UNIVERSITY OF COPENHAGEN

This thesis has been submitted to the PhD School of The Faculty of Science, University of Copenhagen

Reasoning Inconsistencies and How to Mitigate Them in Deep Learning

Erik Arakelyan

Supervised by Isabelle Augenstein and Pasquale Minervini

Submission date: 31st December 2024

"This thesis is dedicated to the cherished memory of Professor Norair Arakelian - a brilliant mathematician and an even more extraordinary grandfather whose legacy echoes lovingly through the corridors of time."

Acknowledgements

As I am writing this segment, the pouring words inadvertently remind me of the past three years full of seemingly insurmountable perils and great triumphs. This journey of self-discovery would not have been possible without the support of many people who will always have a warm spot in my heart. These are the people who have shown me guidance and tough and tender love through their critiques and reassurances.

First of all, I would like to thank both of my excellent supervisors, who have been there with me and for me each step of the way. I want to express my gratitude to Isabelle for believing in me, providing me with this incredible journey and this life-changing experience, and mentoring me on the true way of (ninja) being a scientist. Her pursuit of excellence and outstanding scientific narratives is something I took to heart. I want to dearly thank Pasquale for all the time and effort he spent helping me in all things professional and personal; without those late-night guidance sessions and talks, PhD would have felt tougher than bringing the one ring to Mordor. I am deeply indebted that he introduced me to the world of academic research during my MSc days leading me to where I am today.

This journey would not have been possible without my amazing collaborations - Daniel Daza and Michael Cochez, who have been a blessing to work with, constantly inspiring me for greater scientific discoveries and curious undertakings.

I also want to thank Zhaoqi Liu and Gayane Ghazaryan, whom I was lucky to mentor during their time at the university. It was very inspiring to see such bright minds at the start of their careers but showing maturity in their research.

I definitely want to express my gratitude to all of my friends I met at the CopeNLU lab and for our fun times together - Marta, Sara, Haeun, Nadav, Jingyi, Siddhesh, Dustin and Greta. A profound thanks goes to Arnav and Pepa, with whom we had a fantastic time talking and laughing about all sorts of things meaningful, funny or mundane in the office. Thanks for all the support.

A very special corner of this acknowledgement goes to my dear friends Karolina and Karen, with whom we share countless hours of deep scientific discussions and heartwarming, friendly banter. I will always cherish these times with a nostalgic smile on my face.

This journey would not have been possible without my amazing family's complete support. Words cannot express my deep gratitude for the unconditional love, understanding, support and encouragement that I have received from my wife and soulmate, Tamara and my mother, Araksya. That gratitude also goes to my grandmother Marietta, grandfather Norair, my aunt Anahit, her husband Armen, and my brothers Vahan and David, who have been the inspiration for me to pursue this endeavour.

Abstract

The recent advancements in Deep Learning (DL) models and techniques have led to significant strides in performance across diverse tasks and modalities. However, while the overall capabilities of models show promising growth, our understanding of their internal reasoning processes remains limited, particularly concerning systematic inconsistencies or errors—patterns of logical or inferential flaws. These inconsistencies may manifest as contradictory outputs, failure to generalize across similar tasks, or erroneous conclusions in specific contexts. Even detecting and measuring such reasoning discrepancies is challenging, as they may arise from opaque internal procedures, biases and imbalances in training data, or the inherent complexity of the task. Without effective methods to detect, measure, and mitigate these errors, there is a significant risk of deploying models that are biased, exploitable, or logically unreliable.

This thesis aims to address these issues by producing novel methods for deep learning models that reason over knowledge graphs, natural language, and images. The first part of the thesis contributes two techniques for detecting and explicitly quantifying predictive inconsistencies originating from opaque internal procedures in natural language and image processing models. We systematically evaluate a wide range of model families within novel adversarial setups that explicitly expose those internal procedures, allowing us to quantify significant reasoning discrepancies within these models. To mitigate inconsistencies from biases in training data, this thesis presents a data-efficient sampling method to improve fairness and performance and a synthetic dataset generation approach to rigorously evaluate and enhance reasoning in low-resource scenarios. Finally, the thesis offers two novel techniques to explicitly optimize the models for complex reasoning tasks in natural language and knowledge graphs. These methods directly enhance model performance while allowing for more faithful and interpretable exploration and exploitation during inference. Critically, by addressing reasoning inconsistencies through quantifying and mitigating them with deep learning models, this thesis provides a comprehensive framework to improve the robustness, fairness, and interpretability of deep learning models across diverse tasks and modalities.

Resumé

De seneste fremskridt inden for dyb læring (DL) modeller og teknikker har ført til en betydelig forbedring af ydeevnen på tværs af forskellige opgaver og modaliteter. Imidlertid, mens modellernes overordnede kapacitet viser lovende vækst, vores forståelse af deres interne tankevirksomheder forbliver begrænset, især med hensyn til systematiske uoverensstemmelser eller fejl — mønstre af logiske eller inferentielle mangler. Disse uoverensstemmelser kan manifestere sig som modstridende udgange, manglende generalisering på tværs af lignende opgaver eller fejlagtige konklusioner i specifikke sammenhænge. Selv det er en udfordring at opdage og måle sådanne tankeforskelle, da de kan opstå som følge af uigennemsigtige interne procedurer, skævheder og ubalancer i træningsdata eller fordi denne opgave er meget kompleks. Uden effektive metoder til at opdage, måle og afbøde disse fejl er der en betydelig risiko for at implementere modeller, der er partiske, udnyttelige eller logisk upålidelige.

Denne afhandling har til formål at løse disse problemer ved at producere nye metoder til dybe læringsmodeller, der ræsonnerer over videngrafer, naturligt sprog og billeder. Den første del af afhandlingen bidrager med to teknikker til at detektere og eksplicit kvantificere forudsigelige uoverensstemmelser, der stammer fra uigennemsigtige interne procedurer i naturlige sprog- og billedbehandlingsmodeller. Vi evaluerer systematisk en bred vifte af modelfamilier inden for nye kontradiktoriske opsætninger, der eksplicit udsætter disse interne procedurer, giver os mulighed for at kvantificere betydelige tankeforskelle inden for disse modeller. For at afbøde uoverensstemmelser fra fordomme i træningsdata, denne afhandling præsenterer en dataeffektiv prøveudtagningsmetode til forbedring af retfærdighed og ydeevne og en syntetisk datasætgenereringsmetode til nøje at evaluere og forbedre ræsonnement i scenarier med lav ressource. Endelig tilbyder afhandlingen to nye teknikker til eksplicit at optimere modellerne til komplekse tankeopgaver i naturlige sprog- og vidensgrafer. Disse metoder forbedrer direkte modelens ydeevne, samtidig med at de giver mulighed for mere trofast og fortolkbar udforskning og udnyttelse under inferens. Kritisk, ved at adressere tankeforskelle gennem kvantificering og afbødning af dem med dybe læringsmodeller, denne afhandling giver en omfattende ramme for

at forbedre robustheden, retfærdighed, og fortolkbarhed af dybe læringsmodeller på tværs af forskellige opgaver og modaliteter.

Publications

This thesis includes the following papers as chapters, listed in the order of their appearance (* denotes equal contribution):

1. (Arakelyan et al., 2024b) Erik Arakelyan*, Zhaoqi Liu* and Isabelle Augenstein, *Semantic Sensitivities and Inconsistent Predictions: Measuring the Fragility of NLI Models*, 2024, European Chapter of the Association for Computational Linguistics (EACL), *outstanding paper award*, pages 432-444

2. (Arakelyan et al., 2024a) Erik Arakelyan, Karen Hambardzumyan, Davit Papikyan, Pasquale Minervini, Aram H. Markosyan, Albert Gordo and Isabelle Augenstein, *With Great Backbones Comes Great Adversarial Transferability*, 2024, CoRR (Under Review for ICML 2025)

3. (Arakelyan et al., 2023a) Erik Arakelyan, Arnav Arora and Isabelle Augenstein, 2023, *Topic-Guided Sampling For Data-Efficient Multi-Domain Stance Detection*, Annual Meeting of the Association for Computational Linguistics (ACL), pages 13448-13464

4. (Ghazaryan et al., 2024) Gayane Ghazaryan*, Erik Arakelyan*, Pasquale Minervini and Isabelle Augenstein, *SynDARin: Synthesising Datasets for Automated Reasoning in Low-Resource Languages*, 2024, Proceedings of the 31st International Conference on Computational Linguistics (COLING)

5. (Arakelyan et al., 2023b) Erik Arakelyan*, Pasquale Minervini*, Daniel Daza, Michael Cochez and Isabelle Augenstein, *Adapting Neural Link Predictors for Data-Efficient Complex Query Answering*, 2023, Advances in Neural Information Processing Systems 36 (NeurIPS)

6. (Arakelyan et al., 2024c) Erik Arakelyan, Pasquale Minervini, Pat Verga, Patrick S. H. Lewis and Isabelle Augenstein, *FLARE: Faithful Logic-Aided Reasoning and Exploration*, 2024, CoRR (Under Review for ICLR 2025)

7. (Cochez et al., 2023) Michael Cochez, Dimitrios Alivanistos, Erik Arakelyan, Max Berrendorf, Daniel Daza, Mikhail Galkin, Pasquale Minervini, Mathias Niepert and Hongyu Ren, *Approximate Answering of Graph Queries*, 2023, Compendium of Neurosymbolic Artificial Intelligence, pages 373-386

The survey paper 7 is presented through definitions and problem formulations across different parts of the thesis, but not as a separate chapter.

