whether external documents are needed by producing special reflection tokens (*[Retrieve]* or *[No retrieve]*). If the model decides that retrieval is unnecessary, the answer is generated using only the question and the answer prefix, following the standard Anspre approach described in Section 3.2. Conversely, if retrieval is triggered, the model retrieves k relevant documents using an external retriever. These documents are then passed to the LLM for evidence-aware generation.

In the response generation stage, Self-Rag traditionally embeds reflection tokens that assess aspects such as relevance, support, and usefulness of the retrieved evidence. Self-Anspre extends this mechanism by prepending the *answer prefix* to every generated response, thereby encouraging more consistent and structured output. The segment between the prefix and the next reflection token is extracted as the *answer phrase*, for which the Anspre score $\mathcal{S}_{\text{Anspre}}$ is computed following Section 3.3.

In the final critique and ranking phase, Self-Rag employs three categories of critic tokens—IsRel, IsSup, and IsUse—to evaluate the quality of generated answers. During inference, beam search explores multiple trajectories, and the final response is selected based on a weighted sum of the normalized critic token probabilities:

$$\mathcal{S}_{\text{critique}} = \sum_{G \in \mathcal{G}} w^G s_t^G,$$

where $\mathcal{G} = \{\text{IsRel}, \text{IsSup}, \text{IsUse}\}$ denotes the set of critic categories, s_t^G represents the probability of producing the correct critic token of type G, and w^G is the associated weight.

To integrate Anspre into this process, we extend the critique score to include the Anspre score as an additional weighted term:

$$\mathcal{S}_{\text{critique}} = \sum_{G \in \mathcal{G}} w^G s_t^G + w^A \mathcal{S}_{\text{Anspre}},$$

where w^A is a hyperparameter controlling the contribution of the Anspre

score. The candidate with the highest combined score is ultimately selected as the system's final response.

By integrating structured prefix generation and ANSPRE-based scoring into the self-reflective reasoning framework, SELF-ANSPRE achieves more reliable and interpretable answer synthesis. This combination improves factual precision and promotes a more principled evaluation of generated responses in open-domain QA tasks.

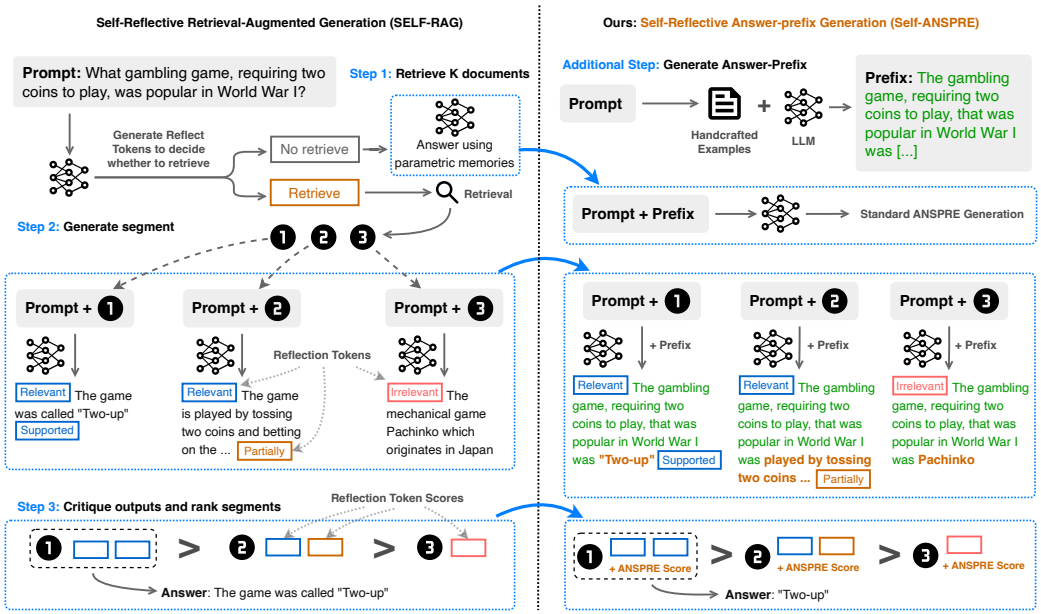


Figure 3.2: Comparison between the original SELF-RAG framework and the extended SELF-ANSPRE approach. Key modifications in SELF-ANSPRE are highlighted on the right.

Chapter 4

ANSPRE Adaptation to Multilingual Settings

4.1 Overview

The ANSPRE framework is designed to be language-agnostic, enabling adaptation to a wide range of linguistic systems beyond English. Extending this paradigm to multilingual environments primarily involves addressing the structural and grammatical variations that influence question formation and declarative transformations. By tailoring the prefix-generation step to account for language-specific interrogative markers and syntax, ANSPRE can maintain its effectiveness across typologically diverse languages.

To illustrate this adaptability, we conduct two case studies focusing on Vietnamese and Japanese—languages that represent distinct linguistic families and orthographic systems. Vietnamese employs a Latin-based alphabet with analytic grammar, while Japanese integrates multiple scripts (Kanji, Hiragana, and Katakana) within a syntactically agglutinative structure. These two languages pose different challenges for constructing answer prefixes and question-declarative mappings. Following the general approach outlined in Section 3.2, we apply the same methodological pipeline to generate answer prefixes while incorporating language-specific adjustments to reflect each language's grammatical and syntactic patterns.

4.2 Vietnamese

For Vietnamese, the few-shot examples were developed by a native speaker to ensure grammatical and semantic correctness. The design of the examples was informed by prior research on Vietnamese machine reading comprehen-

sion, particularly the VIMQA dataset [37]. Table 4.2 summarizes the central Vietnamese question words (CQs) and their English equivalents.

Vietnamese grammar shares many surface similarities with English, such as subject–verb–object (SVO) word order and limited inflectional morphology. As a result, the procedure for generating declarative forms closely mirrors that of English. For example, given the question *"Hồ Hoàn Kiếm trước đây còn được gọi là gì?"* ("What was Hoàn Ki´m Lake previously called?"), the declarative transformation becomes *"Hồ Hoàn Kiếm trước đây còn được gọi là [ANSWER]."* ("Previously, Hoàn Ki´m Lake was called [ANSWER]."), where *[ANSWER]* marks the placeholder for the final answer.

To achieve comprehensive linguistic coverage, we include examples for all major Vietnamese question forms, including wh-questions ("ai"(who), "gì"(what), "ở đâu"(where), "khi nào"(when), "tại sao/vì sao"(why)) as well as yes–no interrogatives such as "phải không" and "đúng không". Each question type is accompanied by one or two declarative examples, leading to a total of 24 few-shot examples. These examples guide the model in converting interrogative structures into declarative templates with the correct placement of the answer placeholder. Details of all examples and their transformations are presented in Table 4.2.

Group	**English CQW**	**Vietnamese CQW**
Yes/No	Copulas (is, are) Auxiliaries (does, did)	Phải không, Đúng không
Which	Which	Nào
What	What What ordinal number	Là gì Thứ mấy, Thứ bao nhiêu
Who	Who By whom	Ai Bởi ai
How	How many How often How long How far	Bao nhiêu Bao lâu một lần Bao lâu Bao xa
When	When	Khi nào
Where	Where	Ở đâu, Tại đâu
Why	Why	Vì sao, Tại sao

Table 4.1: List of Vietnamese central question words (CQs) used in few-shot prefix generation.

4.3 Japanese

Unlike the Vietnamese case, where we relied on native expertise, our Japanese setup does not involve native speaker curation. Instead, we draw on publicly available Japanese QA datasets—JAQKET [70] and JaQuAD [66]—to design few-shot examples representative of common question types. While the coverage is not exhaustive, it sufficiently captures frequent interrogative structures used in factual QA.

Japanese differs from English and Vietnamese in several key respects: (1) word order follows a subject–object–verb (SOV) structure, (2) particles mark grammatical relations rather than fixed word positions, and (3) varying levels of politeness affect question endings. Despite these challenges, the declarative transformation process can still be modeled effectively with minor modifications to word order and particle placement.

For instance, consider the question "東京タワーはいつ建てられましたか？" ("When was Tokyo Tower built?"). A direct declarative transformation would yield "東京タワーは[ANSWER]に建てられました。" ("Tokyo Tower was built in [ANSWER]."). However, to ensure that the answer token appears naturally at the end of the sentence, we adjust the declarative form to "東京タワーが建てられたのは[ANSWER]。", where *[ANSWER]* occupies the final syntactic slot, consistent with Japanese discourse norms.

For prefix generation, the model $\mathcal{M}$ is provided with approximately two few-shot examples per interrogative type, resulting in 25 total examples covering question forms such as "誰" (who), "何" (what), "どこ" (where), "いつ" (when), "なぜ" (why), "どうやって" (how), and others. Although these examples were not verified by native speakers, they were manually reviewed to maintain grammatical validity and natural word order.

Overall, this multilingual extension demonstrates that the ANSPRE framework is not restricted to English-based QA systems. By incorporating linguistic and morphological adjustments in the prefix generation step, it can be effectively applied to low-resource or morphologically rich languages, facilitating a more inclusive and language-agnostic approach to open-domain question answering.

4.4 Adaptation Details

Vietnamese Examples. Table 4.2 provides illustrative examples for Vietnamese, a low-resource language with structural similarities to English but unique morphological and syntactic features. Each interrogative sentence is paired with its declarative transformation, ensuring that the placeholder

[ANSWER] appears at the syntactically appropriate position. For example, the question "Hồ Hoàn Kiếm trước đây còn được gọi là gì?" ("What was Ho Hoan Kiem previously called?") becomes "Hồ Hoàn Kiếm trước đây còn được gọi là [ANSWER]." ("Ho Hoan Kiem was previously called [ANSWER].").

These examples were designed by a native Vietnamese speaker to ensure grammatical accuracy and natural phrasing. They comprehensively cover a wide range of interrogative constructions, including yes/no questions, "Ai" (who), "Ở đâu" (where), "Bao nhiêu" (how many), and "Vì sao" (why). Such linguistic diversity enables the model to generalize across various syntactic templates during multilingual adaptation.

4.4.1 List of Omitted Copulas

During the construction of Japanese examples, several copulas were intentionally omitted to maintain grammatical simplicity and avoid ambiguity. The following copular forms were excluded from the dataset: "です。", "である。", "でした。", "というでしょう。", "という。", "になるでしょう。", "と言います。", "と言うでしょう。", "。"

This list does not represent an exhaustive set of Japanese copulas but rather the subset most frequently encountered in formal declarative expressions. A more detailed linguistic exploration of copular variation in Japanese could further refine declarative generation, potentially improving cross-linguistic generalization in future work.

Type	Interrogative Form	Declarative Form
Có/Không (Yes/No)	Diego Maradona nhỏ tuổi hơn Rutherford B. Hayes phải không? Đội bóng của Nathan Dyer thành lập năm 1812 đúng không?	Diego Maradona nhỏ tuổi hơn Rutherford B. Hayes là [ANSWER]. Đội bóng của Nathan Dyer thành lập năm 1812 là [ANSWER].
Nào (Which)	Đội bóng nào Danny Drinkwater từng thi đấu còn được biết đến là The Foxes? Câu lạc bộ nào James Milner từng chơi có trụ sở tại Swindon, Wiltshire, Anh?	Đội bóng Danny Drinkwater từng thi đấu còn được biết đến là The Foxes là [ANSWER]. Câu lạc bộ James Milner từng chơi có trụ sở tại Swindon, Wiltshire, Anh là [ANSWER].
Là gì (What)	Hồ Hoàn Kiếm trước đây còn được gọi là gì? Cristiano Ronaldo còn có biệt danh là gì?	Hồ Hoàn Kiếm trước đây còn được gọi là [ANSWER]. Cristiano Ronaldo còn có biệt danh là [ANSWER].
Thứ mấy (What ordinal number) Thứ bao nhiêu	Quốc gia của câu lạc bộ bóng đá Queens Park thống nhất vào thế kỷ thứ mấy? Tổng thống chọn sẵn William Howard Taft để kế nhiệm chức vụ là Tổng thống thứ bao nhiêu của Hoa Kỳ?	Quốc gia của câu lạc bộ bóng đá Queens Park thống nhất vào thế kỷ thứ [ANSWER]. Tổng thống chọn sẵn William Howard Taft để kế nhiệm chức vụ là Tổng thống Hoa Kỳ thứ [ANSWER].
Ai (Who) Bởi ai (By whom)	Ai là người lập nên chính quyền Tào Ngụy thời Tam Quốc? Ai là giám đốc của Panasonic? Album có ca khúc In My Life được sản xuất bởi ai?	Người lập nên chính quyền Tào Ngụy thời Tam Quốc là [ANSWER]. Giám đốc của Panasonic là [ANSWER]. Album có ca khúc In My Life được sản xuất bởi [ANSWER].
Bao nhiêu (How many)	Diện tích của Châu Phi là bao nhiêu? Ban nhạc có sản phẩm nổi tiếng nhất là đĩa đơn đầu năm 2003 có bao nhiêu thành viên?	Diện tích của Châu Phi là [ANSWER]. Số thành viên của ban nhạc có sản phẩm nổi tiếng nhất là đĩa đơn đầu năm 2003 là [ANSWER].
Bao lâu một lần (How often)	Tạp chí có danh sách "100 nghệ sĩ guitar vĩ đại nhất"xuất bản bao lâu một lần?	Thời gian tạp chí có danh sách "100 nghệ sĩ guitar vĩ đại nhất"xuất bản một lần là [ANSWER].
Bao xa (How far)	Nhật Bản cách Việt Nam bao xa?	Khoảng cách giữa Nhật Bản cách Việt Nam là [ANSWER].
Bao lâu (How long)	Mùa hè kéo dài bao lâu?	Khoảng thời gian mùa hè kéo dài là [ANSWER].
Khi nào (When)	Thơ Đường luật xuất hiện từ khi nào? Bhutan thiết lập quan hệ với Đế quốc anh khi nào?	Thơ Đường luật xuất hiện từ [ANSWER]. Bhutan thiết lập quan hệ với Đế quốc anh từ [ANSWER].
Ở đâu (Where)	Mạch chỉnh lưu có thể sử dụng ở đâu? Quần đảo Cát Bà nằm tại đâu?	Mạch chỉnh lưu có thể sử dụng ở [ANSWER]. Quần đảo Cát Bà nằm tại [ANSWER].
Vì sao (Why) Tại sao	Vì sao nhà Lý chấm dứt? Tại sao trái đất quay tròn?	Nhà Lý chấm dứt vì [ANSWER]. Trái đất quay tròn vì [ANSWER].

Table 4.2: Question-Answer Pairs in Interrogative and Declarative Forms in Vietnamese

Chapter 5

ANSPRE-VLM: Adaptation to Vision-Language Models

5.0.1 Problem Formulation

Consider the question *"Which kind of technology had the highest priority in 2016?"* paired with a chart image titled *"Most popular technology by year"*. In this example, a vision-language model (VLM) must integrate textual and visual cues to infer that the correct answer is *"Internet of Things"*. The goal is to generate this answer phrase along with an associated confidence score that reflects the model's certainty. Achieving this requires the model to jointly reason about the semantics of the question and the information encoded within the visual content.

Formally, given a question Q, an image I, and a pre-trained vision-language model $\mathcal{M}$, the objective is to generate an answer sequence $Y = [y_1, y_2, \ldots, y_T]$ with an estimated confidence measure. At each decoding step t, the model predicts the next token y_t conditioned on the question, the visual input, and previously generated tokens as follows:

$$P_{\mathcal{M}}(y_t \mid Q, I, y_{<t}), \tag{5.1}$$

where $y_{<t}$ represents the sequence of tokens generated before step t. The joint probability of the entire answer phrase can then be expressed as:

$$P_{\mathcal{M}}(Y \mid Q, I) = \prod_{t=1}^{T} P_{\mathcal{M}}(y_t \mid Q, I, y_{<t}). \tag{5.2}$$

The input image I is first encoded into a set of visual embeddings, which are integrated into the transformer's cross-attention mechanism to enable

multimodal reasoning. The cumulative log-likelihood of the generated sequence is often used as a surrogate for the model's confidence in its output.

Despite their success, existing VLMs exhibit several limitations when applied to open-ended question answering tasks (Figure 5.1, Top). These include: (i) difficulty in producing precise and concise answer phrases, (ii) a weak correlation between token-level probabilities and actual model confidence, and (iii) a lack of structured methods for aggregating answers from multiple samples or decoding paths. To mitigate these issues, we extend the ANSPRE framework to the multimodal domain and propose ANSPRE-VLM, a method that improves visual-textual grounding and introduces more reliable confidence calibration for VLM outputs.

5.0.2 Answer Generation in Anspre-VLM

The ANSPRE generation framework can be naturally extended to vision-language models. The general process follows the same principle as the text-only ANSPRE (Figure 1.2, Bottom), with the main difference being the integration of visual features during the answer generation phase.

Given the answer prefix E generated similarly to Section 3.2, the VLM is prompted to generate the subsequent tokens conditioned on both textual and visual inputs. Specifically, the probability distribution for token generation at time step t is given by:

$$P_{\mathcal{M}}(y_t \mid Q \oplus E, I, y_{<t}), \tag{5.3}$$

where $\oplus$ denotes the concatenation operator between the question Q and the answer prefix E. The generation process continues until an end-of-phrase condition is reached, typically signaled by punctuation or a designated stop token. We employ beam search with width B to generate the top N most likely candidate answers, which are subsequently aggregated following the same strategy as the base ANSPRE framework.

To integrate both the model's parametric knowledge and its visually grounded reasoning, we define two scoring components analogous to those used in the text-only setting:

$$\mathcal{S}_{\text{Phrase}} = P_{\mathcal{M}}(Y \mid Q \oplus E, I), \tag{5.4}$$

$$\mathcal{S}_{\text{Sentence}} = P_{\mathcal{M}}(E \oplus Y). \tag{5.5}$$

The final score is computed as a weighted combination of these two components:

$$\mathcal{S}_{\text{ANSPRE-VLM}} = w_{\text{Phrase}} \cdot \mathcal{S}_{\text{Phrase}} + w_{\text{Sentence}} \cdot \mathcal{S}_{\text{Sentence}}, \tag{5.6}$$

where w_{Phrase} and w_{Sentence} are predefined weights satisfying $w_{\text{Phrase}} + w_{\text{Sentence}} = 1$. This formulation ensures a balanced contribution between visual grounding and textual fluency, leading to more accurate and confidence-calibrated multimodal responses.

5.0.3 Additional Factors for Multimodal Adaptation

In vision-language models, the textual prompt is often augmented with special tokens that specify the position of the image input relative to the text. The text and image embeddings are then fused within a multimodal encoder-decoder architecture to enable cross-modal attention during decoding. For instruction-tuned VLMs such as LLaVA [48] and BLIP-2 [46], a chat-style template is typically employed during inference. This template enforces a specific conversational structure, marking user queries and assistant responses, and inserting placeholder tokens for image inputs.

When adapting ANSPRE to such VLMs, care must be taken to ensure that the answer prefix is appended in the correct position—specifically, after the tokens that denote the start of the assistant's response. This placement guarantees that the model continues generation directly from the prefix, maintaining grammatical coherence and semantic relevance.

In the base ANSPRE framework, answer aggregation operates along two axes: (i) across retrieved documents and (ii) across multiple response samples per document. However, in the multimodal setting, retrieval-based variation does not exist since the input is limited to a single image-question pair. Therefore, ANSPRE-VLM performs aggregation only across multiple generated responses, applying the same probabilistic averaging and consistency filtering techniques as in the base method. This enables robust answer consolidation and improved prediction reliability in multimodal reasoning tasks.

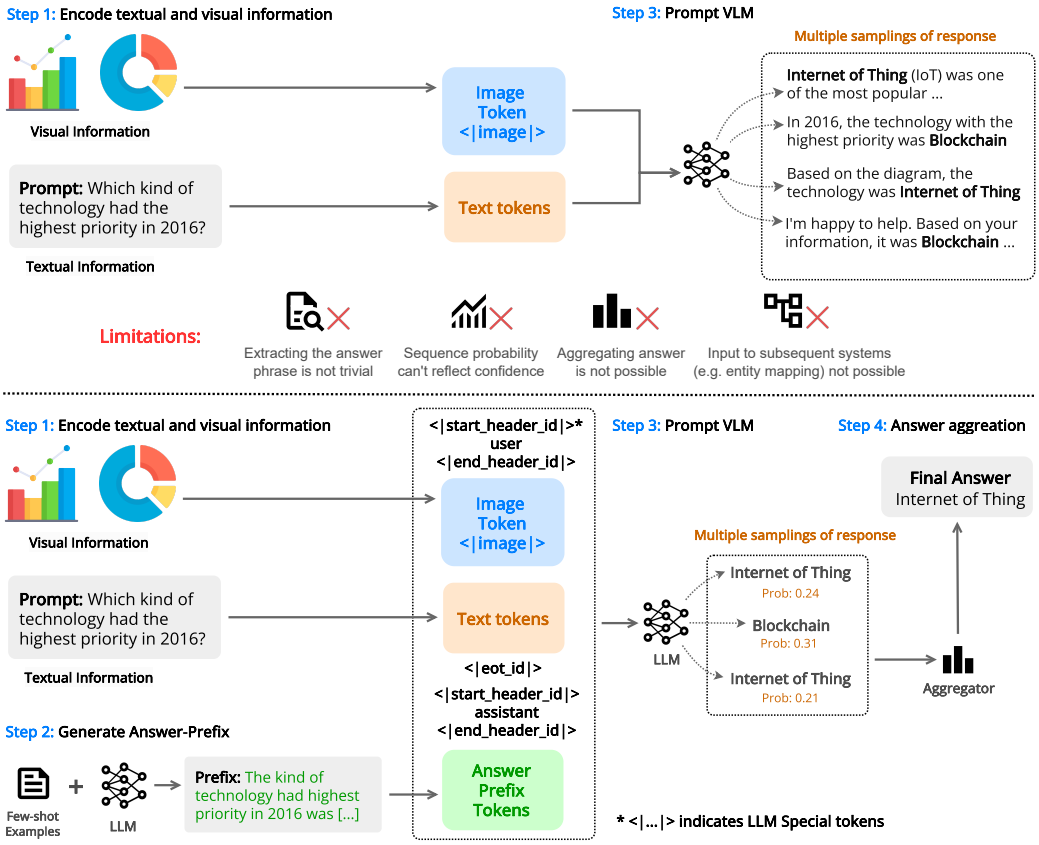


Figure 5.1: Overview of the ANSPRE-VLM framework compared to standard vision-language generation. The proposed method enhances multimodal grounding and confidence calibration.

Chapter 6

Experiments and Results on Anspre Generation Techniques

6.1 Tasks and Datasets

6.1.1 Textual Question Answering

To assess the effectiveness of our proposed approach, we conduct extensive evaluations on three widely adopted English open-domain question-answering (ODQA) benchmarks.

PopQA [51] is designed to test a model's ability to recall factual knowledge. Following prior work, we focus on the long-tail subset, which contains 1,399 questions centered on rare entities—specifically, those with fewer than 100 monthly page views on Wikipedia. This subset is particularly challenging, as it emphasizes knowledge that is underrepresented in large-scale training corpora.

TriviaQA [32] evaluates factual reasoning across diverse domains, such as history, science, and literature. We employ the open-domain (unfiltered) version of TriviaQA, which does not provide access to the original test set. Therefore, consistent with established practice, we use the validation and test splits adopted in prior studies, totaling 7,313 test examples.

Natural Questions (NQ) [35] consists of real-world questions posed by Google search users, offering a realistic evaluation setting for fact-based QA. We utilize the Open-NQ subset introduced by Lee2019LatentAnswering, which comprises 3,610 questions in its validation set.

6.1.2 Multilingual Question Answering

To evaluate the generalization of our approach across languages, we further perform experiments on multiple multilingual QA datasets.

VIMQA [37] is a Vietnamese multi-hop question-answering benchmark containing over 10,000 manually crafted questions and answers derived from Wikipedia articles. We report results on its test set.

ViQuAD [61] is another Vietnamese reading comprehension dataset with 23,000 human-annotated QA pairs based on 5,109 Wikipedia passages. It provides a standard benchmark for evaluating Vietnamese QA performance.

JAQKET [70] is an open-domain Japanese QA dataset intended to promote research on question answering and machine reading comprehension in Japanese. Answers are derived from Wikipedia article titles. Our evaluation is performed on version 2.0 of the dataset's validation set, which includes more than one thousand samples.

JaQuAD [66] is a large-scale, human-annotated Japanese MRC dataset inspired by SQuAD, comprising 39,696 QA pairs derived from Japanese Wikipedia. The dataset is commonly used to evaluate contextual comprehension in Japanese models. We report results using its validation split, which contains approximately four thousand instances.

6.1.3 Visual Question Answering

We further extend our evaluation to the multimodal domain to investigate whether our approach generalizes beyond text-based tasks.

InfographicVQA [52] features questions grounded in real-world infographics, requiring the model to reason over textual, visual, and graphical content simultaneously. Many queries also demand basic arithmetic or reasoning over data visualizations.

DocVQA [53] consists of 50,000 questions spanning over 12,000 document images. The task requires understanding complex document layouts, recognizing textual and visual elements, and synthesizing information from structured data sources.

All evaluations are conducted in a **zero-shot** setting, where only task instructions are provided to the model without any few-shot examples. Model outputs are evaluated using three metrics: (i) **Exact Match (EM)**, which measures the proportion of predictions that exactly match the reference answers; (ii) **F1-score**, which computes the token-level overlap between the prediction and gold answer; and (iii) **Match Accuracy**, which checks whether the gold answer appears anywhere in the model's response.

6.2 Baselines

We compare our proposed method against a diverse set of publicly available large language models (LLMs), covering various architectures, training paradigms, and adaptation strategies.

Pre-trained Foundation LLMs. We first evaluate standard foundation models without additional fine-tuning, including Llama2$_{7B,\ 13B}$ [73], Llama3$_{8B}$ [2], Llama3.1$_{8B}$, Llama3.2$_{3B}$, Mistral$_{7B}$ [29], Gemma$_{7B}$ [71], and OPT$_{6.7B}$ [86].

LLMs without retrieval. In this configuration, we assess models such as Llama2$_{7B,\ 13B}$ [73] operating solely with their internal parametric memory, without incorporating external retrieval components.

Multilingual LLMs. We evaluate multilingual foundation models optimized for both English and Chinese, which have demonstrated strong cross-lingual performance. This group includes Qwen$_{7B,\ 14B}$ [5], Qwen2$_{7B}$ [83], Qwen2.5$_{7B}$ [72], and Baichuan2$_{7B,\ 13B}$ [82].

Instruction-tuned and Reinforcement-optimized LLMs. We also evaluate instruction-following models fine-tuned on human feedback or proprietary data, including Llama2-Chat$_{7B,\ 13B}$ [74] and Vicuna$_{13B}$ [8].

Self-Reflective Retrieval-Augmented Models. We further compare our approach with Self-RAG$_{7B,\ 13B}$ [4], a retrieval-augmented model that integrates self-reflection during inference. This comparison highlights the contribution of our proposed Self-Anspre framework relative to a strong self-reflective RAG baseline.

Multimodal LLMs. For visual QA tasks, we include multimodal LLMs capable of processing both image and text inputs. These include InternVL 2.0$_{8B}$ [7], Llama 3.2-Vision$_{11B}$, Qwen2-VL$_{2B,\ 7B}$ [77], and LLaVA-v1.6-Vicuna$_{7B}$ [47].

Finally, for each model, we compare performance with and without the integration of our proposed Anspre method to quantify its effectiveness in improving model reasoning and factual accuracy.

6.3 Experimental Settings

For all experiments within the Anspre framework, we generate the answer prefix using greedy decoding. This choice aligns with standard practices in recent studies on retrieval-augmented generation [4]. In our setup, English Wikipedia serves as the primary knowledge source: the 2018 snapshot is used for general evaluation, while the December 2020 version is specifically employed for the PopQA dataset to ensure consistency with prior work.

During the retrieval stage, we leverage the Contriever-MS MARCO retriever [28] to identify the top $k = 5$ most relevant documents for each query. Each retrieved document is then used to generate $N = 10$ candidate answer samples. This multi-sample approach allows for robust answer aggregation and improves the likelihood of producing accurate responses. For the final answer scoring in ANSPRE, we use equal weighting between the phrase-level and sentence-level components, setting $w_{\text{Phrase}} = w_{\text{Sentence}} = 0.5$. The autoregressive generation is limited to a maximum of 100 new tokens per answer to maintain efficiency.

To ensure fair and consistent evaluation, we apply a standard normalization function η to the generated answers. This function performs several preprocessing steps: it removes articles, strips punctuation, normalizes whitespace, and converts all text to lowercase. These steps reduce superficial mismatches between predicted and reference answers, allowing evaluation metrics to focus on meaningful content alignment.

For baseline models that are instruction-tuned or reinforcement-optimized, we employ multiple prompt variants to steer the LLM towards producing concise answer phrases. The best-performing prompt is reported for each model to provide a fair comparison.

In the case of reflective, retrieval-augmented models such as SELF-RAG and SELF-ANSPRE, we adopt the same reflection token weights as defined by [4]. Specifically, the weight for ISREL is set to 1.0, for ISSUP to 1.0, and for ISUSE to 0.5. In SELF-ANSPRE, the default weight for combining multiple answer scores is set to $w^A = 0.5$, balancing contributions from the retrieved evidence and the model's generative predictions. These settings provide a controlled and reproducible experimental environment while ensuring that our proposed method is fairly evaluated against competitive baselines.

6.4 Results and Analysis

6.4.1 Main Results

Model	TriviaQA Match	TriviaQA EM	TriviaQA F1	PopQA Match	PopQA EM	PopQA F1	NaturalQuestion Match	NaturalQuestion EM	NaturalQuestion F1
Fine-tuned & Commercial model baselines (for references)									
Graph Retriever [58]	-	55.8	-	-	-	-	-	34.7	-
Hard EM [57]	-	50.9	-	-	-	-	-	28.8	-
ORQA [40]	-	45.1	-	-	-	-	-	31.3	-
DPR [33]	-	57.9	-	-	-	-	-	41.5	-
RAG [42]	-	56.1	-	-	-	-	-	44.5	-
ChatGPT	-	74.3	-	-	29.3	-	-	-	-
ChatGPT+Retrieve	-	65.7	-	-	50.8	-	-	-	-
Pre-trained Foundation LLM with retrieval									
Llama-2_{7B}	48.28	9.83	24.83	38.74	6.72	18.65	23.85	0.94	7.42
Llama-2_{7B}-ANSPRE	52.85$_{\uparrow 4.57}$	47.61$_{\uparrow 37.78}$	55.55$_{\uparrow 30.72}$	42.17$_{\uparrow 3.43}$	43.10$_{\uparrow 36.38}$	47.01$_{\uparrow 28.36}$	24.90$_{\uparrow 1.05}$	23.71$_{\uparrow 22.77}$	30.73$_{\uparrow 23.31}$
Llama-2_{13B}	52.48	17.38	32.35	42.17	5.15	18.79	26.15	4.96	14.32
Llama-2_{13B}-ANSPRE	58.51$_{\uparrow 6.03}$	53.64$_{\uparrow 36.26}$	61.78$_{\uparrow 29.43}$	46.03$_{\uparrow 3.86}$	44.39$_{\uparrow 39.24}$	47.98$_{\uparrow 29.19}$	30.36$_{\uparrow 4.21}$	29.34$_{\uparrow 24.38}$	37.52$_{\uparrow 23.2}$
Llama-3_{8B}	60.88	51.68	60.68	48.82	31.59	37.96	33.38	21.14	27.67
Llama-3_{8B}-ANSPRE	63.12$_{\uparrow 2.24}$	57.75$_{\uparrow 6.07}$	67.46$_{\uparrow 6.78}$	48.96$_{\uparrow 0.14}$	48.03$_{\uparrow 16.44}$	51.83$_{\uparrow 13.87}$	30.50$_{\downarrow -2.88}$	29.36$_{\uparrow 8.22}$	39.58$_{\uparrow 11.91}$
Llama-3.1_{8B}	57.71	18.72	32.36	49.96	7.01	20.37	33.35	5.98	14.10
Llama-3.1_{8B}-ANSPRE	64.16$_{\uparrow 6.45}$	57.90$_{\uparrow 39.18}$	67.95$_{\uparrow 35.59}$	49.61$_{\downarrow -0.35}$	48.68$_{\uparrow 41.67}$	52.21$_{\uparrow 31.84}$	31.25$_{\downarrow -2.1}$	28.98$_{\uparrow 23.0}$	39.82$_{\uparrow 25.72}$
Llama-3.2_{3B}	54.85	0.96	8.12	47.53	0.29	6.04	31.14	0.00	3.17
Llama-3.2_{3B}-ANSPRE	55.76$_{\uparrow 0.91}$	50.08$_{\uparrow 49.12}$	59.31$_{\uparrow 51.19}$	47.53$_{\uparrow 0.0}$	46.96$_{\uparrow 46.67}$	49.77$_{\uparrow 43.73}$	27.23$_{\downarrow -3.91}$	25.43$_{\uparrow 25.43}$	34.87$_{\uparrow 31.7}$
Mistral$_{7B}$	54.29	12.59	26.25	44.53	4.50	17.05	33.52	3.32	10.31
Mistral$_{7B}$-ANSPRE	61.11$_{\uparrow 6.82}$	55.13$_{\uparrow 42.54}$	64.88$_{\uparrow 38.63}$	47.53$_{\uparrow 3.0}$	45.82$_{\uparrow 41.32}$	49.97$_{\uparrow 32.92}$	32.52$_{\downarrow 1.0}$	29.89$_{\uparrow 26.57}$	40.23$_{\uparrow 29.92}$
Gemma$_{7B}$	58.91	31.40	44.23	42.74	21.30	31.45	38.67	9.72	20.06
Gemma$_{7B}$-ANSPRE	61.89$_{\uparrow 2.98}$	55.18$_{\uparrow 23.78}$	65.35$_{\uparrow 21.12}$	46.53$_{\uparrow 3.79}$	44.89$_{\uparrow 23.59}$	47.88$_{\uparrow 16.43}$	31.58$_{\downarrow 7.09}$	28.48$_{\uparrow 18.76}$	39.46$_{\uparrow 19.4}$
OPT$_{6.7B}$	46.37	3.08	11.22	40.89	1.36	9.12	22.13	0.14	3.28
OPT$_{6.7B}$-ANSPRE	41.52$_{\downarrow 4.85}$	35.16$_{\uparrow 32.08}$	43.19$_{\uparrow 31.97}$	40.10$_{\downarrow 0.79}$	37.60$_{\uparrow 36.24}$	41.21$_{\uparrow 32.09}$	19.83$_{\downarrow 2.30}$	18.64$_{\uparrow 18.5}$	26.23$_{\uparrow 22.95}$
Pre-trained Foundation LLM without retrieval (only parametric memories)									
Llama2_{13B}	46.47	33.38	42.62	15.23	9.86	13.88	16.29	8.17	15.48
Llama2_{13B}ANSPRE	61.23$_{\uparrow 14.76}$	55.08$_{\uparrow 21.7}$	64.66$_{\uparrow 22.04}$	24.16$_{\uparrow 8.93}$	23.30$_{\uparrow 13.44}$	26.67$_{\uparrow 12.79}$	30.19$_{\uparrow 13.9}$	28.03$_{\uparrow 19.86}$	38.12$_{\uparrow 22.64}$
Llama2_{7B}	28.39	10.64	20.19	11.58	4.65	9.04	12.13	2.96	8.68
Llama2_{7B}-ANSPRE	55.59$_{\uparrow 27.2}$	49.31$_{\uparrow 38.67}$	58.34$_{\uparrow 38.15}$	24.37$_{\uparrow 12.79}$	25.66$_{\uparrow 21.01}$	27.13$_{\uparrow 18.09}$	25.24$_{\uparrow 13.11}$	23.68$_{\uparrow 20.72}$	32.41$_{\uparrow 23.73}$
Pre-trained Foundation LLM on multilingual data (focus on Chinese and English) with retrieval									
Baichuan2_{7B}	44.20	13.58	26.46	40.74	4.65	15.34	20.78	2.11	7.78
Baichuan2_{7B}-ANSPRE	54.98$_{\uparrow 10.78}$	48.48$_{\uparrow 34.9}$	57.88$_{\uparrow 31.42}$	41.24$_{\uparrow 0.5}$	40.81$_{\uparrow 36.16}$	43.92$_{\uparrow 28.58}$	24.43$_{\uparrow 3.65}$	21.33$_{\uparrow 19.22}$	31.05$_{\uparrow 23.27}$
Baichuan2_{13B}	48.99	8.10	21.54	39.10	4.29	14.62	27.12	0.83	6.49
Baichuan2_{13B}-ANSPRE	53.89$_{\uparrow 4.9}$	48.04$_{\uparrow 39.94}$	57.07$_{\uparrow 35.53}$	40.53$_{\uparrow 1.43}$	39.53$_{\uparrow 35.24}$	42.85$_{\uparrow 28.23}$	27.34$_{\uparrow 0.22}$	25.40$_{\uparrow 24.57}$	34.90$_{\uparrow 28.41}$
Qwen$_{7B}$	54.49	0.44	5.27	48.96	0.14	2.10	33.93	0.19	3.68
Qwen$_{7B}$-ANSPRE	57.09$_{\uparrow 2.6}$	48.00$_{\uparrow 47.56}$	59.64$_{\uparrow 54.37}$	48.61$_{\downarrow 0.35}$	46.18$_{\uparrow 46.04}$	50.94$_{\uparrow 48.84}$	30.72$_{\downarrow 3.21}$	24.43$_{\uparrow 24.24}$	36.17$_{\uparrow 32.49}$
Qwen$_{14B}$	57.04	1.38	5.76	47.18	0.36	2.86	36.62	0.50	3.55
Qwen$_{14B}$-ANSPRE	61.00$_{\uparrow 3.96}$	52.00$_{\uparrow 50.62}$	63.48$_{\uparrow 57.72}$	47.75$_{\uparrow 0.57}$	45.89$_{\uparrow 45.53}$	49.40$_{\uparrow 46.54}$	29.53$_{\downarrow 7.09}$	22.91$_{\uparrow 22.41}$	35.07$_{\uparrow 31.52}$
Qwen2$_{7B}$	66.24	1.97	24.26	56.33	0.00	19.33	47.23	1.05	16.46
Qwen2$_{7B}$-ANSPRE	60.30$_{\downarrow -5.94}$	50.49$_{\uparrow 48.52}$	62.41$_{\uparrow 38.15}$	49.54$_{\downarrow -6.79}$	46.32$_{\uparrow 46.32}$	50.38$_{\uparrow 31.05}$	33.19$_{\downarrow -14.04}$	27.20$_{\uparrow 26.15}$	38.42$_{\uparrow 21.96}$
Qwen2.5_{7B}	66.98	3.28	24.38	57.97	0.00	20.24	47.51	0.94	15.00
Qwen2.5_{7B}-ANSPRE	59.13$_{\downarrow -7.85}$	48.61$_{\uparrow 45.33}$	60.81$_{\uparrow 36.43}$	49.11$_{\downarrow -8.86}$	44.82$_{\uparrow 44.82}$	49.41$_{\uparrow 29.17}$	30.83$_{\downarrow -16.68}$	24.02$_{\uparrow 23.08}$	35.61$_{\uparrow 20.61}$
Instruction-tuned & Reinforced LLM									
Llama2-C_{7B}	60.41	33.20	48.32	45.18	23.73	36.33	38.14	9.81	21.19
Llama2-C_{7B}-ANSPRE	57.73$_{\downarrow 2.68}$	40.97$_{\uparrow 7.77}$	55.54$_{\uparrow 7.22}$	50.25$_{\uparrow 5.07}$	40.53$_{\uparrow 16.8}$	48.05$_{\uparrow 11.72}$	31.08$_{\downarrow 7.06}$	19.58$_{\uparrow 9.77}$	31.38$_{\uparrow 10.19}$
Llama2-C_{13B}	66.84	5.67	33.03	54.32	0.36	17.40	41.69	1.30	15.92
Llama2-C_{13B}-ANSPRE	62.03$_{\downarrow 4.81}$	47.00$_{\uparrow 41.33}$	61.08$_{\uparrow 28.05}$	50.46$_{\downarrow 3.86}$	42.82$_{\uparrow 42.46}$	49.11$_{\uparrow 31.71}$	33.13$_{\downarrow 8.56}$	22.99$_{\uparrow 21.69}$	35.43$_{\uparrow 19.51}$
Vicuna$_{13B}$	66.53	30.86	48.98	56.25	3.65	23.60	43.21	5.29	20.33
Vicuna$_{13B}$-ANSPRE	62.35$_{\downarrow 4.18}$	44.33$_{\uparrow 13.47}$	59.98$_{\uparrow 11.0}$	47.46$_{\downarrow 8.79}$	39.67$_{\uparrow 36.02}$	46.04$_{\uparrow 22.44}$	34.10$_{\downarrow 9.11}$	18.01$_{\uparrow 12.72}$	32.46$_{\uparrow 12.13}$
Llama-3.1-Instr$_{8B}$	63.86	19.40	36.71	52.61	8.22	21.66	39.70	21.80	37.38
Llama-3.1-Instr$_{8B}$-ANSPRE	46.11$_{\downarrow 17.75}$	35.29$_{\uparrow 15.89}$	46.08$_{\uparrow 9.37}$	45.68$_{\downarrow 6.93}$	40.67$_{\uparrow 32.45}$	46.00$_{\uparrow 24.34}$	21.97$_{\downarrow 17.73}$	16.70$_{\downarrow 5.1}$	26.22$_{\downarrow 11.16}$
Llama-3.2-Instr$_{3B}$	49.17	19.66	31.79	49.89	1.36	12.09	37.78	3.82	12.75
Llama-3.2-Instr$_{3B}$-ANSPRE	37.78$_{\downarrow 11.39}$	23.04$_{\uparrow 3.38}$	36.18$_{\uparrow 4.39}$	24.30$_{\downarrow 25.59}$	13.51$_{\uparrow 12.15}$	21.68$_{\uparrow 9.59}$	14.40$_{\downarrow 23.38}$	7.34$_{\uparrow 3.52}$	16.12$_{\uparrow 3.37}$
Llama-3.1-Instr$_{70B}$	66.95	38.51	59.17	49.32	25.02	39.89	47.06	17.51	36.80
Llama-3.1-Instr$_{70B}$-ANSPRE	57.21$_{\downarrow 9.74}$	42.80$_{\uparrow 4.29}$	56.05$_{\downarrow 3.12}$	51.25$_{\uparrow 1.93}$	45.25$_{\uparrow 20.23}$	51.24$_{\uparrow 11.35}$	27.89$_{\downarrow 19.17}$	18.37$_{\uparrow 0.86}$	31.37$_{\downarrow 5.43}$
Self-Reflective Retrieval-Augmented-Generation									
SELF-RAG $_{7B}$	66.18	20.91	38.70	54.97	1.14	19.67	36.15	36.73	44.89
SELF-ANSPRE $_{7B}$	66.21$_{\uparrow 0.03}$	40.15$_{\uparrow 19.24}$	58.42$_{\uparrow 19.72}$	54.11$_{\downarrow 0.86}$	42.17$_{\uparrow 41.03}$	51.91$_{\uparrow 32.24}$	38.14$_{\uparrow 1.99}$	31.25$_{\downarrow 5.47}$	39.73$_{\downarrow 5.16}$
SELF-RAG $_{13B}$	67.59	19.14	38.37	55.83	0.79	20.10	38.73	40.75	48.72
SELF-ANSPRE $_{13B}$	66.99$_{\downarrow 0.6}$	46.78$_{\uparrow 27.64}$	62.98$_{\uparrow 24.61}$	52.61$_{\downarrow 3.22}$	38.74$_{\uparrow 37.95}$	49.41$_{\uparrow 29.31}$	40.30$_{\uparrow 1.57}$	35.68$_{\downarrow 5.07}$	44.79$_{\downarrow 3.93}$