Contents

List of Figures

List of Tables

Part I

Executive Summary

Executive Summary | 1

1.1 Introduction

The emergence of data-driven learning and predictive approaches (Carbonell et al., 1983) has unequivocally led to the question "*How do machines reason?*". In the spirit of this, earlier machine learning methods (Anderson, 1983) were constructed with ingrained mechanisms that directly explain the complete sequence for arriving at the solution (Bratko, 1997). The transparency of the reasoning procedure allowed for steering away from potential biases (He and Garcia, 2009) introduced through data imbalances and the evaluation of the complexity of the designated task in terms of the expressivity (Vapnik, 1999) of the model, as well as the sufficiency (Balasubramanian et al., 2014; Vapnik, 1999) of the learning methodology. With the advancement of computing resources (LeCun, 2019; Sze et al., 2017), deep learning models have made considerable strides in pushing state-of-the-art performance in reasoning over natural language (Wang et al., 2019b; et al., 2023), images (Russakovsky et al., 2015; Lin et al., 2014) and knowledge graphs (Chen et al., 2020d). The most recent advancements have been propelled by developing large models pre-trained with vast amounts of data (Vaswani et al., 2017a; Dosovitskiy et al., 2021). However, not all that glitters is gold. The added structural and algorithmic learning complexity within these models (Devlin et al., 2019; Radford et al., 2018) has significantly limited the potential to tractably interpret (Bender et al., 2021) or follow the reasoning processes within them (Atanasova et al., 2020). Consequently, a large portion of modern explainability methods attempt to create explanations from the final model predictions, i.e. *post-hoc* (Madsen et al., 2022), through either attribution methods producing input saliency maps (Arrieta et al., 2020) or a lens for analysing a specific part of the model architecture (Vashishth et al., 2019b). Further, methods attempting a complete mechanistic interpretation have not been shown to be scalable as models get larger (Bereska and Gavves, 2024). Additionally, explainability methods have been shown to struggle with faithfulness towards the inner workings of the model, rationale and dataset consistency (Atanasova et al., 2020).

With these issues at heart, my dear reader, this thesis aims to create a set of tools for directly detecting, measuring, and mitigating systematic errors in the decision-making of deep learning models. In particular, I am interested in reasoning errors originating from opaque processes that mask erroneous behaviour in these models, data imbalances, and complex reasoning tasks. Towards this end, the research output of this thesis offers two methods for formulating adversarial setups in which the erroneous model behaviour is detected and explicitly quantified. Following this, the consequent research contributions focus on assessing the impact of imbalances in the training data on the reasoning behaviour of the model and further suggesting methods that mitigate these biases. Finally, we present two techniques that directly optimize complex reasoning tasks in knowledge graphs and natural language.

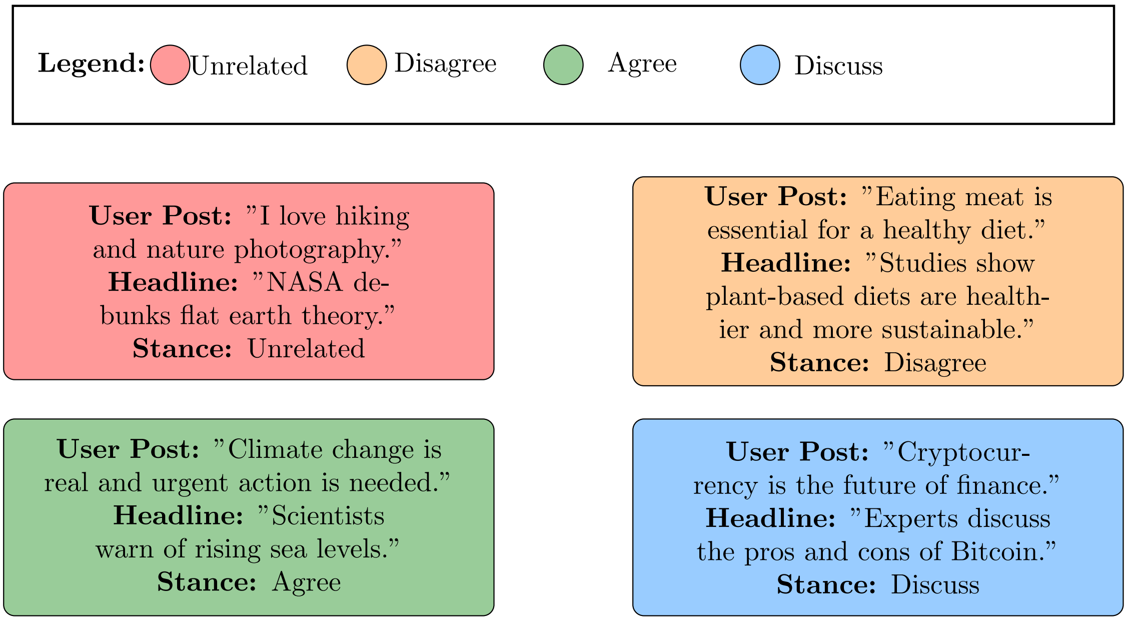
The following introductory sections 1.1.1 to 1.1.3 detail the background concepts and tasks relevant to the presented research contributions. This is followed by section 1.2 with a detailed overview of the individual contributions present within this thesis. section 1.3 provides a discussion of the mentioned research with suggested potential future exploration directions.

1.1.1 What is Reasoning in Deep Learning?

The concept of "*reasoning*" is one that has been studied and discussed vividly across the formulation of modern philosophy (Scriven, 1976). While the philosophical composition of this idea is not explicitly linked to this thesis, let us go on a slight tangent, which will clarify some motivations and ideas within the presented research. The *Dictinoray of Philosophy* defines "reasoning" as the "*The process of inferring conclusions from statement*" (Angeles, 1981). This definition garnered a famous critique (Walton, 1990) w.r.t. the use of the word "*inferring*", as it is ill-defined. The alternative formulation suggests defining an inference as the use of a rule for creating a connection between a set of propositions (statements). The initial set of propositions is the *premise* from which the inference starts and moves towards the *conclusions*. This allows to formalize reasoning as a directional process that links the premises to the conclusions through a rule. This, unsurprisingly, is rather similar to how models operate in Deep Learning (LeCun et al., 2015), with the aim of connecting the inputs to the outputs through a series of learned transformations/rules.

The advancements in deep learning have allowed the creation of architectures (Gu et al., 2018; Yu et al., 2019) that simultaneously learn both local and global features (Kavukcuoglu et al., 2010) through a series of intermediate transformations. Particularly with the emergence of transformers (Vaswani et al., 2017a), that utilise attention mechanisms (Brauwers and Frasincar, 2023) w.r.t. the input and intermediate representations, the ability of deep learning models to process complex tasks has significantly improved. These diverse mechanisms allow the deep learning models to perform a series of intermediate transformations that lead to the final output. This directional process is the definition of reasoning in these models.