Table 6.1: Performance on three ODQA tasks. Each pair of rows contrasts the performance of an LLM with and without ANSPRE.

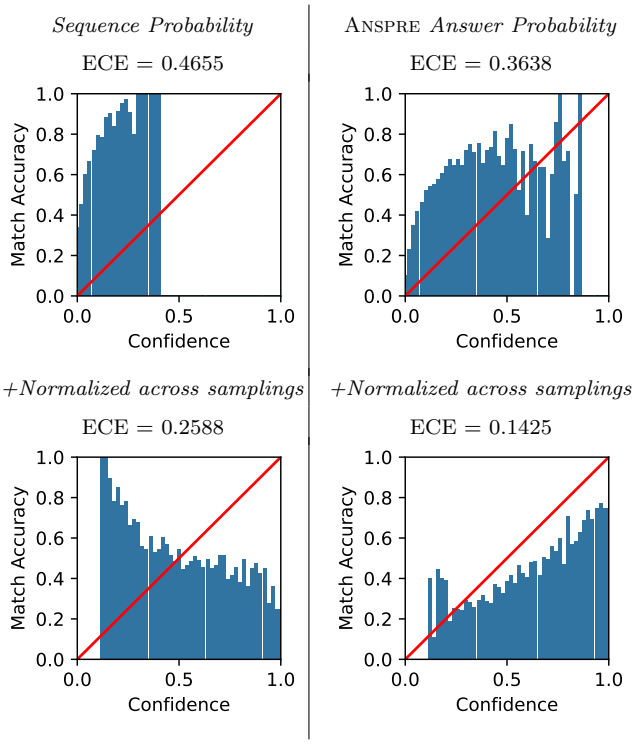
Performance Comparison with Baselines.

Table 6.1 summarizes the evaluation results of various baseline LLMs, comparing their original performance with the enhancements provided by ANSPRE. In the table, blue numbers indicate improvements achieved by ANSPRE, whereas red numbers denote a reduction in performance relative to the corresponding baseline model. To ensure a fair comparison, we apply a consistent aggregation strategy across all ANSPRE evaluations, specifically aggregating multiple sampled responses within the top-ranked retrieved document.

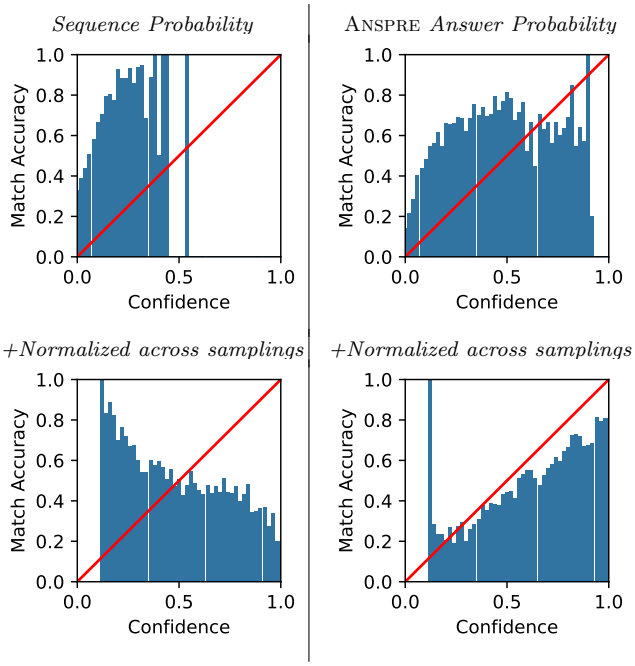
Overall, ANSPRE demonstrates consistent improvements across most evaluation metrics, particularly Exact Match (EM) and F1-score, reflecting its ability to generate more precise and well-aligned answers. However, in certain cases, a minor decrease in Match-accuracy is observed. This phenomenon can be attributed to the baseline LLMs' tendency to produce longer outputs containing additional context, which increases the likelihood of the correct answer appearing somewhere within the generated text.

For instruction-tuned and reinforced LLMs, we explicitly constrain the models to produce only the answer phrase. Without such instructions, baseline outputs are generally longer and substantially underperform in EM and F1 metrics. Even under constrained conditions, baseline models consistently lag behind ANSPRE, highlighting the effectiveness of our framework in enhancing answer precision and reducing irrelevant content. Notably, we observe a pronounced performance gap between Llama2-C_{7B} and Llama2-C_{13B}. Error analysis indicates that, despite the prompt constraints, Llama2-C_{13B} frequently generates conversational fillers or extraneous text, which diminishes EM and F1 scores. The ANSPRE framework alleviates this issue by guiding the model to focus on the core answer phrase, thereby improving output consistency.

In the Self-Reflective retrieval-augmented generation (RAG) group, SELF-ANSPRE outperforms SELF-RAG across most metrics. Specifically, EM and F1 improvements are notable on the TriviaQA and PopQA datasets, while Match-accuracy remains largely comparable. For the NaturalQuestion dataset, SELF-ANSPRE achieves higher Match-accuracy relative to SELF-RAG, even though a marginal decrease in EM and F1 is observed. These results suggest that ANSPRE effectively balances answer precision with broader coverage, mitigating issues caused by extraneous information in the model outputs [4].



(a) Llama2 reliability diagram



(b) Mistral_{7B} reliability diagram.

Figure 6.1: Reliability diagram of confidence score by sequence probability (left) and by ANSPRE (right). Red diagonal represents perfect calibration.

Model	ANSPRE	ECE (default)	ECE (normalized)
$\text{Llama2}_{\text{7B}}$		0.4655	0.2588
	✓	0.3638	**0.1426**
$\text{Mistral}_{\text{7B}}$		0.4572	0.2669
	✓	0.3519	**0.1045**
Gemma_{7B}		0.4518	0.3907
	✓	0.2489	**0.0840**
$\text{Baichuan2}_{\text{7B}}$		0.4150	0.2460
	✓	**0.0884**	0.1055
Qwen_{7B}		0.5489	0.3399
	✓	0.2037	**0.1463**

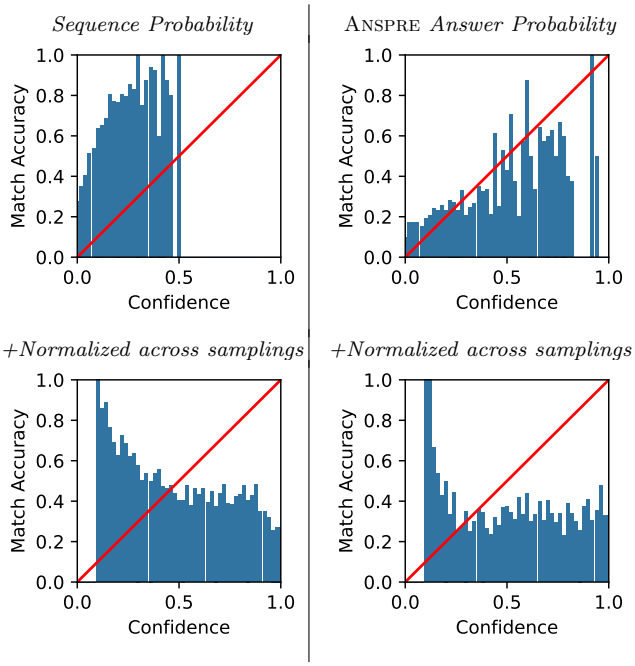
Table 6.2: Comparison of confidence score reliability, measured in Expected Calibration Error (ECE), lower is better. By *default*, the confidence score is obtained by cumulative probability (for baseline) and S_{Phrase} (for ANSPRE). *normalized* indicate normalizing across samplings in generation

Confidence calibration comparison with baseline models.

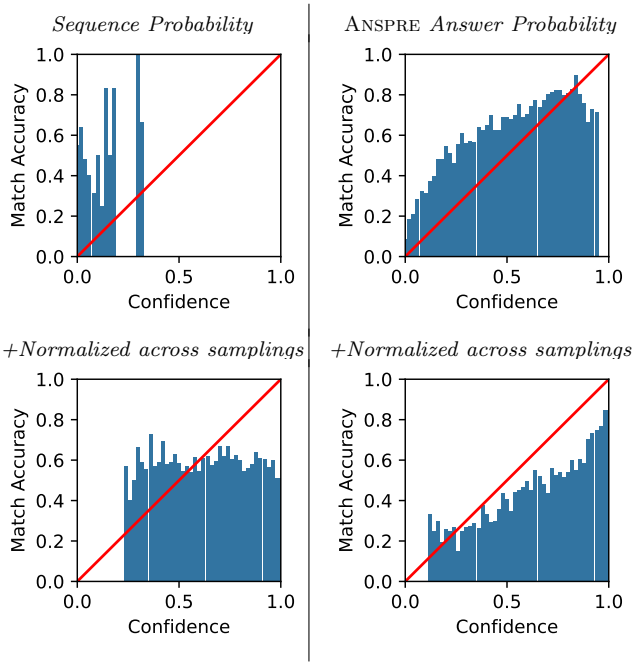
Figure 6.1 and Figure 6.2 presents a reliability diagram that compares the confidence calibration of large language models (LLMs) against that of ANSPRE. In this comparison, the x-axis represents the predicted confidence (i.e., the model's internal probability estimate), while the y-axis denotes the empirical accuracy of the predictions. Ideally, a perfectly calibrated model would align along the diagonal line, where predicted confidence equals actual accuracy.

Confidence scores derived from ANSPRE exhibit significantly improved alignment with the true correctness probabilities, indicating a more faithful estimation of uncertainty. This improvement stems from the aggregation and normalization processes incorporated in ANSPRE, which integrate evidence from multiple generated samples and reduce overconfidence—a common issue in autoregressive LLMs. By averaging likelihoods across sampled responses, ANSPRE effectively mitigates variance in individual predictions, resulting in smoother and more consistent calibration behavior.

Table 6.2 reports the Expected Calibration Error (ECE) for both the baseline LLMs and ANSPRE. Across all evaluated models, ANSPRE consistently achieves lower ECE scores, reflecting a closer correspondence between predicted probabilities and empirical correctness. Furthermore, normalizing



(a) Baichuan2$_{7B}$ reliability diagram



(b) Qwen$_{7B}$ reliability diagram

Figure 6.2: Reliability diagram of confidence score by sequence probability (left) and by ANSPRE (right). Red diagonal represents perfect calibration.

confidence estimates across multiple samples yields the best calibration results, demonstrating that uncertainty aggregation enhances reliability. These findings highlight the robustness of ANSPRE as a general-purpose framework not only for accuracy enhancement but also for improving interpretability and confidence estimation in generative LLMs.

Adaptation to Multilingual Settings.

Tables 6.3 and 6.4 present the results of applying ANSPRE to multilingual settings, focusing on Vietnamese and Japanese question-answering datasets [37, 61, 66, 70]. For these experiments, we adapt ANSPRE to account for linguistic and grammatical structures specific to each language, particularly in the formulation of declarative transformations and answer prefixes.

We evaluate recent multilingual-capable LLMs, including models from the Llama and Qwen families, using three key metrics: Match accuracy, Exact Match (EM), and F1-score. In both languages, the ANSPRE-enhanced models consistently outperform their baselines, as indicated by the blue values (performance gains) in the tables. Red values denote marginal decreases, which typically occur in cases where the model struggles with linguistic morphology or tokenization differences between languages.

Overall, these results demonstrate that ANSPRE maintains its effectiveness beyond English, providing measurable improvements in multilingual question answering. However, the degree of enhancement varies depending on the syntactic complexity and tokenization schemes of the target language. These findings suggest that further adaptation—such as incorporating language-specific templates or morphological normalization—could lead to even greater gains, particularly for typologically diverse languages with rich inflectional systems.

Adaptation to Visual Question Answering.

Table 6.5 summarizes the performance of baseline multimodal large language models (LLMs) when augmented with the proposed ANSPRE framework. Similar to the results observed in purely textual question answering tasks, integrating ANSPRE consistently enhances model performance in terms of Exact Match (EM) and F1 scores across all evaluated architectures and datasets. This improvement suggests that the probabilistic normalization and confidence-aware mechanisms of ANSPRE effectively extend to multimodal reasoning, where both visual and textual modalities jointly influence the model's predictions.