As the deep models grow larger, we see their increased performance across a variety of tasks from natural language understanding and generation (Wang et al., 2019b;


Legend: 🔴 Unrelated 🟠 Disagree 🟢 Agree 🔵 Discuss

User Post: ”I love hiking and nature photography.” **Headline:** ”NASA debunks flat earth theory.” **Stance:** Unrelated

User Post: ”Eating meat is essential for a healthy diet.” **Headline:** ”Studies show plant-based diets are healthier and more sustainable.” **Stance:** Disagree

User Post: ”Climate change is real and urgent action is needed.” **Headline:** ”Scientists warn of rising sea levels.” **Stance:** Agree

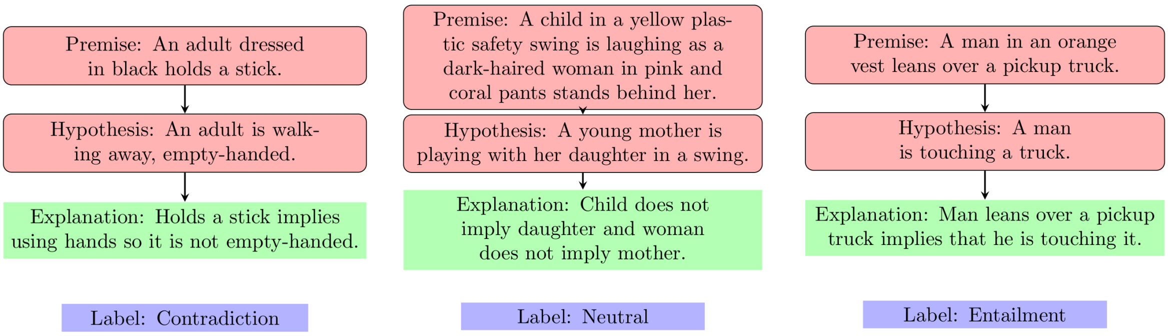
User Post: ”Cryptocurrency is the future of finance.” **Headline:** ”Experts discuss the pros and cons of Bitcoin.” **Stance:** Discuss

Figure 1.1: Examples of stance detection with labels: Unrelated, Disagree, Agree, and Discuss.

et al., 2023; Hendrycks et al., 2021a), image processing (Russakovsky et al., 2015; Lin et al., 2014) and complex query answering over knowledge graphs (Chen et al., 2020d). Although the performance increase is substantial, it must be noted that the tasks do not directly test the intermediate reasoning of the model and are evaluated strictly based on the final output. Some of the tasks used in the thesis are introduced in the following subsections and section 1.1.3.

1.1.1.1 Stance Detection

Stance Detection is a task in natural language processing where, given a piece of text, the model must identify the stance or attitude expressed towards the designated target (Küçük and Can, 2021). This target can be anything from a political issue, a social event, or an entity. Although the label vocabulary might vary for diverse formulations of stance detection (Hardalov et al., 2021b), an example of a task can be seen in fig. 1.1. The task is important in several domains, such as social media monitoring (AlDayel and Magdy, 2021) or political analysis (Lai et al., 2020), where understanding public sentiment can help predict trends, break down and study public opinion regarding existing discourse, or detect harmful content. It must be noted that accomplishing this requires the model to perform contextual and logical inference from the text and the fixed target to the linked attitude between them. This is where reasoning errors can manifest, as models may misinterpret nuanced expressions, irony, or indirect statements, especially when trained on imbalanced datasets.


Premise: An adult dressed in black holds a stick.
Hypothesis: An adult is walking away, empty-handed.
Explanation: Holds a stick implies using hands so it is not empty-handed.
Label: Contradiction
Premise: A child in a yellow plastic safety swing is laughing as a dark-haired woman in pink and coral pants stands behind her.
Hypothesis: A young mother is playing with her daughter in a swing.
Explanation: Child does not imply daughter and woman does not imply mother.
Label: Neutral
Premise: A man in an orange vest leans over a pickup truck.
Hypothesis: A man is touching a truck.
Explanation: Man leans over a pickup truck implies that he is touching it.
Label: Entailment

(a) Contradiction (b) Entailment (c) Neutral

Figure 1.2: Examples of natural language inference (NLI) reasoning with explanations and labels.

The research output discussed in the thesis (Arakelyan et al., 2023a, *Paper 3*) aims to address these challenges by proposing novel methodologies to improve stance detection across diverse topics. Specifically, the contributions include introducing a topic-guided diversity sampling strategy and a contrastive learning objective, both designed to enhance the model's ability to generalize effectively while mitigating class imbalance issues. The topic-guided diversity sampling technique ensures that the training data is balanced not only across classes but also among topics. This is achieved by prioritizing the selection of examples that maximize topic diversity while maintaining a representative sample of stance labels. The method counters the skewed distributions commonly found in stance detection datasets, allowing models to learn more robust and generalized representations.

1.1.1.2 Natural Language Inference

The task of textual entailment (Dagan et al., 2005), otherwise referred to as Natural Language Inference (Bowman et al., 2015, NLI), has been widely used to probe how well the models understand language (Condoravdi et al., 2003; Williams et al., 2017; Nie et al., 2019). This is a pairwise input task, where given a premise and a hypothesis, the objective is to predict if the premise *entails, contradicts* or is *neutral* towards the hypothesis. An example of this task can be seen in fig. 1.2.

This task is rather suited for examining model reasoning patterns, as it demands a logical and contextual understanding to predict the relationships between the premise and the hypothesis. The main challenge arises from the complexity of language nuances, such as ambiguity in wording and latently implied meanings and ideas, which demand deeper semantic comprehension. To solve this task, the model must be capable of mapping a set of transformations from premises to hypotheses, effectively simulating a form of reasoning. Consequently, NLI has become a cornerstone task

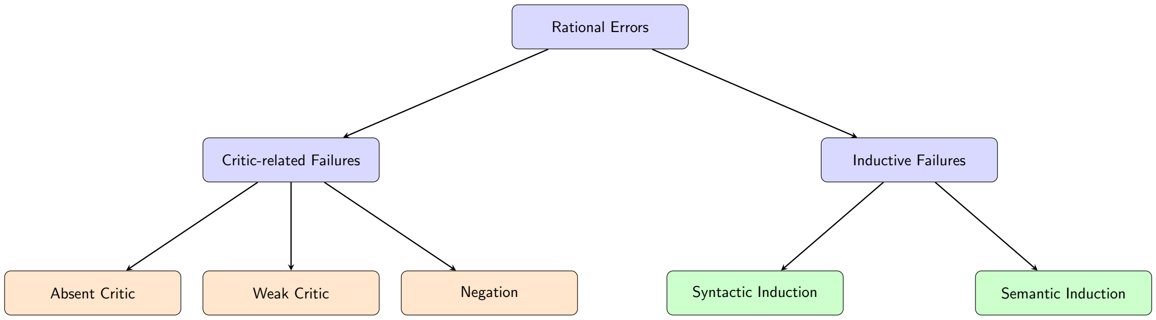


Figure 1.3: Taxonomy of Rationel Errors in human cognition (Ben-Zeev, 1998): Critic-related and Inductive Failures.

for evaluating reasoning in deep learning models due to its structured nature and availability of large-scale datasets like MNLI (Williams et al., 2017), SNLI (Bowman et al., 2015) and ANLI (Nie et al., 2020).

However, one must consider the limitations inherent in current benchmarks. For example, studies have shown that models often rely on shallow spurious correlations (McCoy et al., 2019a; Schuster et al., 2019), such as lexical overlap (Rajaee et al., 2022), lack generalisation out of distribution (Bhargava et al., 2021) or fail to acquire capabilities for abstract or logical reasoning (Talmor et al., 2020). Models have even been shown to achieve high performance without the presence of hypothesis (Gururangan et al., 2018b). This misalignment underscores the importance of developing evaluation methods that test reasoning directly, as opposed to proxy metrics.

In this thesis, we extend the exploration of NLI beyond conventional datasets by introducing adversarially constructed examples aimed at exposing reasoning flaws (Arakelyan et al., 2024b, *Paper 1*). We demonstrate that state-of-the-art Natural Language Inference models are sensitive towards minor surface-form variations that preserve semantics, which can cause significant inconsistencies in their inference decisions. Critically, this behaviour contrasts with a genuine, nuanced understanding of compositional semantics. However, it remains undetected when assessing model accuracy on traditional benchmarks or when probing for syntactic, monotonic, and logical reasoning capabilities. To analyze this phenomenon, we test NLI models on adversarially crafted examples featuring semantics-preserving surface-level noise. These examples are generated using conditional text generation, with a specific requirement that the NLI model identifies the relationship between the original and adversarial inputs as symmetric equivalence entailment.

1.1.2 Why does reasoning go wrong?

After we defined the notion of reasoning in the previous section, the next important task would be trying to formalise what a reasoning error is and gauge the potential

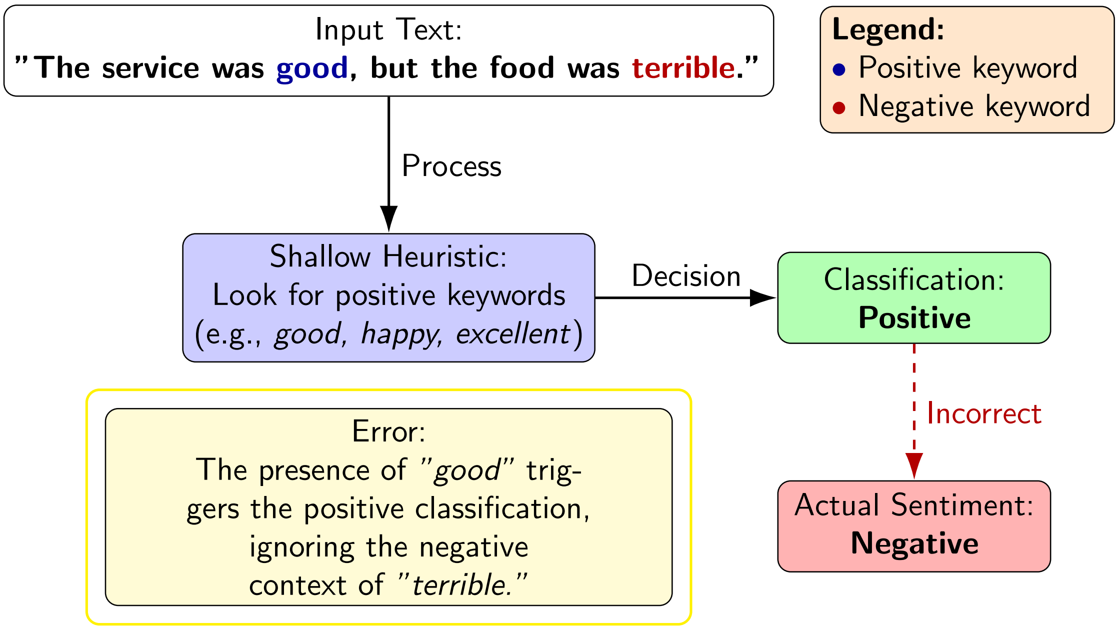

Input Text:
"The service was good, but the food was terrible."
Legend:
● Positive keyword
● Negative keyword
Process
Shallow Heuristic:
Look for positive keywords
(e.g., good, happy, excellent)
Decision
Classification:
Positive
Incorrect
Error:
The presence of "good" triggers the positive classification, ignoring the negative context of "terrible."
Actual Sentiment:
Negative

Figure 1.4: Example of a shallow heuristic used in a simple sentiment analysis reasoning process.

origins of those errors in deep learning. Prior research of errors in human rationale (Ben-Zeev, 2012) has suggested that systematic reasoning flaws observed in human cognition, tend to emerge not from randomness but from deliberate, rule-based processes (Ben-Zeev, 1998). As seen in fig. 1.3, the errors arise from misapplications of principles during learning or problem-solving, which are categorised into *critic-related* failures and *inductive misgeneralizations*. Critic-related failures occur when mechanisms to detect inconsistencies are missing, weak, or suppressed. For example, a model may fail to identify contradictions in the text due to inadequate intermediate validation. Inductive misgeneralizations arise from overgeneralizing or overspecializing rules based on patterns priorly internalised patterns. Semantic induction involves errors due to flawed analogies or understanding of ambiguous concepts.

This is directly related to reasoning inconsistencies we find in deep learning models. Models often exploit spurious correlations (Izmailov et al., 2022) internalized from the training data, such as associating specific keywords with outcomes, which breaks down in nuanced contexts (Wang et al., 2022a). A model might also be unable to create a comprehensive internal representation for further reasoning because of the complexity of the task (Vu et al., 2020). For instance, a model might misinterpret sarcasm by focusing on isolated words rather than broader context (Verma et al., 2021), or fail to produce a cohesive set of inferences when solving mathematical tasks (Patel et al., 2021a).

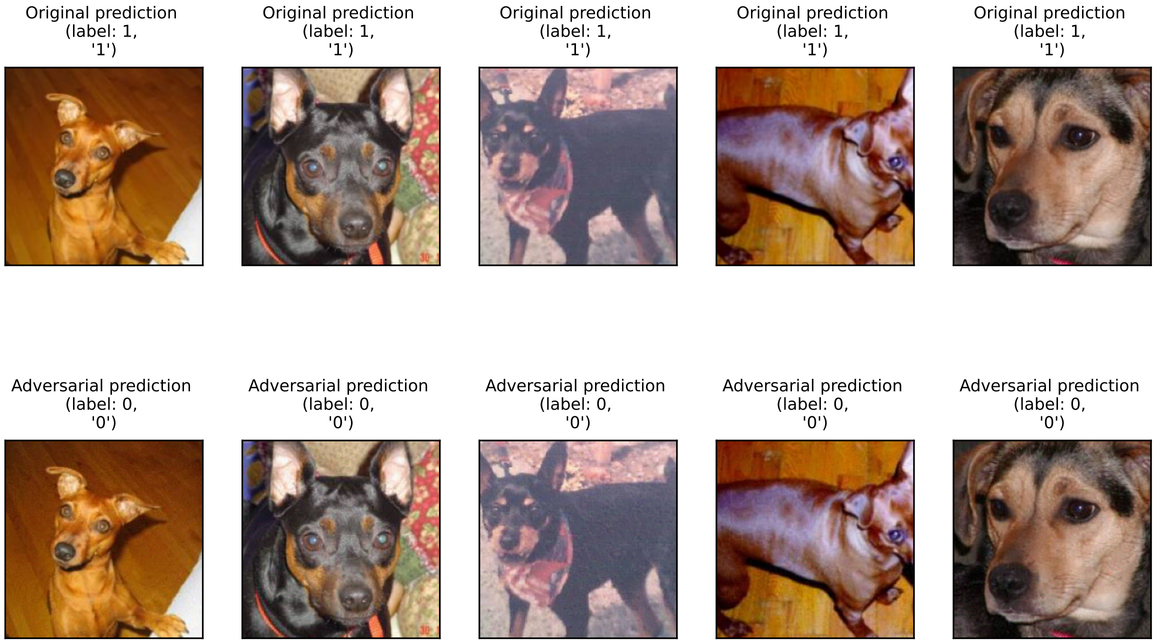

Original prediction (label: 1, '1')
Original prediction (label: 1, '1')
Original prediction (label: 1, '1')
Original prediction (label: 1, '1')
Original prediction (label: 1, '1')
Adversarial prediction (label: 0, '0')
Adversarial prediction (label: 0, '0')
Adversarial prediction (label: 0, '0')
Adversarial prediction (label: 0, '0')
Adversarial prediction (label: 0, '0')

Figure 1.5: Example of an imperceptible adversarial noise added (bottom) to the original image (top) that changes the final prediction of the model.

1.1.2.1 Internal Procedures

Errors arising from the internal procedures of a model often stem from a misalignment between the learned representations and their transitions into the final output and the ground truth. Deep learning models perform a sequence of transformations on input data, mapping it to intermediate representations and eventually to outputs. However, these transformations may inadvertently optimize for surface-level patterns rather than deeper semantic or logical relationships (McCoy et al., 2019a). For instance, models frequently rely on shallow heuristics, as shown in fig. 1.4, where specific tokens or phrases disproportionately influence predictions. These heuristics often lead to spurious correlations that degrade performance in nuanced contexts or under distributional shifts (Schuster et al., 2019). Moreover, models fail to perform necessary internal validation, leaving inconsistencies undetected and unresolved (Minervini et al., 2020).

Another reasoning inconsistency in deep learning models is the adversarial exploitability of their internal representations (Akhtar et al., 2021). These representations encode intermediate abstractions of input data through high-dimensional embeddings, ideally capturing semantic and logical relationships and reasoning required for the task. However, adversarial attacks exploit discrepancies within these representations, by introducing imperceptible noise to the input as seen in fig. 1.5 (Bhambri et al., 2019). This noise can significantly shift the internal representations, causing erroneous predictions, thus highlighting the susceptibility of the models towards non-semantic perturbations. This means maintaining consistency in inference is a non-trivial task in deep learning.

In this thesis, I systematically explore the reasoning inconsistencies stemming from both inductive and critic-related failures. Our findings (Arakelyan et al., 2024a, *Paper 2*) show that models trained from pre-trained backbones, like ResNet and ViT, are highly vulnerable to adversarial attacks, even when attackers possess only partial knowledge of the target model's tuning details. I introduce *backbone attacks*, which solely rely on the available feature extractors, and show that even such knowledge can induce significant disruptions, often matching the effectiveness of *white-box* attack strategies.

1.1.2.2 Data Imbalances

Imbalances in the data are a common origin for reasoning inconsistencies and discrepancies in deep learning models (Kaur et al., 2019). They occur when certain classes, features, or relationships are over or underrepresented in the training data, leading to biases emerging in the predictions of the model. Imbalances can manifest in various forms, such as class imbalance (Johnson and Khoshgoftaar, 2019a), topic imbalance, or semantic skew (Garrido-Muñoz et al., 2021), limiting the generalization capabilities of the model.

The presence of class imbalances means that a particular label has a dominant presence in the training data, which can cause the model to overfit the majority class while maintaining suboptimal reasoning patterns for minority classes, thus biasing predictions because of over-reliance on spurious correlations (Wang and Culotta, 2020). The topic or semantic imbalances limit the diversity of relationships the model internalizes, not allowing the model to adapt its reasoning to diverse contexts (Johnson and Khoshgoftaar, 2019b). Although strategies for mitigating class imbalances exist (Hasanin et al., 2019; Rendon et al., 2020), the pursuit of the same success for imbalances of semantic representation has been studied to a lesser degree.