The performance gains highlight ANSPRE's ability to refine the alignment

	ViQUAD			VIMQA		
Model	Match	EM	F1	Match	EM	F1
Llama-3_{8B}	17.74	6.56	21.73	15.65	11.47	22.35
Llama-3_{8B}-ANSPRE	$9.68_{\downarrow 8.06}$	$8.10_{\uparrow 1.54}$	$23.06_{\uparrow 1.33}$	$20.74_{\uparrow 5.09}$	$19.64_{\uparrow 8.17}$	$33.19_{\uparrow 10.84}$
Llama-3.1_{8B}	13.94	2.26	17.36	17.05	3.99	15.03
Llama-3.1_{8B}-ANSPRE	$9.86_{\downarrow 4.08}$	$8.51_{\uparrow 6.25}$	$23.70_{\uparrow 6.34}$	$25.62_{\uparrow 8.57}$	$25.22_{\uparrow 21.23}$	$39.01_{\uparrow 23.98}$
Llama-3.2_{3B}	13.80	0.18	9.74	18.74	0.50	5.99
Llama-3.2_{3B}-ANSPRE	$8.01_{\downarrow 5.79}$	$6.83_{\uparrow 6.65}$	$20.09_{\uparrow 10.35}$	$20.64_{\uparrow 1.9}$	$19.44_{\uparrow 18.94}$	$30.82_{\uparrow 24.83}$
$Qwen2_{7B}$	27.15	0.50	25.17	21.73	0.50	16.24
$Qwen2_{7B}$-ANSPRE	$10.59_{\downarrow 16.56}$	$8.19_{\uparrow 7.69}$	$23.96_{\downarrow 1.21}$	$25.02_{\uparrow 3.29}$	$22.93_{\uparrow 22.43}$	$37.37_{\uparrow 21.13}$
$Qwen2.5_{7B}$	24.52	0.27	19.42	23.43	0.50	12.38
$Qwen2.5_{7B}$-ANSPRE	$11.49_{\downarrow 13.03}$	$9.00_{\uparrow 8.73}$	$24.96_{\uparrow 5.54}$	$25.52_{\uparrow 2.09}$	$23.23_{\uparrow 22.73}$	$37.53_{\uparrow 25.15}$

Table 6.3: Results on two Vietnamese ODQA tasks comparing baseline LLMs with and without ANSPRE. Each pair of rows contrasts the performance of an LLM with and without ANSPRE.

	JAQKET			JaQuAD		
Model	Match	EM	F1	Match	EM	F1
Llama-3_{8B}	65.21	22.77	31.11	65.02	15.87	20.91
Llama-3_{8B}-ANSPRE	$81.10_{\uparrow 15.89}$	$52.23_{\uparrow 29.46}$	$52.35_{\uparrow 21.24}$	$75.25_{\uparrow 10.23}$	$35.54_{\uparrow 19.67}$	$35.54_{\uparrow 14.63}$
Llama-3.1_{8B}	64.52	24.83	30.09	67.78	18.00	20.71
Llama-3.1_{8B}-ANSPRE	$80.93_{\uparrow 16.41}$	$44.33_{\uparrow 19.5}$	$44.44_{\uparrow 14.35}$	$75.68_{\uparrow 7.90}$	$36.89_{\uparrow 18.89}$	$36.89_{\uparrow 16.18}$
Llama-3.2_{3B}	57.22	1.55	5.31	53.47	0.86	2.72
Llama-3.2_{3B}-ANSPRE	$55.58_{\downarrow 1.64}$	$28.09_{\uparrow 26.54}$	$28.30_{\uparrow 22.99}$	$62.10_{\uparrow 8.63}$	$24.98_{\uparrow 24.12}$	$25.00_{\uparrow 22.28}$
$Qwen2_{7B}$	75.86	10.57	11.03	86.70	2.26	2.31
$Qwen2_{7B}$-ANSPRE	$77.49_{\uparrow 1.63}$	$37.11_{\uparrow 26.54}$	$37.29_{\uparrow 26.26}$	$72.66_{\downarrow 14.04}$	$33.46_{\uparrow 31.2}$	$33.49_{\uparrow 31.18}$
$Qwen2.5_{7B}$	75.09	6.62	7.79	88.12	2.08	2.24
$Qwen2.5_{7B}$-ANSPRE	$81.70_{\uparrow 6.61}$	$34.28_{\uparrow 27.66}$	$34.30_{\uparrow 26.51}$	$78.98_{\downarrow 9.14}$	$37.62_{\uparrow 35.54}$	$37.64_{\uparrow 35.4}$

Table 6.4: Results on two Japanese ODQA tasks comparing baseline LLMs with and without ANSPRE. Each pair of rows contrasts the performance of an LLM with and without ANSPRE.

between generated answers and ground-truth responses by reducing uncertainty in multimodal fusion. Specifically, by leveraging the normalized confidence derived from multiple candidate responses, ANSPRE mitigates the effect of spurious correlations between visual features and textual prompts—an issue commonly encountered in visual question answering systems.

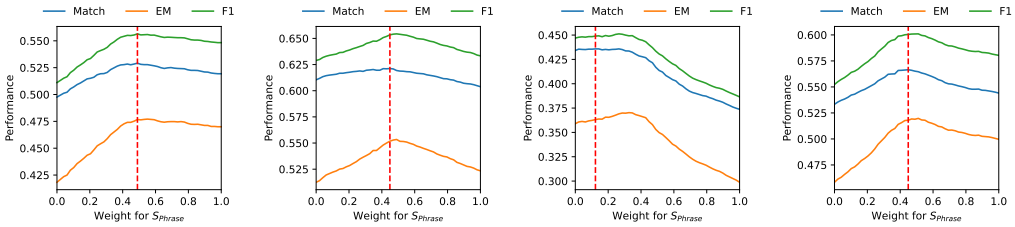
Nonetheless, a slight decrease in Match accuracy is observed in certain models after applying ANSPRE. This degradation is likely attributed to the baseline models' inclination to produce more verbose or descriptive re-

Model	InfographicVQA			DocVQA		
	Match	EM	F1	Match	EM	F1
InternVL2$_{8B}$	53.84	0.82	12.67	81.27	7.66	37.98
InternVL2$_{8B}$-ANSPRE	56.41$_{\uparrow 2.57}$	18.85$_{\uparrow 18.03}$	31.30$_{\uparrow 18.63}$	74.59$_{\downarrow 6.68}$	30.73$_{\uparrow 23.07}$	50.93$_{\uparrow 12.95}$
Llama-3.2-Vision$_{11B}$	41.31	0.04	2.63	48.55	1.89	9.01
Llama-3.2-Vision$_{11B}$-ANSPRE	49.66$_{\uparrow 8.35}$	0.18$_{\uparrow 0.14}$	3.35$_{\uparrow 0.72}$	62.95$_{\uparrow 14.4}$	18.45$_{\uparrow 16.56}$	27.91$_{\uparrow 18.9}$
Qwen2-VL-Instr$_{2B}$	50.84	0.36	11.63	79.88	0.60	32.25
Qwen2-VL-Instr$_{2B}$-ANSPRE	49.30$_{\downarrow 1.54}$	28.31$_{\uparrow 27.95}$	39.33$_{\uparrow 27.7}$	73.90$_{\downarrow 5.98}$	61.13$_{\uparrow 60.53}$	74.25$_{\uparrow 42.0}$
Qwen2-VL-Instr$_{7B}$	64.30	1.75	17.05	86.30	0.45	35.13
Qwen2-VL-Instr$_{7B}$-ANSPRE	63.16$_{\downarrow 1.14}$	36.42$_{\uparrow 34.67}$	50.94$_{\uparrow 33.89}$	79.60$_{\downarrow 6.7}$	66.93$_{\uparrow 66.48}$	80.0$_{\uparrow 44.87}$
LLaVA-v1.6-Vicuna$_{7B}$	23.03	0.00	2.78	48.10	1.51	17.97
LLaVA-v1.6-Vicuna$_{7B}$-ANSPRE	26.92$_{\uparrow 3.89}$	4.28$_{\uparrow 4.28}$	10.00$_{\uparrow 7.22}$	48.01$_{\downarrow 0.09}$	31.41$_{\uparrow 29.9}$	43.84$_{\uparrow 25.87}$

Table 6.5: Results on two VQA tasks comparing baseline multimodal LLMs with and without ANSPRE. Each pair of rows contrasts the performance of a model with ANSPRE and its version without ANSPRE.

sponses when prompted with visual inputs, leading to mismatches with concise ground-truth answers. Despite this minor limitation, the overall trend indicates that ANSPRE provides a robust and generalizable calibration framework for enhancing the reasoning consistency of multimodal LLMs across diverse visual question answering benchmarks.

6.4.2 Analysis



(a) Llama2$_{7B}$ (b) Gemma$_{7B}$ (c) OTP$_{6.7B}$ (d) Mistral$_{7B}$

Figure 6.3: Effect of ANSPRE weight term w_{Phrase}. The dotted red line indicates the weight with the highest Match accuracy.

Impact of ANSPRE Weighting.

Figure 6.3 illustrates how varying the weighting parameters w_{Phrase} and w_{Sentence} influences the performance of the Llama2 model. These weights are constrained by $w_{\text{Phrase}} + w_{\text{Sentence}} = 1$, ensuring that emphasizing one component proportionally reduces the contribution of the other. The experimental results reveal that the model achieves optimal performance when both weights are approximately equal. This observation suggests that phrase-level parametric knowledge ($\mathcal{S}_{\text{Phrase}}$) and sentence-level non-parametric knowledge ($\mathcal{S}_{\text{Sentence}}$) contribute complementary strengths to the reasoning process, and neither should be overly prioritized.

A balanced weighting enables the model to leverage the precision of phrase-level representations, which capture fine-grained lexical and syntactic cues, while simultaneously benefiting from sentence-level representations that encode broader contextual and semantic relationships. This synergy allows ANSPRE to better align retrieved information with the model's internal knowledge, leading to improved answer consistency and confidence calibration.

Notably, this trend is consistent across other LLM architectures beyond Llama2. Comparable performance gains are observed when maintaining a balanced ratio between w_{Phrase} and w_{Sentence}, highlighting the robustness and architectural independence of the proposed weighting scheme. These findings emphasize that both parametric and non-parametric components play indispensable roles in enhancing retrieval-augmented generation systems. Future investigations could explore whether this equilibrium persists across multilingual or multimodal tasks, thereby shedding light on the universality of ANSPRE's weighting dynamics under diverse linguistic and contextual conditions.

Effect of Aggregation Strategies.

Table 6.6 compares the performance of the ANSPRE framework under different aggregation strategies across multiple large language models (LLMs). Three configurations are evaluated: (1) no aggregation, where each generated answer is treated independently; (2) aggregation based on selecting the best sampling across all retrieved documents ("Best sampling across docs"); and (3) aggregation of samplings within the most relevant document ("Best sampling within doc"). To further validate the generalizability of these approaches, Table 6.7 reports the corresponding results on Vietnamese datasets, providing insight into how aggregation techniques perform in a non-English linguistic environment.

Overall, the results consistently show that incorporating aggregation mechanisms yields substantial performance gains across all metrics, datasets, and model families. This improvement can be attributed to the ensemble-like effect of aggregation, which mitigates random generation noise and amplifies high-confidence responses. By integrating evidence from multiple samplings, ANSPRE effectively enhances answer stability and reduces variance in the output quality.

Interestingly, neither of the two aggregation strategies emerges as a universal winner. The relative effectiveness of "across-document" versus "within-document" aggregation appears to depend on factors such as the underlying model architecture, retrieval diversity, and document relevance distribution. For instance, larger models with stronger retrieval alignment may benefit more from "within-document" aggregation, whereas smaller models often gain from "across-document" aggregation, which introduces broader contextual diversity.

The relatively small performance gap between the two methods suggests that both are viable and complementary, and the optimal choice may vary with the task characteristics and the LLM's inductive biases. These findings highlight the importance of model- and dataset-specific tuning when selecting aggregation strategies. In practice, ANSPRE demonstrates robustness to this choice, indicating that aggregation—regardless of its specific form—serves as a key mechanism for improving answer consistency and reliability in retrieval-augmented generation systems.

Model	Aggregator	TriviaQA			PopQA			NaturalQuestion		
		Match	EM	F1	Match	EM	F1	Match	EM	F1
$Llama2_{7B}$	None	37.59	31.52	37.71	**43.46**	39.03	44.06	14.71	12.88	17.73
	Best samp across docs	46.66	39.68	47.49	40.39	36.31	41.04	20.86	19.14	25.08
	Samps in best doc	**52.85**	**47.61**	**55.55**	42.17	**43.10**	**47.01**	**24.90**	**23.71**	**30.73**
$Llama2_{13B}$	None	29.89	24.91	29.69	38.81	35.17	38.20	17.87	16.26	21.77
	Best samp across docs	41.61	35.46	42.04	32.74	29.38	32.68	23.77	22.30	29.15
	Samps in best doc	**58.51**	**53.64**	**61.78**	**46.03**	**44.39**	**47.98**	**30.36**	**29.34**	**37.52**
$Llama3_{8B}$	None	59.48	53.21	62.55	**48.96**	45.68	49.85	29.17	26.45	36.07
	Best samp across docs	63.02	57.32	66.73	41.53	39.96	43.55	**31.97**	**29.92**	**40.39**
	Samps in best doc	**63.12**	**57.75**	**67.46**	**48.96**	**48.03**	**51.83**	30.50	29.36	39.58
$Gemma_{7B}$	None	60.29	50.38	61.92	49.82	44.17	**49.10**	33.38	27.51	39.52
	Best samp across docs	**64.30**	55.15	**66.40**	45.96	42.82	45.99	**36.32**	**31.52**	**43.80**
	Samps in best doc	61.89	**55.18**	65.35	**46.53**	**44.89**	47.88	31.58	28.48	39.46
$OPT_{6.7B}$	None	41.52	34.80	42.47	**42.24**	**38.88**	**42.92**	22.94	20.33	28.39
	Best samp across docs	**46.38**	**39.94**	**47.86**	33.45	30.81	34.10	**26.29**	**23.82**	**32.16**
	Samps in best doc	41.52	35.16	43.19	40.10	37.60	41.21	19.83	18.64	26.23
$Baichuan2_{7B}$	None	50.20	43.94	52.11	**46.75**	**45.10**	**48.09**	23.07	19.53	28.96
	Best samp across docs	53.14	47.00	55.39	39.17	38.38	40.54	**24.93**	21.22	**31.16**
	Samps in best doc	**54.98**	**48.48**	**57.88**	41.24	40.81	43.92	24.43	**21.33**	31.05
$Baichuan2_{13B}$	None	52.44	45.23	54.14	**48.32**	**45.46**	**49.36**	28.56	24.35	34.88
	Best samp across docs	**56.19**	**49.46**	**58.39**	39.39	37.74	41.03	**30.39**	**26.93**	**37.23**
	Samps in best doc	53.89	48.04	57.07	40.53	39.53	42.85	27.34	25.40	34.90
$Mistral_{7B}$	None	58.62	50.29	60.53	51.11	**46.82**	**51.48**	33.19	28.17	38.98
	Best samp across docs	**63.27**	**55.30**	**65.67**	44.75	42.17	45.46	**35.76**	**31.39**	**42.44**
	Samps in best doc	61.11	55.13	64.88	**47.53**	45.82	49.97	32.52	29.89	40.23
$Qwen_{7B}$	None	55.34	43.63	56.16	48.61	40.46	48.01	31.14	21.80	34.62
	Best samp across docs	**60.40**	**49.76**	**61.79**	43.82	39.81	45.10	**34.63**	**25.90**	**39.21**
	Samps in best doc	57.09	48.00	59.64	**48.61**	**46.18**	**50.94**	30.72	24.43	36.17
$Qwen_{14B}$	None	58.68	46.86	59.37	**51.82**	45.32	**50.68**	31.50	18.56	32.64
	Best samp across docs	**63.65**	**52.65**	**64.70**	44.75	41.67	45.53	**34.93**	22.19	**36.80**
	Samps in best doc	61.00	52.00	63.48	47.75	**45.89**	49.40	29.53	**22.91**	35.07
$Qwen2_{7B}$	None	58.68	44.47	58.12	**50.39**	43.10	48.57	33.13	22.85	35.64
	Best samp across docs	**63.76**	50.29	**63.47**	41.39	38.46	41.39	**37.31**	**27.70**	**41.00**
	Samps in best doc	60.30	**50.49**	62.41	49.54	**46.32**	**50.38**	33.19	27.20	38.42
$Qwen2.5_{7B}$	None	58.43	42.47	56.98	**50.39**	39.10	47.42	31.36	21.08	33.64
	Best samp across docs	**63.11**	47.56	**62.00**	40.53	34.38	39.41	**34.93**	**25.07**	**38.33**
	Samps in best doc	59.13	**48.61**	60.81	49.11	**44.82**	**49.41**	30.83	24.02	35.61
$Phi\text{-}2_{2.7B}$	None	48.69	36.37	48.68	**50.25**	42.67	49.02	**28.03**	19.47	31.65
	Best samp across docs	**54.66**	**42.73**	**54.91**	40.39	36.53	40.69	31.02	**23.57**	**36.07**
	Samps in best doc	49.80	40.60	51.81	48.61	**45.18**	**49.46**	25.90	20.61	31.22

Table 6.6: Performance of LLMs with different aggregator settings: no aggregator (None), aggregation of the best samplings across all documents (Best samp across docs), and aggregation of samplings within the best document (Samps in best doc).

Model	Aggregator	ViQUAD Match	ViQUAD EM	ViQUAD F1	VIMQA Match	VIMQA EM	VIMQA F1
Llama-3_{8B}	None	**11.22**	8.55	**23.07**	17.25	14.36	28.03
	Best samp across docs	10.41	**9.00**	21.96	**18.74**	**16.55**	**29.04**
	Samps in best doc	8.78	7.19	20.80	17.15	14.66	28.34
Llama-3.1_{8B}	None	**11.58**	8.60	23.43	**18.44**	14.46	28.43
	Best samp across docs	11.49	**9.28**	**23.67**	18.25	**15.35**	28.63
	Samps in best doc	8.69	7.06	20.57	17.25	14.86	**28.87**
Llama-3.2_{3B}	None	6.47	4.62	13.34	**12.86**	10.37	**21.54**
	Best samp across docs	**6.74**	**4.84**	**13.59**	12.46	11.17	21.49
	Samps in best doc	5.66	4.21	12.35	11.76	**11.37**	20.99
$Qwen2_{7B}$	None	**9.59**	6.24	17.88	**0.00**	**9.09**	**16.80**
	Best samp across docs	**9.59**	**7.19**	**18.07**	**0.00**	**9.09**	16.36
	Samps in best doc	7.19	5.70	15.39	**0.00**	**9.09**	16.79
$Qwen2.5_{7B}$	None	**13.30**	8.19	24.63	**15.55**	10.77	24.51
	Best samp across docs	12.94	**8.73**	**25.06**	15.45	11.37	**25.11**
	Samps in best doc	10.27	7.96	22.40	14.06	**11.47**	24.45

Table 6.7: Performance of multilingual LLMs with different aggregator settings: no aggregator (None), aggregation of the best samplings across all documents (Best samp across docs), and aggregation of samplings within the best document (Samps in best doc).