In this thesis, I expand the current research on data-efficient sampling, introducing a topic-guided diversity sampling method that ensures that the training data is balanced not only across classes but also across topics and semantic nuances (Arakelyan et al., 2023a, *Paper 3*). By integrating this technique into model training, we show significant improvements in model accuracy, robustness on out-of-domain evaluation and reasoning consistency across domains.

1.1.2.3 Task Complexity

The last potential cause of reasoning inconsistencies discussed in this thesis is connected to task complexity. Since the emergence of probabilistic predictive methods, various frameworks have been proposed to measure and analyse the complexity of tasks w.r.t. the capacity of the predictive model (Blumer et al., 1989; Haussler, 1990) and the capability of the learning algorithms (Kearns and Vazirani, 1994) to find the optimal model for the designated task, forming what's known as its *effective capacity*.

Assessing the capacity of a deep learning model is challenging as the effective capacity depends on the chosen optimization algorithm in a non-convex setting, offering little theoretical insight (Hu et al., 2021). Consequently, quantifying the difficulty of the task given the designated model becomes an insurmountable challenge. The task complexity can be compounded by a plethora of factors, such as the number of reasoning steps required, the presence of hierarchical or nested dependencies, ambiguity in data representation, or the inherent difficulty of capturing abstract relationships. The inability to explicitly measure the complexity of the task, along with these challenges, limits the means to optimize a model for generalizing towards a comprehensive logical representation for that task, resulting in reasoning inconsistencies. For instance, in compositional generalization tasks (Keysers et al., 2020), where the goal is to generalize learned components to novel combinations, models often struggle to extrapolate rules to unseen contexts. Similarly, in mathematical reasoning (Saxton et al., 2019), solving problems with nested or multi-step operations requires structured reasoning pathways, retaining and recurrently reusing prior inductions and deductions across different reasoning stages. However, models often fail to maintain consistency in intermediate representations, leading to errors that accumulate over inference steps.

To address these challenges, the research output in the thesis presents a novel reasoning method over knowledge graphs (Arakelyan et al., 2023b, *Paper 5*) that includes learnable adaptation layers that directly optimise the intermediate answers and representations during the inference. This boosts the generalisation towards unseen types of queries and increases the *effective capacity* of the model, along with the added benefit that the method remains data-efficient.

Another discrepancy present in modern Large Language Models (LLM) is that while they exhibit strong performance on numerous language reasoning tasks, they often lack a structured and faithful inference mechanism when solving complex queries. This means that while the model might output tokens of intermediate reasoning, their exact impact on the final answer is not explicitly known. Moreover, the reasoning written in natural language lacks explicit verifiability because it is inherently freeform. To overcome this, we introduce **F**aithful **L**ogic-**A**ided **R**easoning and **E**xploration (**FLARE**) (Arakelyan et al., 2024c, *Paper 6*), a novel interpretable approach for traversing the problem space using task decompositions. The method enhances reasoning interpretability and faithfulness by combining task decomposition, Prolog-like logical formalization, and LLM simulated search. Critically, FLARE addresses task complexity by enhancing the reasoning capacity of LLMs without solely relying on deterministic algorithms. It supports multi-hop reasoning, task decomposition, and logical consistency verification. The results highlight FLARE's state-of-the-art performance on several datasets, achieving significant improvements in reasoning faithfulness and task accuracy. To overcome this, we introduce **F**aithful **L**ogic-**A**ided **R**easoning and **E**xploration (**FLARE**), a novel interpretable approach designed to navigate the problem space through task decompositions.

1.1.3 Reasoning with Complex Questions

The emergence of strong deep learning models in natural language (Vaswani et al., 2017b; Touvron et al., 2023a), image processing (Dosovitskiy et al., 2021; Team, 2024) and query answering over knowledge graphs (Galkin et al., 2024), created the necessity for more elaborate evaluation benchmarks. The main added components for reasoning over these new datasets (Yang et al., 2018; Kwiatkowski et al., 2019; Cobbe et al., 2021), was that the models needed to adapt to the presence of semantic ambiguity (Geva et al., 2021) within the questions, the necessity for multi-hop reasoning (Yang et al., 2018), the need for adaptability to diverse logical paradigms (Saparov and He, 2023; Zhong et al., 2021; et al., 2023) and the ability for more rigorous task formalization, decomposition and exploration (Hendrycks et al., 2021b; Glazer et al., 2024). In this thesis, we detect and mitigate reasoning inconsistencies in deep learning models that operate over natural language and knowledge graphs.

1.1.3.1 Complex Multi-hop Question Answering

Complex multi-hop question-answering tasks require models to reason over ambiguously worded information distributed across various contexts, sources, and commonsense or logical implications. Unlike single-hop tasks, where a direct relationship exists between the query and the answer, multi-hop reasoning involves intermediate steps, where the output of one step serves as input for the next. Examples include connecting facts across sentences, documents, or knowledge graph entities to arrive at a final answer. For instance, consider a question like, "Which author wrote the book that inspired the movie 'Blade Runner'?" To answer this, a model must connect multiple pieces of information: identifying that Blade Runner was inspired by the book "*Do Androids Dream of Electric Sheep?*" and then recognizing that the book's author is Philip K. Dick. Such tasks demand robust semantic understanding, logical consistency, and precise chaining of inferences. urrent benchmarks, such as HotpotQA (Yang et al., 2018) and ComplexWebQuestions (Talmor and Berant, 2018), aim to evaluate these multi-step reasoning abilities but are insufficient to assess reasoning faithfulness. Models might produce correct answers without following valid reasoning paths even if they produce tokens that seem like correct intermediate justificaitions.

1.1.3.2 Complex Logical Query Answering over Knowledge Graphs

A Knowledge Graph (KG) is a knowledge base representing the relationships between entities in a relational graph structure. The flexibility of this knowledge representation formalism allows KGs to be widely used in various domains. A Knowledge Graph $\mathcal{G} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ can be defined as a set of subject-predicate-object $\langle s, p, o \rangle$ triples, where each triple encodes a relationship of type $p \in \mathcal{R}$ between the subject $s \in \mathcal{E}$ and the object $o \in \mathcal{E}$ of the triple, where $\mathcal{E}$ and $\mathcal{R}$ denote the set of all entities and relation types, respectively. A Knowledge Graph can be represented as a First-Order Logic Knowledge Base, where each triple $\langle s, p, o \rangle$ denotes an atomic formula $p(s, o)$, with $p \in \mathcal{R}$ a binary predicate and $s, o \in \mathcal{E}$ its arguments. We are concerned with answering logical queries over incomplete knowledge graphs. We consider queries that use existential quantification ($\exists$) and conjunction ($\wedge$) operations. Furthermore, we include disjunctions ($\vee$) and atomic negations ($\neg$).

Consider the question "*Which people are German and produced the music for the film Constantine?*". It can be formalised as a complex query $\mathcal{Q} \equiv ?T : \text{country}(\text{Germany}, T) \wedge \text{producerOf}(\text{Constantine}, T)$, where *Germany* and *Constantine* are anchor nodes, and T is the target of the query. The answer $[\mathcal{Q}]$ corresponds to all the entities in the knowledge graph that are German composers for the film Constantine. We propose a novel method for reasoning over knowledge graphs introduced in (Arakelyan et al., 2023b).

1.2 Scientific Contributions

1.2.1 Reasoning Inconsistencies from Internal Processes

1.2.1.1 Paper 1: Semantic Sensitivities and Inconsistent Predictions: Measuring the Fragility of NLI Models

Recent studies of the emergent capabilities of transformer-based Natural Language Understanding (NLU) models have indicated that they have an understanding of lexical and compositional semantics. I provide evidence that suggests these claims should be taken with a grain of salt: finding that state-of-the-art Natural Language Inference (NLI) models are sensitive towards minor semantics preserving surface-form variations, which lead to sizable inconsistent model decisions during inference. This behavior diverges from a genuine and robust comprehension of compositional

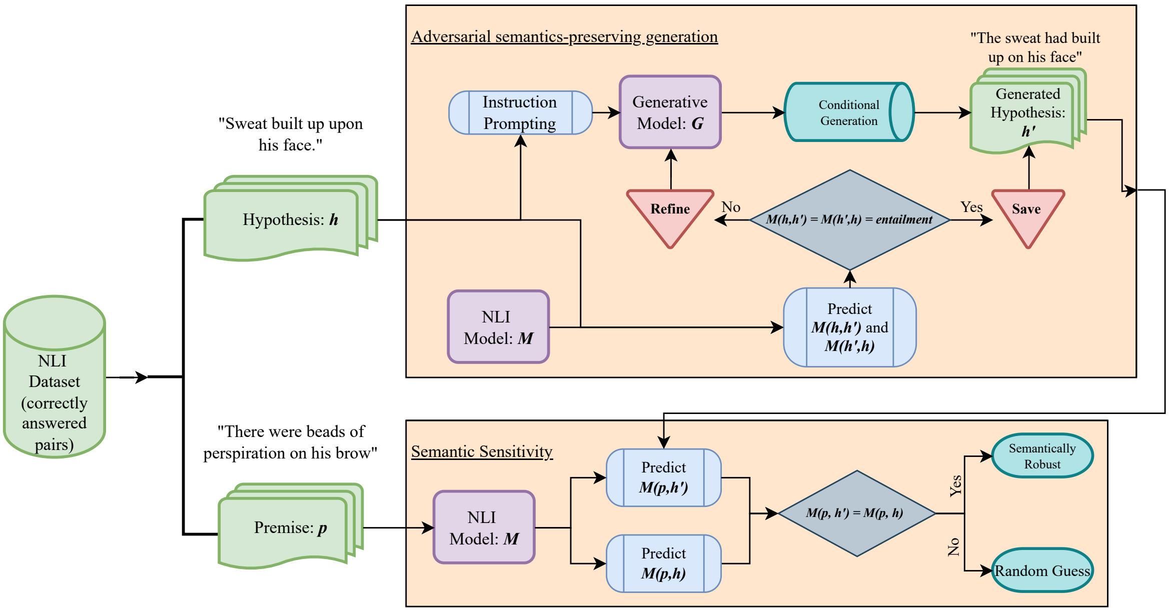

Adversarial semantics-preserving generation
"The sweat had built up on his face"
"Sweat built up upon his face."
Instruction Prompting
Generative Model: *G*
Conditional Generation
Generated Hypothesis: *h'*
Hypothesis: *h*
Refine
No
M(h,h') = M(h',h) = entailment
Yes
Save
NLI Dataset (correctly answered pairs)
NLI Model: *M*
Predict *M(h,h')* and *M(h',h)*
"There were beads of perspiration on his brow"
Semantic Sensitivity
Premise: *p*
NLI Model: *M*
Predict *M(p,h')*
Predict *M(p,h)*
M(p, h') = M(p, h)
Yes
No
Semantically Robust
Random Guess

Figure 1.6: The proposed framework is comprised of two components. (i) a module for generating semantics-preserving surface-form hypothesis variations and (ii) using the generated surface for measuring semantic sensitivity and predictive inconsistency.

semantics. Notably, it does not explicitly emerge when evaluating model accuracy on standard benchmarks or during probing for syntactic, monotonic, and logically robust reasoning. To address this, I propose a novel framework, illustrated in fig. 1.6, for quantifying semantic sensitivity. This framework evaluates NLI models on adversarially generated examples containing minor semantics-preserving surface-form variations. These adversarial examples are created using conditional text generation, with the explicit condition that the NLI model should predict the relationship between the original and adversarial inputs as a symmetric equivalence entailment.

I systematically examine the effects of this phenomenon across NLI models in both *in-domain* and *out-of-domain* settings. Experimental results reveal that semantic sensitivity leads to performance degradations of 12.92% and 23.71% on average for *in-domain* and *out-of-domain* settings, respectively. Furthermore, through ablation studies, I analyze this phenomenon across various models, datasets, and inference variations, demonstrating that semantic sensitivity can cause significant inconsistencies in model predictions.

1.2.1.2 Paper 2: With Great Backbones Comes Great Adversarial Transferability

Advancements in self-supervised learning (SSL) for machine vision have enhanced representation robustness and model performance, leading to the emergence of publicly shared pre-trained backbones, such as *ResNet* and *ViT* models tuned with SSL

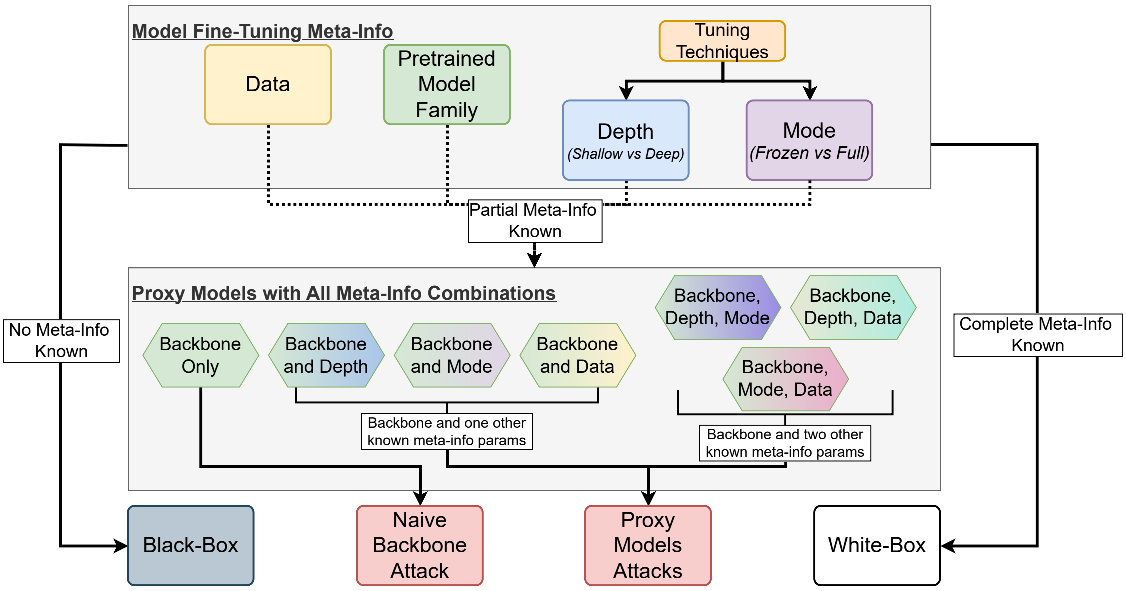


Figure 1.7: The figure depicts all of the settings used to evaluate adversarial vulnerabilities given different information of the target model construction. From left to right, I simulate exhaustive varying combinations of meta-information available about the target model during adversarial attack construction. All of the created proxy models are used separately to assess adversarial transferability.

methods like *SimCLR*. Due to the computational and data demands of pre-training, the utilization of such backbones becomes a strenuous necessity. However, employing such backbones may imply adhering to the existing vulnerabilities towards adversarial attacks. Prior research on adversarial robustness typically examines attacks with either full (*white-box*) or no access (*black-box*) to the target model, but the adversarial robustness of models tuned on known pre-trained backbones remains largely unexplored. Furthermore, it is unclear which tuning meta-information is critical for mitigating exploitation risks. In this work, I systematically study the adversarial robustness of models that use such backbones, evaluating 20000 combinations of tuning meta-information, including fine-tuning techniques, backbone families, datasets, and attack types, as seen in fig. 1.7. To uncover and exploit potential vulnerabilities, I propose using proxy (surrogate) models to transfer adversarial attacks, fine-tuning these proxies with various tuning variations to simulate different levels of knowledge about the target. Our findings show that proxy-based attacks can reach close performance to strong *black-box* methods with sizable budgets and closing to *white-box* methods, exposing vulnerabilities even with minimal tuning knowledge. Additionally, we introduce a naive "backbone attack", leveraging only the shared backbone to create adversarial samples, demonstrating an efficacy surpassing *black-box* and close to *white-box* attacks and exposing critical risks in model-sharing practices. Finally, our