Effect of Grammar.

Grammatical structures vary considerably across languages, shaping not only how information is conveyed but also how it is interpreted in automated evaluation metrics such as Exact Match (EM) and F1. These variations can introduce systematic biases when assessing model outputs, especially in multilingual settings where syntactic and morphological conventions differ substantially from English.

Japanese, for instance, illustrates this challenge clearly. The language employs copular forms such as です, でした, である, which primarily serve pragmatic or stylistic purposes—indicating levels of formality or politeness rather than contributing to the core semantic meaning of a sentence. Because of this, two sentences expressing the same factual content may differ only in their copular choice. Such differences, though semantically equivalent, can lead to mismatches under surface-form metrics like EM and F1, resulting in artificially lower scores.

To mitigate this issue, we introduce a grammar-aware evaluation procedure in which copulas are removed prior to metric computation. This adjustment ensures that comparisons are grounded in semantic equivalence rather than stylistic variation. Table 6.8 presents the results of an ablation study

Model	Revised	JAQKET Match	JAQKET EM	JAQKET F1	JaQuAD Match	JaQuAD EM	JaQuAD F1
Llama-3_{8B}-ANSPRE		81.01	12.03	12.24	75.22	8.86	8.88
	✓	**81.10**	**52.23**	**52.35**	**75.25**	**35.54**	**35.54**
Llama-3_{8B}		65.03	18.13	26.33	65.02	13.63	18.47
	✓	**65.21**	**22.77**	**31.11**	**65.02**	**15.87**	**20.91**
Llama-3.1_{8B}-ANSPRE		80.76	5.07	5.34	75.65	4.54	4.56
	✓	**80.93**	**44.33**	**44.44**	**75.68**	**36.89**	**36.89**
Llama-3.1_{8B}		64.26	20.27	25.19	67.76	14.39	16.91
	✓	**64.52**	**24.83**	**30.09**	**67.78**	**18.00**	**20.71**
Llama-3.2_{3B}-ANSPRE		55.50	1.03	1.41	62.10	0.69	0.70
	✓	**55.58**	**28.09**	**28.30**	**62.10**	**24.98**	**25.00**
Llama-3.2_{3B}		57.04	1.12	4.76	53.47	0.71	2.42
	✓	**57.22**	**1.55**	**5.31**	**53.47**	**0.86**	**2.72**
$Qwen2_{7B}$-ANSPRE		77.41	0.52	0.77	72.66	0.46	0.49
	✓	**77.49**	**37.11**	**37.29**	**72.66**	**33.46**	**33.49**
$Qwen2_{7B}$		75.86	5.84	6.27	86.70	1.42	1.47
	✓	**75.86**	**10.57**	**11.03**	**86.70**	**2.26**	**2.31**
$Qwen2.5_{7B}$-ANSPRE		81.53	0.17	0.37	78.95	0.25	0.30
	✓	**81.70**	**34.28**	**34.30**	**78.98**	**37.62**	**37.64**
$Qwen2.5_{7B}$		74.83	4.90	5.91	88.14	1.85	1.98
	✓	**75.09**	**6.62**	**7.79**	**88.12**	**2.08**	**2.24**

Table 6.8: Comparison of Evaluation Versions on two Japanese ODQA tasks using ANSPRE: Original (Unaltered) vs. Revised (Copula Removed). Bold indicates the best result per group.

contrasting two evaluation configurations: (1) the *original evaluation*, which retains the full sentence structure, and (2) the *revised evaluation*, where copulas are systematically omitted.[1] The findings reveal that while the Match metric remains largely unaffected—since it depends on broader answer inclusion—the EM and F1 scores exhibit notable improvements under the revised setting, reflecting a more accurate measure of semantic consistency.

Interestingly, when the same grammar-sensitive evaluation is applied to baseline models (Table 6.8), the performance improvement is relatively modest. This can be attributed to the design of the ANSPRE framework, which structures outputs using a declarative template that positions the answer phrase at the end of the sentence. As a result, even when copulas are removed, the essential information remains intact and correctly aligned with the ground truth.

These observations emphasize that grammatical normalization plays a crucial role in ensuring fair cross-lingual evaluation. Language-specific grammatical elements—particularly those with stylistic or pragmatic functions—should

[1]The complete list of excluded copular forms is provided in Section 4.4.1.

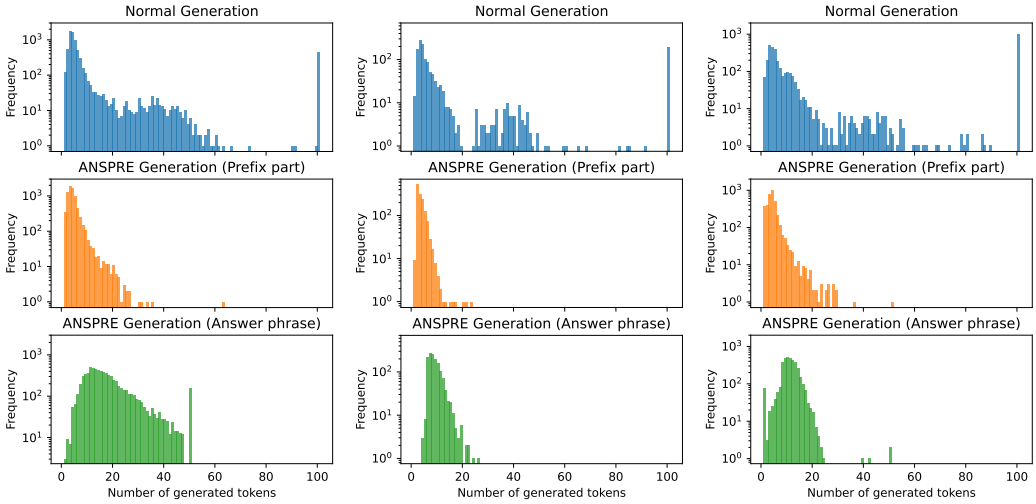
be carefully accounted for to prevent penalizing models for producing valid linguistic variants. In essence, grammar-aware evaluation enhances the fidelity of performance assessment, especially for languages with rich morphological or politeness systems such as Japanese.

Analysis of Computational Cost.

To assess the computational cost introduced by the proposed aggregation module and the ANSPRE generation procedure, we perform a comprehensive runtime analysis comparing standard generation with ANSPRE-based inference on three question-answering benchmarks: TriviaQA, PopQA, and NaturalQuestions. All experiments are conducted using the $\text{LLaMA3}_{\text{8B}}$ model, and we report the average decoding latency in seconds. For the ANSPRE setting, the decoding process is decomposed into two components: (i) prefix construction and (ii) generation of the final answer phrase, with the overall latency obtained by summing both stages. As summarized in Table 6.9, ANSPRE achieves lower end-to-end inference time on PopQA and NaturalQuestions, primarily because the answer phrase decoding step is substantially shorter than that of the baseline. This behavior is corroborated by the output length statistics shown in Figure 6.4, which indicate that ANSPRE produces more concise and targeted responses. Taken together, these findings suggest that ANSPRE not only enhances answer quality but also preserves computational efficiency, resulting in a favorable balance between performance and runtime.

Dataset	Generation time *(ms)*			
	Normal Generation	ANSPRE Generation		
		Prefix part	Answer phrase	Total
TriviaQA	178.49	59.79	292.71	352.49
PopQA	288.70	48.39	121.67	170.06
NaturalQuestion	614.84	59.17	150.26	209.44

Table 6.9: Average generation time of normal and ANSPRE methods on TriviaQA, PopQA, and NaturalQuestion.



(a) TriviaQA (b) PopQA (c) NaturalQuestion

Figure 6.4: Distribution of response lengths for $\text{LLaMA3}_{8\text{B}}$ across TriviaQA, PopQA, and NaturalQuestion under normal and ANSPRE generation methods.

Chapter 7

LLM Efficiency: Smaller LLMs through Optimal Pruning

7.1 Preliminaries on Pruning

7.1.1 Layer-wise Pruning

Post-training pruning is commonly applied in a *layer-wise* manner, where the pruning process is decomposed into independent optimization problems for each layer. The objective is to minimize the reconstruction error between the original dense layer and its pruned counterpart. Formally, for a given input $\mathbf{X}_\ell$ and weight matrix $\mathbf{W}_\ell \in \mathbb{R}^{r\times c}$ in the ℓ-th layer—where r and c denote the numbers of output and input channels, respectively—the goal is to determine a binary pruning mask $\mathbf{M}_\ell \in \{0,1\}^{r\times c}$ and reconstructed weights $\hat{\mathbf{W}}_\ell$ that minimize the following objective:

$$\arg\min_{\mathbf{M}_\ell, \hat{\mathbf{W}}_\ell} \|\mathbf{W}_\ell \mathbf{X}_\ell - (\mathbf{M}_\ell \odot \hat{\mathbf{W}}_\ell)\mathbf{X}_\ell\|_2^2 \tag{7.1}$$

This formulation ensures that the pruned layer approximates the behavior of the original dense layer as closely as possible while achieving a desired sparsity level.

7.1.2 State-of-the-Art Solvers

Layer-wise pruning algorithms, referred to here as *solvers*, can generally be grouped into two categories based on how they handle pruning masks and weight reconstruction.

Mask-based solvers. The first category determines a binary mask $\mathbf{M}_\ell$ while leaving the weights unchanged (i.e., $\hat{\mathbf{W}}_\ell = \mathbf{W}_\ell$). These methods typically compute an *importance score* for each weight parameter and iteratively remove the least important parameters until a target sparsity is reached. Early approaches relied on simple magnitude-based importance estimation [88], whereas recent methods, such as RIA [87], incorporate both parameter connectivity and activation information to produce more accurate importance scores.

Reconstruction-based solvers. The second category reconstructs the pruned weights $\hat{\mathbf{W}}_\ell$ using second-order information derived from the Hessian matrix [38, 24, 45, 17, 18, 67]. During reconstruction, the pruning mask $\mathbf{M}_\ell$ is selected adaptively to minimize the layer's output error. Recent work such as SparseGPT [18] has significantly improved the computational efficiency of this approach, reducing complexity while maintaining strong performance in large language model (LLM) pruning.

7.1.3 Non-uniform Sparsity

When pruning a model to a target sparsity level S, a straightforward approach is to apply *uniform sparsity*, where each layer i has the same sparsity ratio $S_i = S$ for all i. However, this uniform constraint may not be optimal, as different layers contribute unequally to model performance.

To address this, *non-uniform sparsity* assigns distinct sparsity ratios S_i to different layers while preserving the overall sparsity target. This relaxes the constraint $S_i = S$, but enforces the global condition:

$$\frac{1}{N}\sum_{i=1}^{N} S_i = S,$$

where N is the total number of layers.

Recent work such as OWL [84] computes layer-specific sparsity levels based on the distribution of outlier activations, referred to as the *Layerwise Outlier Distribution* (LOD). Layers containing more outliers are assigned lower sparsity (i.e., are pruned less aggressively), while layers with fewer outliers receive higher sparsity. For a given global sparsity S and outlier distribution vector $\mathbf{D} = [D^1, D^2, \ldots, D^N]$, the layerwise sparsity is computed proportionally to $(1 - D_i)$, subject to a deviation constraint:

$$S_i \in [S - \lambda, S + \lambda],$$

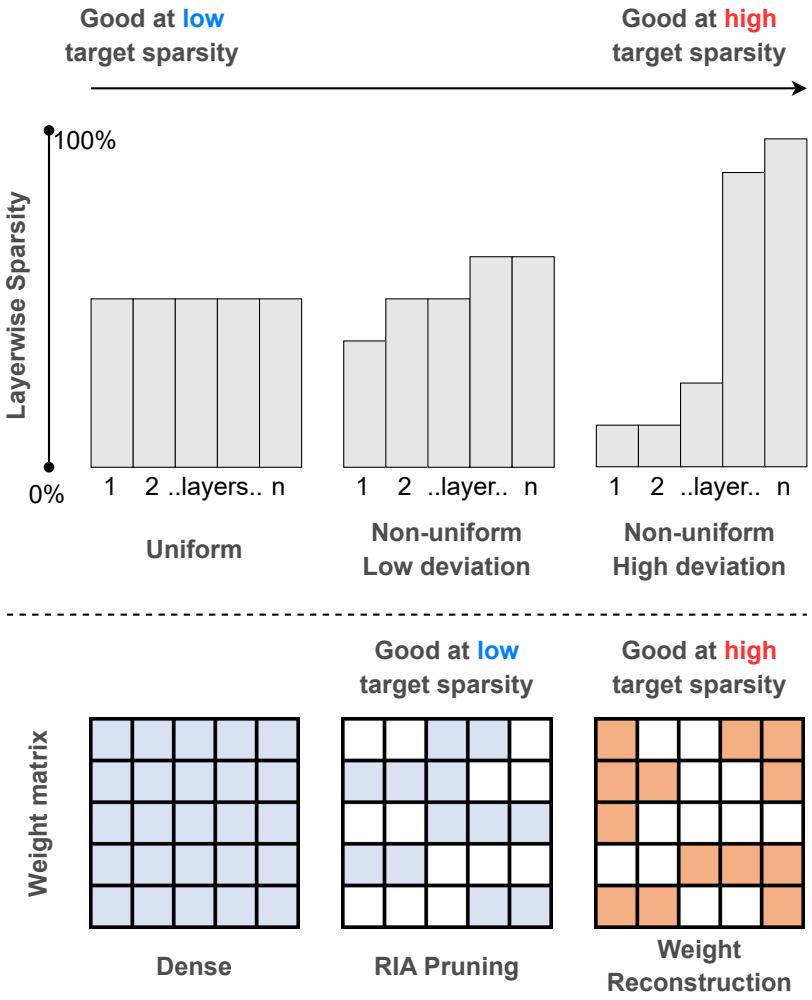


Figure 7.1: Overview of findings. **Top:** Uniform sparsity outperforms at low sparsity, while non-uniform sparsity achieves better performance at high sparsity. **Bottom:** Relative importance-based pruning (RIA) is more effective at low sparsity, whereas Hessian-based weight reconstruction performs better at high sparsity. Orange cells indicate layers with modified weights.

where λ is a hyperparameter controlling the allowable variation in per-layer sparsity. Figure 7.4a illustrates how different values of λ influence the resulting sparsity distribution across layers.

7.2 Findings Overview

State-of-the-art pruning approaches commonly adopt a *layerwise* formulation, where each layer is pruned independently to minimize the deviation between the outputs of the original and pruned layers. Most existing methods assume **uniform sparsity**, meaning that all layers are pruned to the same sparsity level. Pruning can be implemented either by (1) learning a binary sparsity mask or (2) reconstructing the weights to approximate the dense model.

Mask-based methods often rely on magnitude pruning [22] or parameter importance estimation, such as the relative importance and activation (RIA) approach [87]. In contrast, reconstruction-based approaches [45, 17, 18] leverage second-order information from the Hessian matrix to iteratively refine weights, typically selecting masks adaptively during optimization.

Several studies have explored **non-uniform sparsity**, in which the sparsity ratio differs across layers while preserving the global sparsity target [16, 41, 76, 39, 49, 84]. However, our experiments reveal that these strategies tend to perform optimally within specific sparsity ranges and lose efficiency beyond them. Moreover, little prior work examines how such pruning strategies affect multilingual or cross-lingual capabilities. To address these gaps, we systematically evaluate various pruning solvers and sparsity allocation schemes across different target sparsity levels on multiple LLM architectures.

- $\mathcal{F}1$: **Uniform vs. Non-uniform Sparsity.** Uniform sparsity performs better at low sparsity levels, while non-uniform sparsity provides significant gains at high sparsity. Increasing deviation from uniformity improves performance at high sparsity but reduces it at low sparsity.
- $\mathcal{F}2$: **Solver Type.** Importance-based mask pruning (e.g., RIA) achieves superior results at low sparsity, whereas Hessian-based weight reconstruction methods excel at high sparsity.

Figure 7.1 summarizes these findings. We empirically validate them using state-of-the-art pruning solvers and widely adopted LLM architectures. To facilitate fair comparison and deeper analysis, all methods were re-implemented within a unified framework that allows flexible configuration of solver components and supports recent architectures such as Llama3 [14].

7.3 Empirical Study

7.3.1 Study on Uniform vs. Non-uniform Sparsity ($\mathcal{F}1$)

We investigate how uniform and non-uniform sparsity affect model performance across different sparsity levels. The comparison is based on model perplexity, where lower values indicate better performance. Additionally, we study how deviations from uniform sparsity (controlled by the parameter λ) influence non-uniform pruning effectiveness.

Experimental Setup. We employ two representative pruning methods: SparseGPT [18], which reconstructs weights using Hessian information, and RIA [87], which estimates parameter importance through relative activations and connectivity. Non-uniform sparsity ratios are computed using OWL [84], which derives layerwise sparsity based on outlier distributions. Experiments are conducted on Llama2 (7B, 13B), Llama3 (8B), and OPT (6.7B) models, with target sparsity levels ranging from 10% to 80%. Perplexity is evaluated using the Wikitext2 test set [54]. To assess the effect of deviation, we vary $\lambda \in \{0.01, 0.08, 0.15\}$ for SparseGPT on Llama2 (7B).

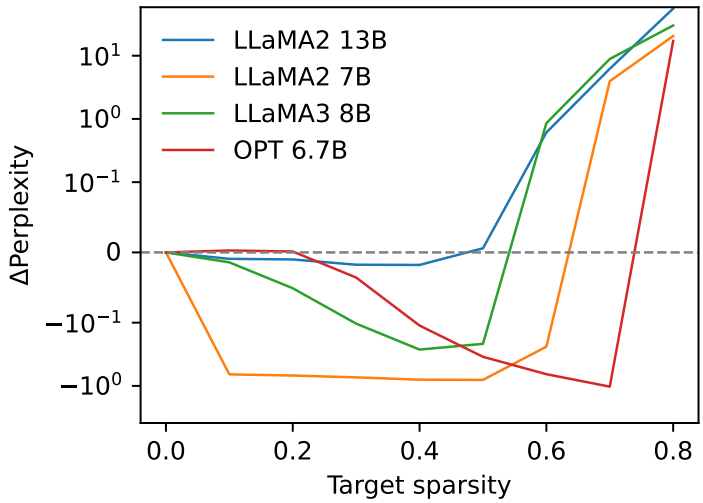


Figure 7.2: Perplexity difference (Δ Perplexity) between uniform and non-uniform sparsity using SparseGPT. Negative Δ values indicate uniform sparsity performs better, while positive values indicate non-uniform sparsity is superior.

Results on Uniform vs. Non-uniform Sparsity. Figure 7.2 presents the performance difference between uniform and non-uniform sparsity under SparseGPT. Non-uniform sparsity consistently outperforms uniform sparsity

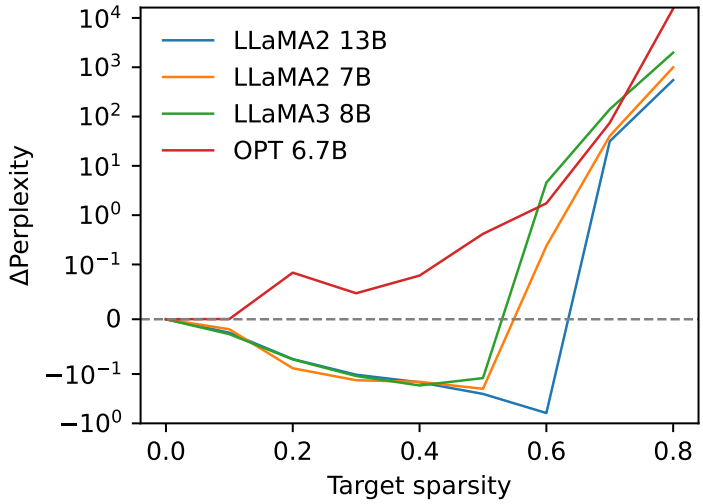


Figure 7.3: Perplexity difference (Δ Perplexity) between RIA (mask-based) and SparseGPT (reconstruction-based). Negative Δ values indicate RIA performs better; positive values indicate SparseGPT performs better.

at higher target sparsity levels, while uniform sparsity performs slightly better at lower sparsity levels. This trend holds across different architectures.

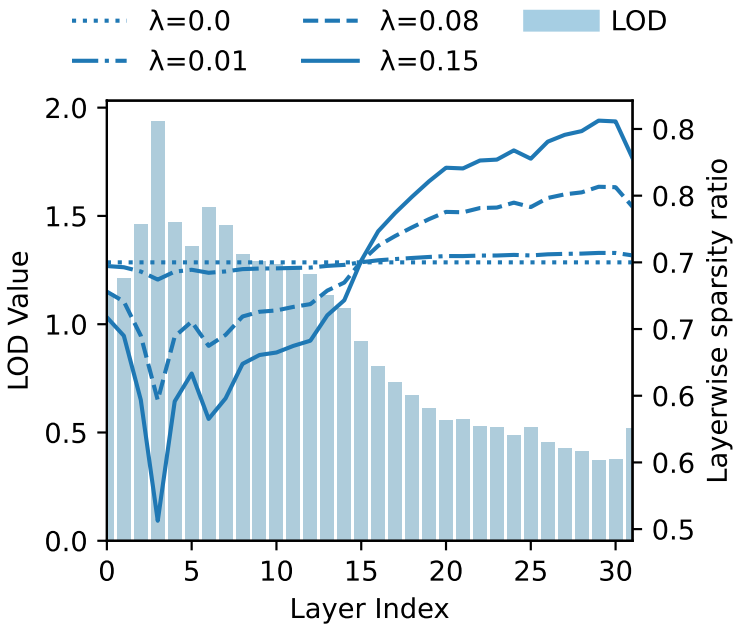
Effect of Deviation in Non-uniform Sparsity. Figure 7.4a visualizes layerwise sparsity distributions for varying λ values at a target sparsity of 70%. Figure 7.4b shows that smaller λ values (closer to uniform sparsity) lead to better performance at low sparsity (10–50%), whereas larger λ values improve performance at higher sparsity (60–80%). This indicates that flexibility in sparsity allocation becomes increasingly beneficial as overall sparsity increases.

7.3.2 Study on State-of-the-art Solvers ($\mathcal{F}2$)

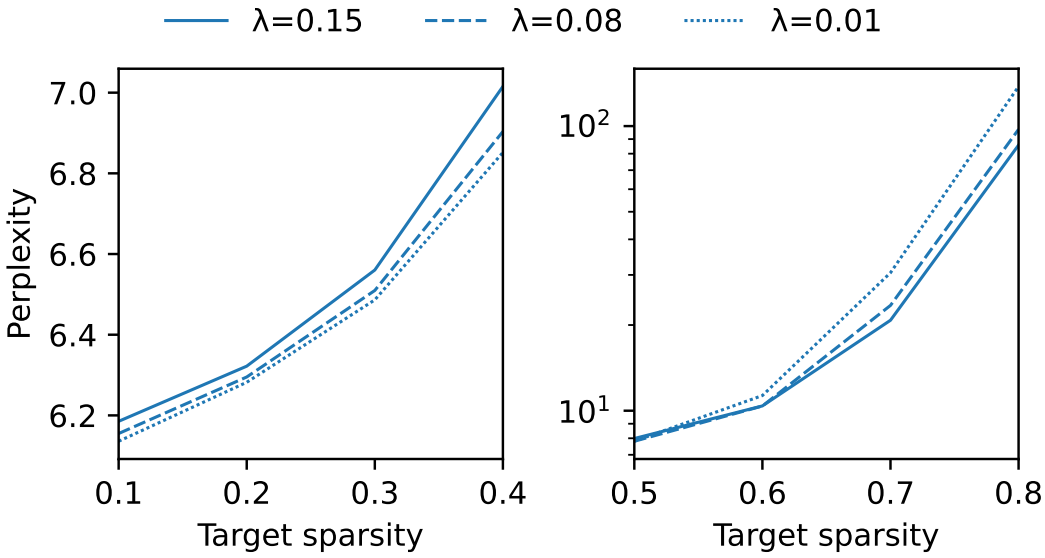
We further compare two representative solvers: RIA and SparseGPT. RIA identifies sparse masks based on relative importance and activation information without altering weight values, while SparseGPT performs layerwise weight reconstruction using the Hessian matrix and adaptively determines the pruning mask.

Experimental Setup. Both solvers are evaluated on Llama2 (7B, 13B), Llama3 (8B), and OPT (6.7B) models across target sparsity levels from 10% to 80%. Model perplexity is measured using the Wikitext2 dataset [54].

Results. As shown in Figure 7.3, RIA achieves lower perplexity at low sparsity levels, indicating its advantage when retaining most parameters.



(a) Layerwise non-uniform sparsity for different λ values at 70% target sparsity. $\lambda = 0.0$ corresponds to uniform sparsity. Bars represent each layer's outlier distribution (LOD).



(b) Perplexity comparison of different λ values on Llama2 (7B) pruned by SparseGPT. Lower perplexity indicates better performance.

Figure 7.4: Effect of deviation parameter λ on pruning performance.

Conversely, SparseGPT surpasses RIA at higher sparsity levels, where its

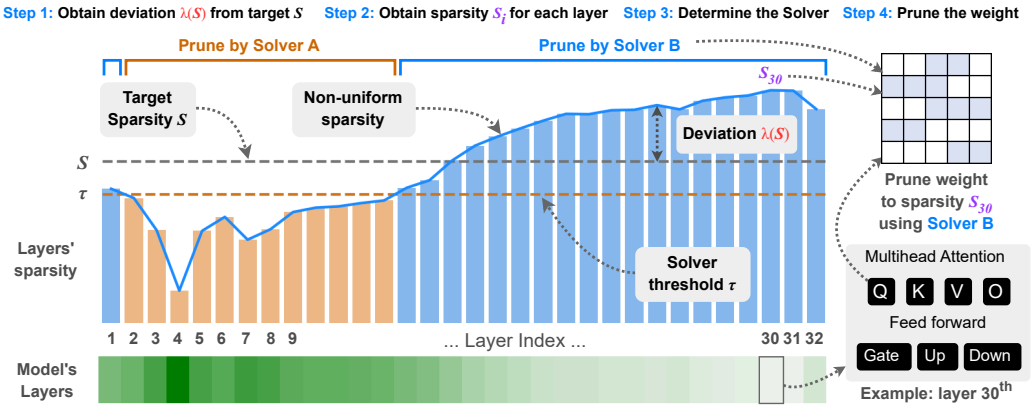


Figure 7.5: Illustration of *OptiPrune* applied to Llama2 (7B). Each layer consists of seven weight matrices: four for multi-head attention (Q, K, V, O) and three for the feed-forward network (Gate, Up, Down). Shades of green represent relative layer importance based on the outlier distribution.

weight reconstruction mechanism becomes beneficial. This crossover trend is consistent across all model architectures. Further analysis in Appendix 9.3.5 confirms that direct weight reconstruction may slightly degrade performance at low sparsity but substantially improves model quality at high sparsity.

7.4 OptiPrune

Building on our validated findings, we propose OptiPrune, a pruning method designed to handle models effectively across different target sparsity levels. Figure 7.5 provides an overview of the method. Since lower deviation values benefit low sparsity and higher values benefit high sparsity (**Finding** $\mathcal{F}1$), we first compute a deviation level $\lambda(S)$ based on the target sparsity S (Step 1). Next, we derive the non-uniform layerwise sparsity S_i (Step 2). To further optimize performance, we compare S_i with a threshold τ to select the pruning solver (**Finding** $\mathcal{F}2$) (Step 3). Finally, we prune each layer using the chosen solver and its corresponding sparsity S_i (Step 4). The following subsections detail these steps.

7.4.1 Determining the Deviation Level

The parameter λ controls how much layerwise sparsity deviates from uniform sparsity. While prior work [84] uses a fixed λ, we dynamically adjust λ based on the target sparsity S, as suggested by **Finding** $\mathcal{F}1$. Specifically,

λ increases with S, allowing flexible adaptation across sparsity ranges. We define $\lambda \in [\lambda_{\min}, \lambda_{\max}]$ and compute it by linear interpolation:

$$\lambda(S) = \lambda_{\min} + \frac{(S - S_{\min})(\lambda_{\max} - \lambda_{\min})}{S_{\max} - S_{\min}}.$$

In all experiments, we use linear interpolation with $\lambda_{\min} = 0.01$, $\lambda_{\max} = 0.16$, and sparsity bounds $S_{\min} = 0\%$, $S_{\max} = 100\%$.

7.4.2 Computing Layerwise Non-uniform Sparsity

With $\lambda(S)$ determined, we compute layerwise sparsity S_i following the OWL method [84] (see Section 7.1.3). Let $\mathbf{D} = [D^1, D^2, \ldots, D^n]$ denote the outlier distributions across layers. We assign

$$S_i \propto 1 - D_i,$$

and scale it within $[S - \lambda, S + \lambda]$, ensuring the global sparsity target S is maintained. Layers with more outliers receive lower sparsity (i.e., retain more weights).

7.4.3 Adaptive Solver Selection

We employ two state-of-the-art solvers: RIA and SparseGPT. According to **Finding** $\mathcal{F}2$, RIA performs better at low sparsity, while SparseGPT excels at high sparsity. We introduce a threshold τ to automatically select the appropriate solver per layer:

$$\text{Solver}_i = \begin{cases} \text{RIA}, & \text{if } S_i < \tau, \\ \text{SparseGPT}, & \text{otherwise.} \end{cases}$$

RIA estimates importance based on activation sensitivity, while SparseGPT reconstructs weights to minimize error after pruning.

7.4.4 Layerwise Pruning

Finally, we prune each layer using the selected solver and its target sparsity S_i. For transformer-based LLMs, pruning is applied to main weight matrices in multi-head attention (`Q`, `K`, `V`, `O`) and feed-forward layers (e.g., 3 matrices for LLaMA-2, 2 for OPT). This process produces models that maintain performance while achieving the desired sparsity distribution.

Chapter 8

LLM Efficiency: Faster LLMs through Retrieval-based Speculative Decoding

8.1 Preliminaries on Speculative Decoding

8.1.1 Autoregressive Decoding in LLMs

Large Language Models (LLMs) generate text in a sequential manner, predicting one token at a time conditioned on previously observed inputs. Formally, given an input sequence $\mathbf{x} = (x_1, x_2, \ldots, x_s)$ of length s, we denote the prefix of length m as $\mathbf{x}_{1:m} = (x_1, x_2, \ldots, x_m)$. At each decoding step t, an LLM outputs a probability distribution over the vocabulary for the next token. The probability of generating token x_t is expressed as:

$$P_M(x_t \mid \mathbf{x}_{1:t-1}), \tag{8.1}$$

where M refers to the model parameters.

During generation, a sampling strategy is required to transform this predictive distribution into a concrete token. Common approaches include greedy decoding, top-k sampling, and nucleus (top-p) sampling [34, 26]. Under greedy decoding, the next token is chosen by selecting the most likely candidate:

$$x_t =_v \ P_M(v \mid \mathbf{x}_{1:t-1}). \tag{8.2}$$

By recursively applying the token generation process, the model produces an autoregressive output sequence $\mathbf{y} = (y_1, y_2, \ldots, y_m)$ of length m, where each token is conditioned on both the input and previously generated outputs:

$$y_i =_v \ P_M(v \mid \mathbf{y}_{1:i-1}, \mathbf{x}). \tag{8.3}$$

This autoregressive procedure ensures that each prediction incorporates contextual information from all prior tokens, enabling coherent text generation but requiring a forward pass per generated token, which can be computationally expensive.

8.1.2 Speculative Decoding

Speculative decoding is designed to accelerate autoregressive generation by predicting and verifying multiple future tokens in parallel. The method follows a *draft-and-validate* paradigm: a lightweight or auxiliary model first generates multiple candidate continuations, and the main LLM subsequently validates them in fewer decoding steps.

Specifically, consider a drafting process that proposes G different continuations, each of length K. Let the set of all speculative hypotheses be:

$$\tilde{Y} = \{\tilde{y}^{(1)}, \tilde{y}^{(2)}, \ldots, \tilde{y}^{(G)}\}, \tag{8.4}$$

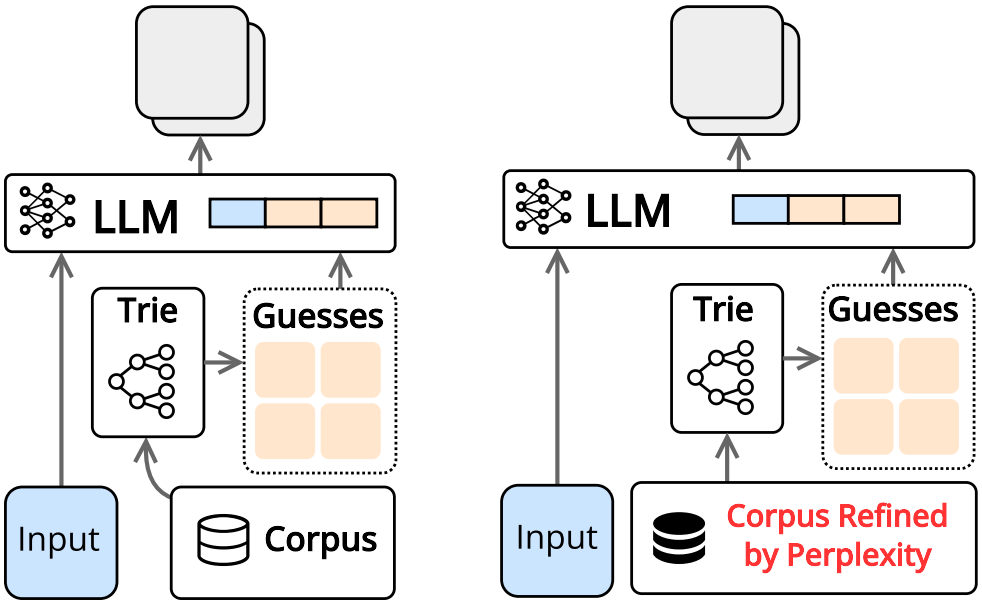
where $\tilde{y}_j^{(i)}$ denotes the j-th token in the i-th draft. Using tree attention mechanisms [55], multiple candidates can be verified simultaneously using a single forward pass of the larger model.

During verification, the primary LLM computes the correct next tokens $(y'_1, y'_2, \ldots, y'_K)$ in parallel. The verification procedure then identifies the longest prefix h for which a draft sequence is consistent with the validated output. Thus, instead of generating only one token, up to $h{+}1$ tokens can be committed in a single decoding iteration.

This yields substantial inference-time speedups by reducing the number of sequential decoding steps without compromising output fidelity.

8.2 Faster LLMs with Efficient Retrieval-based Speculative Decoding

Our proposed retrieval-based speculative decoding method introduces an external knowledge–driven strategy to enhance speculative decoding by incorporating retrieved text as candidate guesses. The key idea is to preprocess a large text corpus into an efficient data structure, such as a trie or compressed index, which allows rapid prefix-based lookup. During decoding, the most recently generated tokens serve as a query prefix to retrieve sequences that are likely to continue the current context in a linguistically coherent manner. These retrieved sequences then act as speculative drafts for verification by



(a) Retrieval-based speculative decoding [25].

(b) Our improved retrieval-based speculative decoding approach.

Figure 8.1: Comparison of (a) Retrieval-based speculative decoding [25] and (b) Our improved approach.

the main model. Figure 8.1 compares the existing retrieval-based speculative decoding method [25] and our improved approach.

Using arbitrary segments from a corpus, however, may lead to low-quality guesses that are often rejected during verification, thereby limiting the acceleration benefits of speculative decoding. To mitigate this, we introduce a refinement step that selects corpus segments most compatible with the target LLM.

Corpus Refinement via Perplexity. To improve the relevance of retrieved guesses, we evaluate each text sequence $u = (u_1, u_2, \ldots, u_t)$ using perplexity, which measures the average uncertainty of the model when predicting each token:

$$\mathrm{PPL}(u) = \exp\left(-\frac{1}{t}\sum_{i=1}^{t}\log P_M(u_i \mid u_{<i})\right). \tag{8.5}$$

A lower perplexity indicates that the sequence is more predictable under the model, suggesting it aligns well with the model's learned distribution and can generate high-quality guesses.

We leverage this property to construct a curated subset of the corpus by selecting sequences with the lowest perplexity. This high-quality subset is then indexed into a trie structure for efficient prefix-based retrieval. By prioritizing sequences that the model naturally predicts with high confidence, our method improves the acceptance rate of speculative guesses, enhances overall decoding efficiency, and facilitates smoother integration with other speculative decoding strategies.

Importantly, this refinement approach is model-agnostic and complements other speculative decoding techniques. By aligning retrieval with the internal probabilistic behavior of the LLM, our method ensures that external knowledge is effectively leveraged while maintaining high generation speed and accuracy.

Chapter 9

Experiments and Results on Efficient LLM Techniques

9.1 Tasks and Datasets

We assess the effectiveness of the proposed method on two major categories: *language modeling* and *zero-shot classification*. For language modeling, we report the perplexity (PPL) metric, where lower values indicate better predictive performance. This evaluation is conducted on the `WikiText-2` dataset.