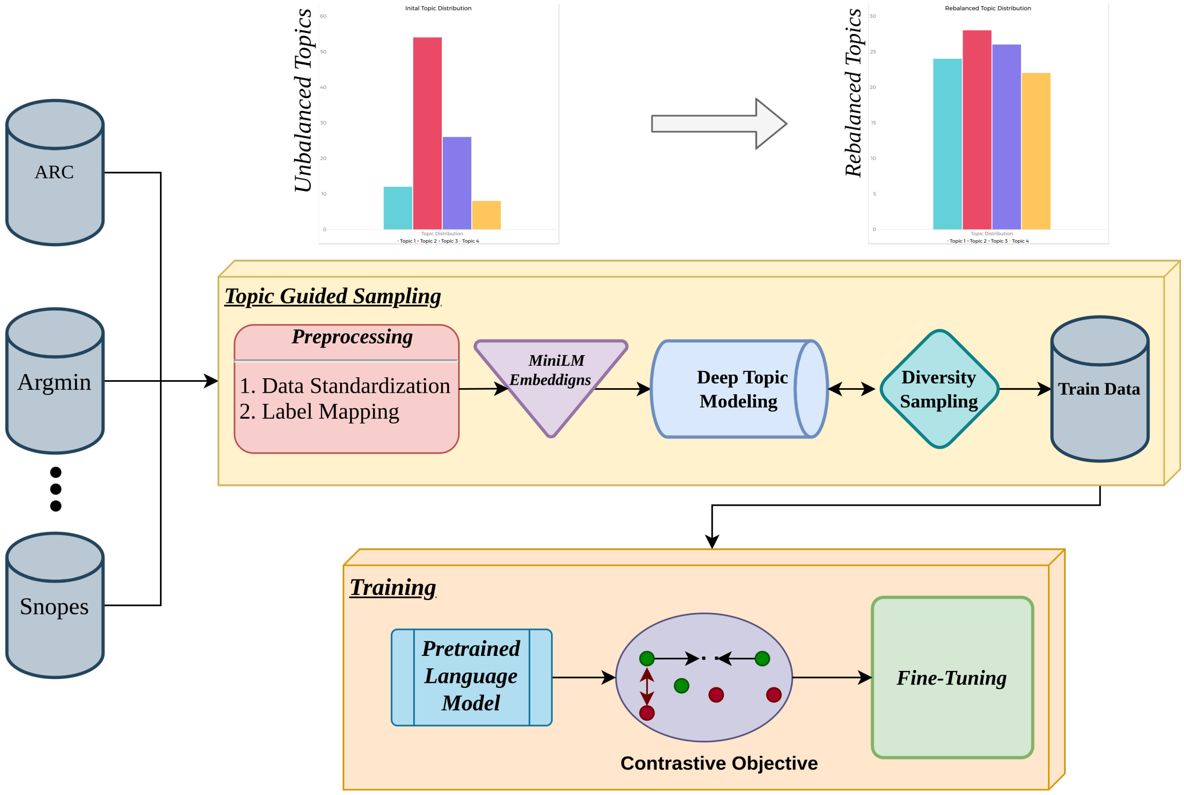


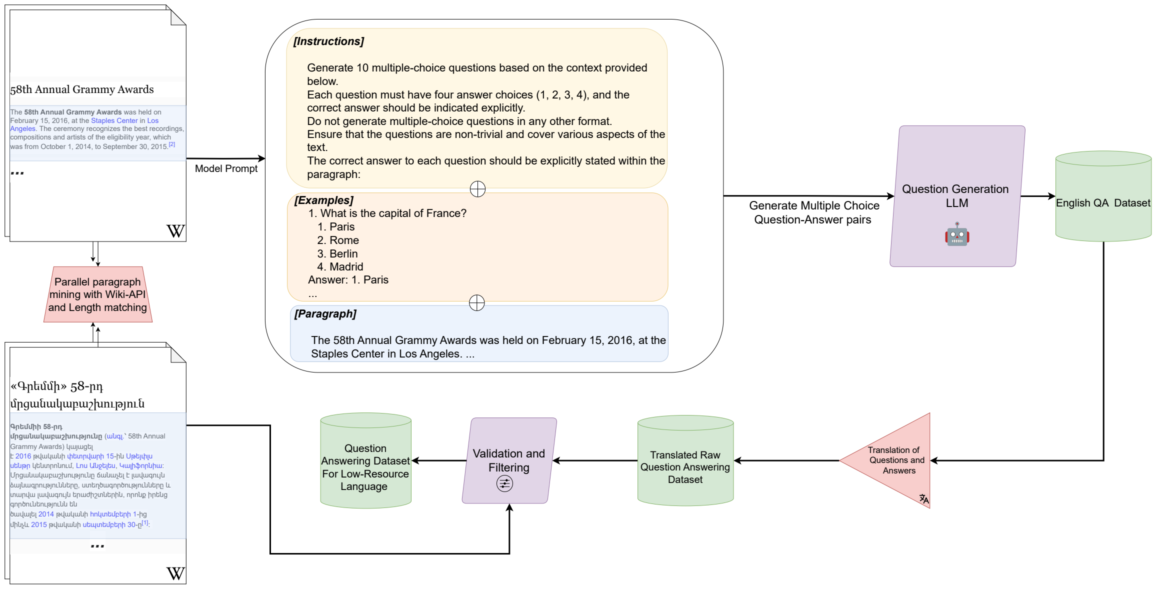
Figure 1.8: The two components of TESTED: Topic Guided Sampling (top) and training with contrastive objective (bottom).

ablations reveal how increasing tuning meta-information impacts attack transferability, measuring each meta-information combination.

1.2.2 Reasoning Inconsistencies from Data

1.2.2.1 Paper 3: Topic-Guided Sampling For Data-Efficient Multi-Domain Stance Detection

Stance Detection is concerned with identifying the attitudes expressed by an author towards a target of interest. This task spans a variety of domains ranging from social media opinion identification to detecting the stance for a legal claim. However, the framing of the task varies within these domains, in terms of the data collection protocol, the label dictionary and the number of available annotations. Furthermore, these stance annotations are significantly imbalanced on a per-topic and inter-topic basis. These make multi-domain stance detection a challenging task, requiring standardization and domain adaptation. To overcome this challenge, I propose **T**opic **E**fficient **St**anc**E** **D**etection (TESTED), seen in fig. 1.8, consisting of a topic-guided diversity sampling technique and a contrastive objective that is used for fine-tuning a stance classifier. I evaluate the method on an existing benchmark of 16 datasets with in-domain, i.e. all topics seen and out-of-domain, i.e. unseen topics, experiments. The


58th Annual Grammy Awards

The 58th Annual Grammy Awards was held on February 15, 2016, at the Staples Center in Los Angeles. The ceremony recognizes the best recordings, compositions and artists of the eligibility year, which was from October 1, 2014, to September 30, 2015.[2]

...

[Instructions]

Generate 10 multiple-choice questions based on the context provided below.
Each question must have four answer choices (1, 2, 3, 4), and the correct answer should be indicated explicitly.
Do not generate multiple-choice questions in any other format.
Ensure that the questions are non-trivial and cover various aspects of the text.
The correct answer to each question should be explicitly stated within the paragraph:

[Examples]

1. What is the capital of France?
   1. Paris
   2. Rome
   3. Berlin
   4. Madrid

Answer: 1. Paris
...

[Paragraph]

The 58th Annual Grammy Awards was held on February 15, 2016, at the Staples Center in Los Angeles. ...

«Գրեմմի» 58-րդ մրցանակաբաշխություն

Գրեմմիի 58-րդ մրցանակաբաշխությունը (անգլ.՝ 58th Annual Grammy Awards) կայացել է 2016 թվականի փետրվարի 15-ին Սթեյփլս սենթր կենտրոնում, Լոս Անջելես, Կալիֆորնիա: Մրցանակաբաշխությունը ճանաչել է լավագույն ձայնագրությունները, ստեղծագործությունները և տարվա լավագույն երաժիշտներին, որոնք իրենց գործունեությունն են ծավալել 2014 թվականի հոկտեմբերի 1-ից մինչև 2015 թվականի սեպտեմբերի 30-ը[1]:

...

W
Model Prompt
Parallel paragraph mining with Wiki-API and Length matching
Generate Multiple Choice Question-Answer pairs
Question Generation LLM
English QA Dataset
Translation of Questions and Answers
Translated Raw Question Answering Dataset
Validation and Filtering
Question Answering Dataset For Low-Resource Language
W

Figure 1.9: The proposed framework is comprised of three components: (i) a module for mining parallel paragraphs using wiki-API and length matching; (ii) generating a synthetic question-answering dataset with an LLM using the mined English paragraphs; (iii) translating the question-answer pairs and Filtering/Validating them for obtaining a high-quality synthetic QA dataset in the low-resource language.

results show that our method outperforms the state-of-the-art with an average of 3.5 F1 points increase in-domain, and is more generalizable with an averaged increase of 10.2 F1 on out-of-domain evaluation while using $\leq 10\%$ of the training data. I show that our sampling technique mitigates both inter- and per-topic class imbalances. Finally, our analysis demonstrates that the contrastive learning objective allows the model a more pronounced segmentation of samples with varying labels.

1.2.2.2 Paper 4: SynDARin: Synthesising Datasets for Automated Reasoning in Low-Resource Languages

Question Answering (QA) datasets have been instrumental in developing and evaluating Large Language Model (LLM) capabilities. However, such datasets are scarce for languages other than English due to the cost and difficulties of collection and manual annotation. This means that producing novel models and measuring the performance of multilingual LLMs in low-resource languages is challenging. To mitigate this, I propose Syn**DAR**in, a method for generating and validating QA datasets for low-resource languages, seen in fig. 1.9. I utilize parallel content mining to obtain *human-curated* paragraphs between English and the target language. I use the English data as context to *generate* synthetic multiple-choice (MC) question-answer pairs, which are automatically translated and further validated for quality. Combining these with their designated non-English *human-curated* paragraphs from the final QA dataset.