For zero-shot classification, we examine model generalization across a diverse suite of benchmarks, including `Hellaswag`, `BoolQ`, `ARC`, `MNLI`, `QNLI`, `RTE`, `OpenBookQA`, `Winogrande`, and `MathQA`. These datasets collectively cover commonsense reasoning, natural language inference, reading comprehension, and mathematical reasoning. Perplexity is assessed on the Wikitext2 [54]. For zero-shot evaluation, we employ a diverse set of benchmarks covering commonsense reasoning, natural language inference, and question answering. These include Hellaswag [85], BoolQ [10], ARC (Challenge and Easy) [11], MNLI [81], QNLI [75], RTE [13], OpenBookQA [56], Winogrande [64], and MathQA [3].

9.2 Baselines

We compare our approach against several strong, publicly available large language models (LLMs) and recent state-of-the-art pruning techniques. The baseline LLMs include `Llama2` (7B, 13B) [73], `Llama3` (8B) [14], and `OPT` (6.7B) [86].

For pruning methods, we benchmark against `SparseGPT` [18], `Wanda` [67], `RIA` [87], and `OWL` [84]. Since OWL focuses on determining non-uniform spar-

sity and requires an underlying pruning algorithm, we integrate `SparseGPT` as its backbone method, following the configuration reported to yield the best performance in [84].

Method	Target sparsity 10%	20%	30%	40%	50%	60%	70%	80%	Avg
Llama2-7B (Dense=53.83)									
OptiPrune (Ours)	53.83	54.06	53.61	**53.39**	**51.50**	**48.05**	**42.83**	**36.79**	**49.26**
OWL	53.89	53.81	53.03	52.67	50.98	47.65	42.36	36.20	48.82
RIA	53.80	**54.17**	**53.79**	53.11	50.29	46.10	36.40	33.90	47.70
SparseGPT	53.88	53.49	53.35	52.36	50.12	47.25	39.89	33.48	47.98
Wanda	**53.90**	54.09	53.26	51.86	49.72	43.46	34.88	33.60	46.85
Llama2-13B (Dense=56.60)									
OptiPrune (Ours)	**56.35**	55.81	55.40	**55.79**	**54.35**	**51.69**	**46.25**	**38.35**	**51.75**
OWL	56.25	**56.14**	55.87	54.20	54.33	51.07	45.53	37.59	51.37
RIA	56.34	55.99	55.65	55.29	53.89	50.60	39.65	33.44	50.11
SparseGPT	56.35	55.98	**55.89**	55.12	54.08	50.51	42.74	35.30	50.75
Wanda	56.19	56.10	55.65	55.45	53.33	48.05	35.81	33.20	49.22
Llama3-8B (Dense=58.62)									
OptiPrune (Ours)	59.06	**58.82**	57.74	**56.72**	53.45	**50.74**	**42.45**	**36.41**	**51.92**
OWL	**59.15**	58.78	**58.06**	56.60	**54.05**	49.85	41.66	36.39	51.82
RIA	58.72	58.52	57.08	55.31	52.42	45.21	34.51	33.42	49.40
SparseGPT	59.10	58.70	57.75	56.59	53.42	48.64	40.20	34.91	51.16
Wanda	59.10	58.60	57.36	54.53	50.97	44.08	35.91	33.54	49.26
OPT-6.7B (Dense=46.89)									
OptiPrune (Ours)	**47.59**	46.98	47.11	**46.64**	45.52	44.18	**41.83**	**37.60**	**44.68**
OWL	47.00	**47.19**	**47.19**	46.59	45.56	44.18	41.54	37.15	44.55
RIA	46.94	46.88	46.72	46.41	45.50	43.31	36.10	35.63	43.44
SparseGPT	47.00	46.88	47.15	46.54	**45.92**	**44.61**	41.82	37.21	44.64
Wanda	46.97	46.93	46.89	46.24	43.35	38.06	35.11	33.15	42.09

Table 9.1: Zero-shot accuracy at each sparsity ratio, averaged across all benchmarks: Hellaswag, BoolQ, ARC (Challenge/Easy), MNLI, QNLI, RTE, OpenBookQA, Winogrande, and MathQA. The results of OptiPrune are highlighted in blue. **Bold** numbers indicate the highest performance among the methods.

9.3 Results

9.3.1 Zero-shot Performance Across Sparsity Levels

Table 9.1 reports zero-shot accuracy across various target sparsity levels. The proposed OptiPrune method consistently outperforms existing baselines

Method	ARC-C	ARC-E	BoolQ	HSwag	MathQA	MNLI	OBQA	QNLI	RTE	WGrande	AVG
Llama-2 (7B) Touvron2023LlamaModels											
Dense	43.43	76.30	77.71	57.16	28.17	42.24	31.40	49.90	62.82	69.14	53.83
OptiPrune	35.85	**65.28**	**73.56**	49.03	26.03	**40.37**	**27.50**	50.22	**58.57**	**66.17**	**49.26**
OWL	**36.09**	64.69	73.46	**49.05**	**26.31**	38.74	27.50	50.45	56.05	65.89	48.82
RIA	34.97	62.71	68.58	47.02	25.47	39.96	25.98	50.28	57.85	64.12	47.69
SparseGPT	35.41	63.60	70.15	48.25	26.25	37.73	27.05	50.51	55.42	65.40	47.98
Wanda	34.79	60.64	66.44	45.86	25.98	37.57	26.05	**50.88**	56.72	63.52	46.85
Llama-2 (13B) Touvron2023LlamaModels											
Dense	48.46	79.38	80.61	60.07	32.06	43.19	35.00	49.53	65.34	72.38	56.60
OptiPrune	**40.00**	**69.76**	**78.01**	**52.42**	**28.60**	40.42	**29.98**	**49.57**	**59.39**	**69.36**	**51.75**
OWL	39.88	69.49	77.25	52.08	28.46	40.09	29.88	49.51	58.21	68.88	51.37
RIA	38.46	66.94	73.12	50.43	28.32	39.92	28.62	49.52	58.98	66.75	50.11
SparseGPT	39.25	67.96	75.65	51.14	28.25	**40.69**	29.12	49.54	57.85	68.01	50.75
Wanda	38.01	63.89	71.15	49.32	27.73	40.12	28.32	49.28	58.08	66.34	49.22
Llama-3 (8B) dubey2024llama3herdmodels$_{brief}$											
Dense	50.26	80.09	80.98	60.11	40.47	47.82	34.60	49.94	68.59	73.40	58.62
OptiPrune	**39.19**	**67.03**	**77.40**	**50.85**	32.62	**42.80**	**27.55**	51.15	**62.59**	**68.06**	**51.92**
OWL	38.76	66.74	76.74	50.52	**32.82**	42.43	27.88	51.81	62.73	67.77	51.82
RIA	37.20	62.97	69.30	47.76	31.70	41.35	26.00	50.23	61.42	66.06	49.40
SparseGPT	38.77	66.00	74.87	49.77	32.73	42.18	27.20	**51.48**	61.60	67.04	51.16
Wanda	37.08	61.93	71.67	47.60	31.45	40.99	26.82	49.96	59.39	65.73	49.26
OPT (6.7B) Zhang2022OPT:Models											
Dense	30.46	65.57	66.06	50.51	24.62	32.81	27.60	50.92	55.23	65.19	46.89
OptiPrune	27.56	59.54	**65.28**	44.97	**24.34**	**34.56**	**24.38**	50.31	**54.18**	61.70	**44.68**
OWL	27.46	59.45	65.25	44.80	23.99	34.23	24.30	50.20	53.97	61.82	44.55
RIA	26.77	55.81	65.01	42.89	23.18	32.86	22.98	50.16	54.15	60.56	43.44
SparseGPT	**27.59**	**60.05**	65.25	**45.25**	23.92	33.81	24.30	**50.32**	53.70	**62.21**	44.64
Wanda	26.43	52.28	60.70	40.79	22.84	32.62	22.50	50.22	53.20	59.31	42.09

Table 9.2: Zero-shot accuracy of all benchmarks, averaged over all target sparsity (from 10% to 80% sparsity). OptiPrune 's results are highlighted in blue. Dense models' results are highlighted in gray. **Bold** numbers indicate highest performance among methods.

at nearly all sparsity ratios. When averaged over all sparsity levels, OptiPrune achieves the highest overall accuracy among competing approaches. This trend remains consistent across multiple model architectures and parameter scales. Although OptiPrune does not always yield the top result at every individual sparsity level, its performance remains highly competitive and stable, demonstrating strong robustness across a wide sparsity range.

9.3.2 Zero-shot Performance Across Benchmarks

Table 9.2 presents zero-shot accuracy for individual benchmarks, averaged over all sparsity levels. Our method achieves superior performance on almost all benchmarks and attains the highest average score across tasks. This improvement is consistent across different model families, indicating that OptiPrune provides reliable generalization and robustness in zero-

shot settings. The gains across diverse benchmarks—spanning commonsense reasoning, NLI, and QA—underscore the method's adaptability and broad effectiveness.

9.3.3 Perplexity Evaluation

Table 9.4 summarizes the average perplexity results across all sparsity ratios. OptiPrune attains the lowest perplexity among all baselines for the Llama series of models. For the OPT architecture, while OptiPrune does not surpass OWL [84], it still performs comparably to other state-of-the-art (SOTA) methods, maintaining competitive perplexity values overall. These results confirm the general effectiveness of the proposed method across model types and sizes.

9.3.4 Additional Results

We further evaluate OptiPrune under semi-structured pruning with an $N:M$ constraint, where at least N of every M consecutive weights are set to zero, enabling structured sparsity with improved hardware efficiency. As shown in Table 9.5, OptiPrune achieves superior performance in most architectures under the $2:4$ semi-structured setting, demonstrating its flexibility and compatibility with hardware-friendly pruning schemes.

Figure 9.1 compares the perplexity differences between models pruned using uniform and non-uniform sparsity patterns under the RIA framework. The results reveal that non-uniform sparsity achieves lower perplexity at higher sparsity ratios, suggesting its effectiveness in retaining important weights under extreme compression. In contrast, at lower sparsity ratios, uniform sparsity tends to yield slightly better performance. This trend remains consistent across all evaluated architectures.

	Sparsity ratio								
λ Calculation	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	Avg
Linear translate	**53.83**	54.06	53.61	**53.39**	51.50	**48.05**	**42.83**	36.79	**49.26**
Sigmoid function	53.78	**54.20**	**53.70**	53.15	**50.56**	47.92	42.82	**36.85**	49.12

Table 9.3: Zero-shot accuracy across different sparsity ratios for Llama2-7B.

9.3.5 Effect of Weight Reconstruction in RIA

To further analyze RIA's behavior, we examine the impact of integrating SparseGPT's weight reconstruction procedure using RIA's pruning mask.

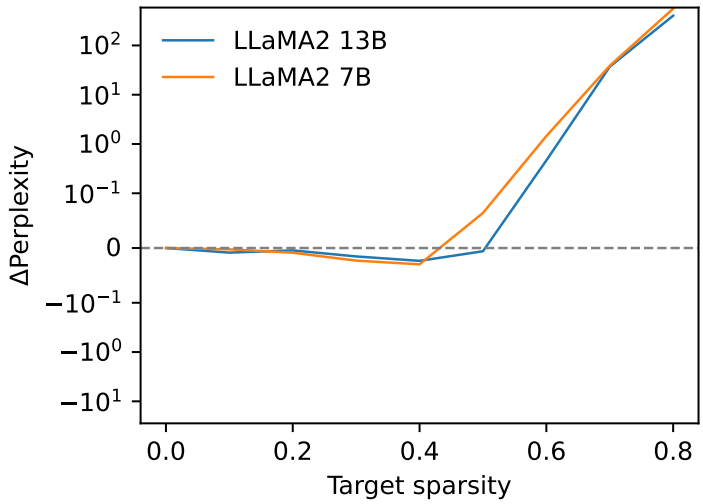


Figure 9.1: Perplexity difference (Δ Perplexity) between uniform and non-uniform sparsity using RIA. Lower perplexity indicates better performance. Negative Δ denotes cases where uniform sparsity performs better, whereas positive Δ indicates superior performance of non-uniform sparsity.

Figure 9.2 presents the perplexity differences between models pruned with and without reconstruction across various sparsity levels. The results suggest that weight reconstruction tends to reduce performance when the sparsity ratio is low but provides noticeable gains at higher sparsity levels, where model capacity is more constrained.

Method	Llama2-7B	Llama2-13B	Llama3-8B	OPT-6.7B
OptiPrune	**18.98**	**13.68**	**39.42**	26.55
RIA	152.88	65.15	310.23	1967.77
SparseGPT	23.18	22.61	44.56	26.35
OWL	20.61	14.81	39.67	**24.49**
Wanda	743.84	251.90	447.14	592.75

Table 9.4: Average perplexity across all sparsity levels. Lower values indicate better performance.

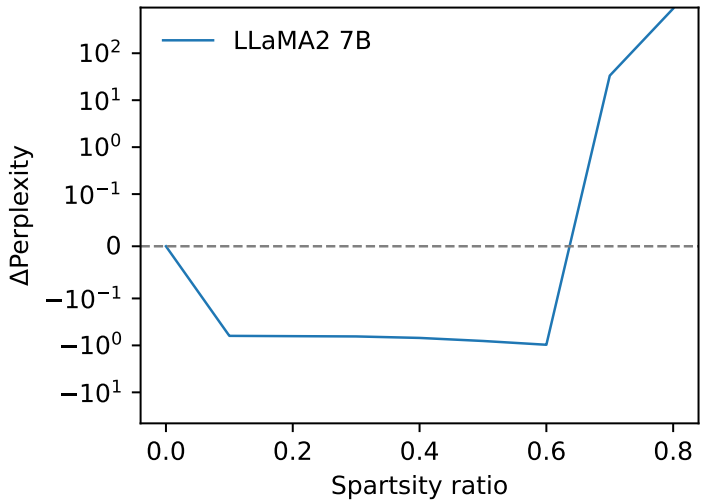


Figure 9.2: Perplexity difference (Δ Perplexity) between RIA with and without weight reconstruction. Negative Δ values indicate that reconstruction degrades performance, while positive Δ values indicate improvements.

Method	Llama2-7B	Llama2-13B	Llama3-8B	OPT-6.7B
OptiPrune	**12.31**	9.86	**18.49**	**16.09**
RIA	12.75	**9.49**	25.81	17.63
SparseGPT	12.41	9.87	18.49	16.11
OWL	12.41	9.86	18.49	16.11
Wanda	13.72	10.17	27.71	17.80

Table 9.5: Average perplexity under semi-structured $2:4$ pruning. Lower values indicate better performance.

9.3.6 Retrieval-based Speculative Decoding

Table 9.6 presents the decoding performance of LLaMA2-7B-Chat on the GSM8K dataset [12], measured in terms of speedup relative to standard autoregressive decoding. The baseline autoregressive method achieves a reference speedup of 1.0x. REST [25], a previous retrieval-based speculative decoding approach, provides a marginal improvement, achieving a 1.01x speedup. In contrast, our retrieval-based speculative decoding method demonstrates a substantial acceleration, reaching a speedup of **1.18x**. Furthermore, our framework is explicitly designed to allow seamless integration with complementary acceleration techniques, such as the N-gram-based

Method	Speedup	AVG Accepted Length
Autoregressive	1.00x	1.00
REST	1.01x	1.47
REST (*Integrate*)	1.08x	1.47
Ours	1.18x	1.31
Ours (*Integrate*)	2.14x	2.64

Table 9.6: Decoding performance (speedup gain) of LLaMA2-7B-Chat, measured on GSM8K dataset [12]. (*Integrate*) indicate the integration with N-gram-based approach[19].

method [19], whereas REST does not support such integration due to its corpus size. When combined with the N-gram strategy, our method achieves a speedup of **2.14x**, representing an almost twofold improvement over standard decoding. This result highlights that by selectively retrieving high-quality, model-aligned text sequences and integrating them as speculative drafts, our approach significantly enhances decoding efficiency while maintaining the fidelity of generated outputs. The notable speedup indicates that our method effectively reduces the number of forward passes required during speculative decoding, outperforming prior retrieval-based techniques.

Chapter 10

Conclusions

10.1 Conclusions

In this thesis, we have addressed key challenges in developing high-quality, reliable, and deployment-efficient Large Language Models (LLMs). Our work focuses on two complementary directions: improving answer quality and confidence estimation, and enabling optimal model pruning across varying sparsity levels.

First, we proposed ANSPRE, a structured answer generation framework that guides LLMs to produce concise, structured answers with reliable confidence scores. By enabling aggregation across multiple retrievals and reasoning paths, ANSPRE improves both accuracy and robustness. We extended this framework to multilingual and vision-language settings through SELF-ANSPRE, demonstrating significant gains on Vietnamese and Japanese QA benchmarks, as well as visual QA tasks. Extensive experiments show that ANSPRE consistently improves Exact Match (EM) and F1 scores, while producing better-calibrated confidence estimates suitable for high-stakes applications.

Second, we conducted an empirical study of pruning techniques, revealing two key findings: (i) layerwise uniform sparsity performs best at low sparsity, while non-uniform sparsity excels at high sparsity, and (ii) relative importance-based mask pruning is more effective at low sparsity, whereas Hessian-based weight reconstruction is superior at high sparsity. Building on these insights, we developed OPTIPRUNE, a pruning method that dynamically selects the optimal sparsity distribution and solver for each layer. Our experiments demonstrate that OPTIPRUNE outperforms existing state-of-the-art pruning methods across a variety of LLM architectures, sparsity levels, benchmarks, and language calibrations.

Overall, this thesis contributes to the design of LLMs that are simultaneously accurate, reliable, and deployment-friendly. Future work includes exploring adaptive pruning strategies for continual learning, extending ANSPRE to more complex reasoning tasks, and investigating hardware-aware compression techniques to further improve efficiency in practical deployments.

10.2 Published Works

- [Q1 Journal] [2nd Revision] **Le, Nguyen-Khang**, Nguyen, D. H., & Nguyen, L. M. "Answer-Prefix Prompting for High-Quality and Confidence-Aware Generation in Large Language and Vision-Language Models." *Information Processing & Management* (Reviewing).

- [Q1 Journal] Nguyen, D. H., **Le, Nguyen-Khang**, & Nguyen, L. M. "ViWiQA: Efficient end-to-end Vietnamese Wikipedia-based Open-domain Question-Answering systems for single-hop and multi-hop questions." *Information Processing & Management*, Vol. 60, Issue 6, 2023, Article 103514. `https://doi.org/10.1016/j.ipm.2023.103514`

- [A* Conference] [SAC Highlights Award (top 1%)] **Le, Nguyen-Khang**, Truong Do, & Nguyen, Le Minh. "SPECTRA: Faster Large Language Model Inference with Optimized Internal and External Speculation." ***ACL** 2025*.

- [A* Conference] **Le, Nguyen-Khang**, Truong Do, & Nguyen, Le Minh. "AdaSpec: Adaptive Multilingual Speculative Decoding with Self-Synthesized Language-Aware Training and Vocabulary Simplification." ***AAAI** 2026*.

- [A Conference] **Le, Nguyen-Khang**, Nguyen, D. H., & Nguyen, L. M. "ANSPRE: Improving Question-Answering in Large Language Models with Answer-Prefix Generation." *Frontiers in Artificial Intelligence and Applications*, Vol. 392 (**ECAI** 2024), pp. 2500–2507, 2024. `https://doi.org/10.3233/FAIA240778`

- [Prestigious B] **Khang Nguyen Le**, Ryo Sato, Dai Nakashima, Takeshi Suzuki, & Minh Le Nguyen. "OptiPrune: Effective Pruning Approach for Every Target Sparsity." In *Proceedings of the 31st International Conference on Computational Linguistics (**COLING** 2025)*, pp. 3600–3612, Abu Dhabi, UAE. Association for Computational Linguistics.

- [**Prestigious B**] Dieu-Hien Nguyen, **Nguyen-Khang Le**, Truong Do, and Le-Minh Nguyen. 2025. "LangCompress: Language-Aware Compression of Large Language Models" *Proceedings of the **AACL** Conference.*

- [**B Conference**] **Le, Nguyen-Khang**, Nguyen, D. H., & Nguyen, L. M. "Integrating Vision-Tool to Enhance Visual-Question-Answering in Special Domains." In: Hadfi, R., Anthony, P., Sharma, A., Ito, T., Bai, Q. (eds), *PRICAI 2024: Trends in Artificial Intelligence*, Lecture Notes in Computer Science, Vol. 15283, Springer, Singapore, pp. 158–169, 2025. `https://doi.org/10.1007/978-981-96-0122-6_15`

- [**B Conference**] Nguyen, D. H., **Le, Nguyen-Khang**, & Nguyen, L. M. "Multi-target Contrastive Objective for Learning Property-Aware Vision-Language Representation." In: Wu, S., Su, X., Xu, X., Kang, B.H. (eds), *PKAW 2024*, Lecture Notes in Computer Science, Vol. 15372, Springer, Singapore, pp. 164–175, 2025. `https://doi.org/10.1007/978-981-96-0026-7_13`

- [**JSAI Annual Award**] **Nguyen-Khang Le**, Quan Minh Bui, Minh Ngoc Nguyen, Hiep Nguyen, Trung Vo, Son T. Luu, Shoshin Nomura, & Minh Le Nguyen. "Automated Web Application Testing: End-to-End Test Case Generation with Large Language Models and Screen Transition Graphs." (JSAI 2025).

- **Le, Nguyen-Khang**, Nguyen, D. H., Do, D. T., Nguyen, C., & Nguyen, M. L. "Vietnamese Elementary Math Reasoning Using Large Language Model with Refined Translation and Dense-Retrieved Chain-of-Thought." In: Suzumura, T., Bono, M. (eds), *New Frontiers in Artificial Intelligence, JSAI-isAI 2024*, Lecture Notes in Computer Science, Vol. 14741, Springer, Singapore, pp. 260–268, 2024. `https://doi.org/10.1007/978-981-97-3076-6_18`

- Hai Nguyen, Hiep Nguyen, Trang Pham, Minh Nguyen, An Trieu, Dinh-Truong Do, **Nguyen-Khang Le**, Le-Minh Nguyen "JNLP at COLIEE 2025: Hybrid Large Language Model-based Framework for Legal Information Retrieval and Entailment", COLIEE 2025 in association with the 20th International Conference on Artificial Intelligence and Law

- Nguyen, D. H., **Le, Nguyen-Khang**, & Nguyen, M. L. "AuthNet: A Framework for Research Expert Discovery and Network Visualization Based on Topic-Specific Queries." In: Nakano, Y., Suzumura, T.

(eds), *JSAI-isAI 2025*, Lecture Notes in Computer Science, Vol. 15692, Springer, Singapore, pp. 196–209, 2025. https://doi.org/10.1007/978-981-96-7071-0_13

- Chau Nguyen, Thanh Tran, Son T. Luu, **Nguyen-Khang Le**, Dinh-Truong Do, Shintaro Kawamura, Shoichi Naito, Yuki Mogi, & Le-Minh Nguyen. "Fostering Business Innovation with AI: Performance of Fine-Tuned Japanese Language Models under Resource Restrictions." In *AI-Biz 2024 proceedings*, 2023.

- Bui, M. Q., Do, D. T., **Le, Nguyen-Khang**, Nguyen, D. H., Nguyen, K. V. H., Anh, T. P. N., & Nguyen, M. L. "Data Augmentation and Large Language Model for Legal Case Retrieval and Entailment." *Review of Socionetwork Strategies*, Vol. 18, pp. 49–74, 2024. https://doi.org/10.1007/s12626-024-00158-2

- Bui, Q. M., Do, D. T., **Le, Nguyen-Khang**, D. H., Nguyen, K. V. H., Anh, T. P. N., & Nguyen, M. L. "JNLP COLIEE-2023: Data Argumentation and Large Language Model for Legal Case Retrieval and Entailment." In: Workshop of the Tenth Competition on Legal Information Extraction/Entailment (COLIEE'2023), held at the 19th International Conference on Artificial Intelligence and Law (ICAIL), June 2023.

- Nguyen, C., Luu, S., Tran, T., Trieu, A., Dang, A., Nguyen, D., Nguyen, H., Pham, T., Pham, T., Vo, T. T., Dol, D. T., **Le, Nguyen-Khang**, Nguyen, D. H., Le, N. C., Le, T. T., Bui, Q., Nguyen, P., Nguyen, H. T., Tran, V., & Nguyen, L. M. "A Summary of the ALQAC 2023 Competition." In *15th International Conference on Knowledge and Systems Engineering (KSE)*, pp. 1–6, 2023.

- Dinh-Truong Do, Son T. Luu, Trang Pham, Trung Vo, Nguyen-Hoang Chu, Quang-Huy Chu, Cuong Nguyen, Minh Nguyen, An Trieu, Dat Nguyen, Thanh Tran, Cong Nguyen, Hiep Nguyen, Chau Nguyen, **Nguyen-Khang Le**, Dieu-Hien Nguyen, Binh Dang, Phuong Nguyen, Ha-Thanh Nguyen, Vu Tran, Le-Minh Nguyen. *A Summary of the ALQAC 2024 Competition. Proceedings of the 16th International Conference on Knowledge and System Engineering (KSE 2024)*, 2024.

Bibliography

[1] Vaibhav Adlakha, Parishad BehnamGhader, Xing Han Lu, Nicholas Meade, and Siva Reddy. Evaluating correctness and faithfulness of instruction-following models for question answering. *Transactions of the Association for Computational Linguistics*, 12:681–699, 2024.

[2] AI@Meta. Llama 3 model card. 2024.

[3] Aida Amini, Saadia Gabriel, Earl Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2357–2367, 2019.

[4] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. In *The Twelfth International Conference on Learning Representations*, 2024.

[5] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen Technical Report, 2023.

[6] Akash Wamanrao Bhiwgade, Nilesh Nagrale, Pragati Patil Bedekar, and Sayara Bano Sheikh. Integrating open-source llms with retrieval-augmented generation for obstetrics and gynecology domain. In *2025*

IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS), pages 1–5, 2025.

[7] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198, 2024.

[8] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, Ion Stoica, and Eric P Xing. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality, 3 2023.

[9] Peter Christen. *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Springer Publishing Company, Incorporated, 2012.

[10] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, 2019.

[11] Peter Clark, Liam Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

[12] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training Verifiers to Solve Math Word Problems, 2021. _eprint: 2110.14168.

[13] Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pages 177–190, 2005.

[14] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, and et al. The llama 3 herd of models, 2024.

[15] Ahmed K Elmagarmid, Panagiotis G Ipeirotis, and Vassilios S Verykios. Duplicate Record Detection: A Survey. *IEEE Trans. on Knowl. and Data Eng.*, 19(1):1–16, 1 2007.

[16] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019.

[17] Elias Frantar and Dan Alistarh. Spdy: Accurate pruning with speedup guarantees. In *International Conference on Machine Learning*, pages 6726–6743. PMLR, 2022.

[18] Elias Frantar and Dan Alistarh. SparseGPT: massive language models can be accurately pruned in one-shot. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[19] Yichao Fu, Peter Bailis, Ion Stoica, and Hao Zhang. Break the sequential dependency of LLM inference using LOOKAHEAD DECODING. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024. Place: Vienna, Austria.

[20] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 1321–1330. JMLR.org, 2017.

[21] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval Augmented Language Model Pre-Training. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR, 10 2020.

[22] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *Advances in neural information processing systems*, volume 28, 2015.

[23] Ching Nam Hang, Pei-Duo Yu, and Chee Wei Tan. Trumorgpt: Graph-based retrieval-augmented large language model for fact-checking. *IEEE Transactions on Artificial Intelligence*, pages 1–15, 2025.

[24] Babak Hassibi, David G Stork, and Gregory J Wolff. Optimal brain surgeon and general network pruning. In *IEEE international conference on neural networks*, pages 293–299. IEEE, 1993.

[25] Zhenyu He, Zexuan Zhong, Tianle Cai, Jason Lee, and Di He. REST: Retrieval-Based Speculative Decoding. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1582–1595, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[26] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The Curious Case of Neural Text Degeneration. In *International Conference on Learning Representations*, 2020.

[27] Yinghao Hu, Leilei Gan, Wenyi Xiao, Kun Kuang, and Fei Wu. Fine-tuning large language models for improving factuality in legal question answering. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert, editors, *Proceedings of the 31st International Conference on Computational Linguistics*, pages 4410–4427, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics.

[28] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised Dense Information Retrieval with Contrastive Learning. *Transactions on Machine Learning Research*, 2022.

[29] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B, 2023.

[30] Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. How Can We Know When Language Models Know? On the Calibration of Language Models for Question Answering. *Transactions of the Association for Computational Linguistics*, 9:962–977, 2021.

[31] Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active Retrieval Augmented Generation. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore, 12 2023. Association for Computational Linguistics.

[32] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada, 7 2017. Association for Computational Linguistics.

[33] Vladimir Karpukhin, Barlas Oguz, Sewon Min, and et al. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[34] Wouter Kool, Herke van Hoof, and Max Welling. Ancestral Gumbel-Top-k Sampling for Sampling Without Replacement. *Journal of Machine Learning Research*, 21(47):1–36, 2020.

[35] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019.

[36] Yanis Labrak, Adrien Bazoge, Emmanuel Morin, Pierre-Antoine Gourraud, Mickael Rouvier, and Richard Dufour. Biomistral: A collection of open-source pretrained large language models for medical domains, 2024.

[37] Khang Le, Hien Nguyen, Tung Le Thanh, and Minh Nguyen. VIMQA: A Vietnamese Dataset for Advanced Reasoning and Explainable Multihop Question Answering. In *Proceedings of the Language Resources and Evaluation Conference*, pages 6521–6529, Marseille, France, 6 2022. European Language Resources Association.

[38] Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. In *Advances in neural information processing systems*, volume 2, 1989.

[39] Jaehong Lee, Sejung Park, Seul-Kee Mo, Sungsoo Ahn, and Jinwoo Shin. Layer-adaptive sparsity for the magnitude-based pruning. *arXiv preprint arXiv:2010.07611*, 2020.

[40] Jon Lee, Tom Kwiatkowski, Marcin Chrzanowski, and et al. Open retrieval question answering. In *Proceedings of the 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

[41] Namhoon Lee, Thalaiyasingam Ajanthan, and Philip Torr. Snip: Single-shot network pruning based on connection sensitivity. In *International Conference on Learning Representations*, 2019.

[42] Patrick Lewis, Ethan Perez, Aleksandra Piktus, and et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *NeurIPS 2020*, 2020.

[43] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[44] Dong Li, Jintao Tang, Pancheng Wang, Shasha Li, and Ting Wang. Maximizing discrimination masking for faithful question answering with machine reading. *Information Processing Management*, 62(1):103915, 2025.

[45] Jiajun Li and Ahmed Louri. Adaprune: An accelerator-aware pruning technique for sustainable cnn accelerators. volume 7, pages 47–60, 2021.

[46] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23, Honolulu, Hawaii, USA, 2023. JMLR.org.

[47] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023.

[48] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023.

[49] Shiwei Liu, Tianlong Chen, Xiaohan Chen, Zahra Atashgahi, Lu Yin, Huanyu Kou, Li Shen, Mykola Pechenizkiy, Zhangyang Wang, and Decebal Constantin Mocanu. Sparse training via boosting pruning plastic-

ity with neuroregeneration. *Advances in Neural Information Processing Systems*, 34:9908–9922, 2021.

[50] Hongyin Luo, Tianhua Zhang, Yung-Sung Chuang, Yuan Gong, Yoon Kim, Xixin Wu, Helen Meng, and James Glass. Search Augmented Instruction Learning. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3717–3729, Singapore, 12 2023. Association for Computational Linguistics.

[51] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When Not to Trust Language Models: Investigating Effectiveness of Parametric and Non-Parametric Memories. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada, 7 2023. Association for Computational Linguistics.

[52] Minesh Mathew, Viraj Bagal, Rubèn Pérez Tito, Dimosthenis Karatzas, Ernest Valveny, and C. V Jawahar. Infographicvqa, 2021.

[53] Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. Docvqa: A dataset for vqa on document images, 2021.

[54] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

[55] Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Zhengxin Zhang, Rae Ying Yee Wong, Alan Zhu, Lijie Yang, Xiaoxiang Shi, Chunan Shi, Zhuoming Chen, Daiyaan Arfeen, Reyna Abhyankar, and Zhihao Jia. SpecInfer: Accelerating Large Language Model Serving with Tree-based Speculative Inference and Verification. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*, ASPLOS '24, pages 932–949, New York, NY, USA, 2024. Association for Computing Machinery. event-place: La Jolla, CA, USA.

[56] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, 2018.

[57] Sewon Min, Patrick Lewis, Wen-tau Yih, and Luke Zettlemoyer. Hard em for open-domain question answering. In *Proceedings of the*

2019 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2019.

[58] Sewon Min, Wen-tau Yih, and Graham Neubig. Graphretriever: Exploiting keyword graphs for document retrieval in open-domain qa. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

[59] Decebal Constantin Mocanu, Elena Mocanu, Peter Stone, Phuong H Nguyen, Madeleine Gibescu, and Antonio Liotta. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature communications*, 9(1):2383, 2018.

[60] Dieu-Hien Nguyen, Nguyen-Khang Le, and Le-Minh Nguyen. Viwiqa: Efficient end-to-end vietnamese wikipedia-based open-domain question-answering systems for single-hop and multi-hop questions. *Information Processing Management*, 60(6):103514, 2023.

[61] Kiet Nguyen, Vu Nguyen, Anh Nguyen, and Ngan Nguyen. A Vietnamese dataset for evaluating machine reading comprehension. In Donia Scott, Nuria Bel, and Chengqing Zong, editors, *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2595–2605, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.

[62] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S Koyejo, S Mohamed, A Agarwal, D Belgrave, K Cho, and A Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc., 2022.

[63] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. 2019.

[64] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8732–8740, 2020.

[65] Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language Models Can Teach Themselves to Use Tools. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[66] ByungHoon So, Kyuhong Byun, Kyungwon Kang, and Seongjin Cho. JaQuAD: Japanese Question Answering Dataset for Machine Reading Comprehension, 2022.

[67] Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. A Simple and Effective Pruning Approach for Large Language Models. In *The Twelfth International Conference on Learning Representations*, 2024.

[68] Shiqi Sun, Kun Zhang, Jingyuan Li, Min Yu, Kun Hou, Yuanzhuo Wang, and Xueqi Cheng. Retriever-generator-verification: A novel approach to enhancing factual coherence in open-domain question answering. *Information Processing Management*, 62(4):104147, 2025.

[69] Yanhan Sun, Jiangtao Qi, Zhenfang Zhu, Kefeng Li, Liang Zhao, and Lei Lv. Bias-guided margin loss for robust visual question answering. *Information Processing Management*, 62(2):103988, 2025.

[70] Masatoshi Suzuki, Jun Suzuki, Koji Matsuda, Kyosuke Nishida, and Naoya Inoue. Jaqket: Kuizu o daizai ni shita nihongo qa dētasetto no kōchiku [jaqket: Construction of a japanese qa dataset on the subject of quizzes]. In *The 26th Annual Conference of the Association for Natural Language Processing*, 2020.

[71] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie

Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu-hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. Gemma: Open Models Based on Gemini Research and Technology, 2024.

[72] Qwen Team. Qwen2.5: A party of foundation models, September 2024.

[73] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open Foundation and Fine-Tuned Chat Models, 2023.

[74] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann,

Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[75] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, 2018.

[76] Chaoqi Wang, Guodong Zhang, and Roger Grosse. Picking winning tickets before training by preserving gradient flow. In *International Conference on Learning Representations*, 2020.

[77] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[78] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations*, 2023.

[79] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned Language Models are Zero-Shot Learners. In *International Conference on Learning Representations*, 2022.

[80] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, 2023.

[81] Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, 2018.

[82] Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, Fan Yang, Fei Deng, Feng Wang, Feng Liu, Guangwei Ai, Guosheng Dong, Haizhou Zhao, Hang Xu, Haoze Sun, Hongda Zhang, Hui Liu, Jiaming Ji, Jian Xie, JunTao Dai, Kun Fang, Lei Su, Liang Song, Lifeng Liu, Liyun Ru, Luyao Ma, Mang Wang, Mickel Liu, MingAn Lin, Nuolan Nie, Peidong Guo, Ruiyang Sun, Tao Zhang, Tianpeng Li, Tianyu Li, Wei Cheng, Weipeng Chen, Xiangrong Zeng, Xiaochuan Wang, Xiaoxi Chen, Xin Men, Xin Yu, Xuehai Pan, Yanjun Shen, Yiding Wang, Yiyu Li, Youxin Jiang, Yuchen Gao, Yupeng Zhang, Zenan Zhou, and Zhiying Wu. Baichuan 2: Open Large-scale Language Models, 2023.

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

[84] Lu Yin, You Wu, Zhenyu Zhang, Cheng-Yu Hsieh, Yaqing Wang, Yiling Jia, Mykola Pechenizkiy, Yi Liang, Zhangyang Wang, and Shiwei Liu. Outlier Weighed Layerwise Sparsity (OWL): A Missing Secret Sauce for Pruning LLMs to High Sparsity, 2024.

[85] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, 2019.

[86] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria

Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. OPT: Open Pre-trained Transformer Language Models, 2022.

[87] Yingtao Zhang, Haoli Bai, Haokun Lin, Jialin Zhao, Lu Hou, and Carlo Vittorio Cannistraci. Plug-and-Play: An Efficient Post-training Pruning Method for Large Language Models. In *The Twelfth International Conference on Learning Representations*, 2024.

[88] Michael Zhu and Suyog Gupta. To prune, or not to prune: exploring the efficacy of pruning for model compression, 2017.