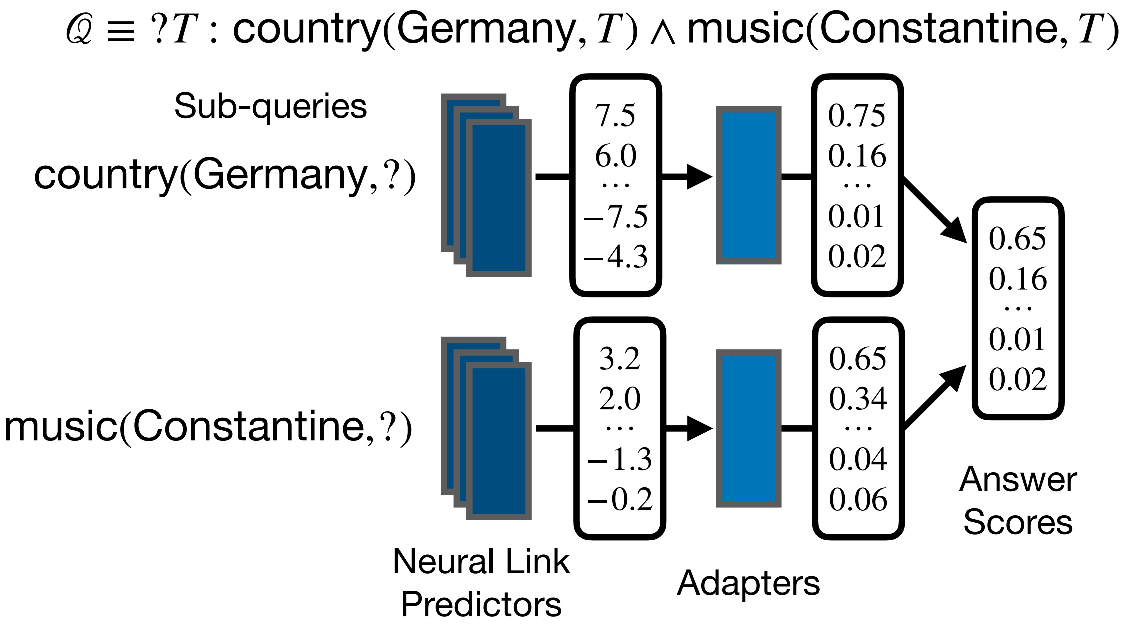


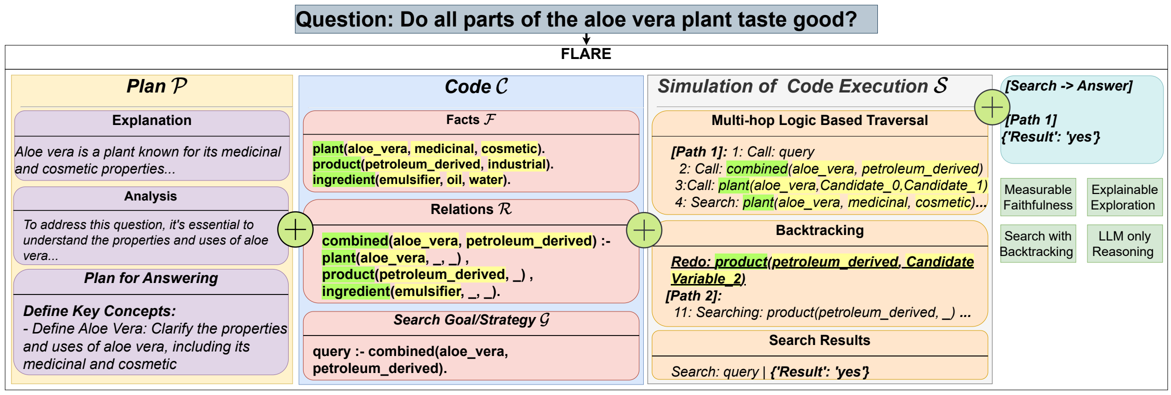
Figure 1.10: Given a complex query $\mathcal{Q}$, CQD$^{\mathcal{A}}$ adapts the neural link prediction scores for the sub-queries to improve the interactions between them.

The method allows to maintain content quality, reduces the likelihood of factual errors, and circumvents the need for costly annotation. To test the method, I created a QA dataset with 1.2K samples for the Armenian language. The human evaluation shows that 98% of the generated English data maintains quality and diversity in the question types and topics, while the translation validation pipeline can filter out $\sim 70\%$ of data of poor quality. I use the dataset to benchmark state-of-the-art LLMs, showing their inability to achieve human accuracy with some model performances closer to random chance. This shows that the generated dataset is non-trivial and can be used to evaluate reasoning capabilities in low-resource language.

1.2.3 Reasoning Inconsistencies from Task Complexity

1.2.3.1 Paper 5: Adapting Neural Link Predictors for Data-Efficient Complex Query Answering

Answering complex queries on incomplete knowledge graphs is a challenging task where a model needs to answer complex logical queries in the presence of missing knowledge. Prior work in the literature has proposed to address this problem by designing architectures trained end-to-end for the complex query answering task with a reasoning process that is hard to interpret while requiring data and resource-intensive training. Other lines of research have proposed re-using simple neural link predictors to answer complex queries, reducing the amount of training data by orders



Aloe vera is a plant known for its medicinal and cosmetic properties...

To address this question, it's essential to understand the properties and uses of aloe vera...

Define Key Concepts:
- Define Aloe Vera: Clarify the properties and uses of aloe vera, including its medicinal and cosmetic

```
plant(aloe_vera, medicinal, cosmetic).
product(petroleum_derived, industrial).
ingredient(emulsifier, oil, water).
```

```
combined(aloe_vera, petroleum_derived) :-
plant(aloe_vera, _, _) ,
product(petroleum_derived, _) ,
ingredient(emulsifier, _, _).
```

```
query :- combined(aloe_vera,
petroleum_derived).
```

```
[Path 1]: 1: Call: query
  2: Call: combined(aloe_vera, petroleum_derived)
 3:Call: plant(aloe_vera,Candidate_0,Candidate_1)
 4: Search: plant(aloe_vera, medicinal, cosmetic)...
```

```
Redo: product(petroleum_derived, Candidate
Variable_2)
[Path 2]:
 11: Searching: product(petroleum_derived, _) ...
```

```
Search: query | {'Result': 'yes'}
```

```
[Path 1]
{'Result': 'yes'}
```

Figure 1.11: A depiction of the *plan*, *code* and simulated *search* in FLARE. Each module is generated separately and iteratively, allowing us to obtain the final answer. The green and yellow highlighted text shows the overlap between the facts and the relations between the code and the simulated search.

of magnitude while providing interpretable answers. The neural link predictor used in such approaches is not explicitly optimised for the complex query answering task, implying that its scores are not calibrated to interact together. We propose to address these problems via CQD$^{\mathcal{A}}$, a parameter-efficient score *adaptation* model optimised to re-calibrate neural link prediction scores for the complex query answering task. While the neural link predictor is frozen, the adaptation component – which only increases the number of model parameters by 0.03% – is trained on the downstream complex query answering task. Furthermore, the calibration component enables us to support reasoning over queries that include atomic negations, which was previously impossible with link predictors. In our experiments, CQD$^{\mathcal{A}}$ produces significantly more accurate results than current state-of-the-art methods, improving from 34.4 to 35.1 Mean Reciprocal Rank values averaged across all datasets and query types while using $\leq 30\%$ of the available training query types. We further show that CQD$^{\mathcal{A}}$ is data-efficient, achieving competitive results with only 1% of the complex training queries, and robust in out-of-domain evaluations.

1.2.3.2 Paper 6: FLARE: Faithful Logic-Aided Reasoning and Exploration

Modern Question Answering (QA) and Reasoning approaches based on Large Language Models (LLMs) commonly use prompting techniques, such as Chain-of-Thought (CoT), assuming the resulting generation will have a more granular exploration and reasoning over the question space and scope. However, such methods struggle with generating outputs that are faithful to the intermediate chain of reasoning produced by the model. On the other end of the spectrum, neuro-symbolic methods such as Faithful CoT (F-CoT) and Logic-LM propose to combine LLMs with external symbolic solvers.

	Internal Procedures			Data Imbalances			Complex Reasoning		
	M	**D**	**A**	**M**	**D**	**A**	**M**	**D**	**A**
1. Arakelyan et al. (2024b)	✓	✓	✓						
2. Arakelyan et al. (2024a)	✓		✓						
3. Arakelyan et al. (2023a)				✓		✓			
4. Ghazaryan et al. (2024)				✓	✓	✓		✓	
5. Arakelyan et al. (2023b)							✓		✓
6. Arakelyan et al. (2024c)							✓		✓
7. Cochez et al. (2023)									✓

Table 1.1: Contributions of referenced works across three main reasoning inconsistency categories: Internal Procedures, Data Imbalances, and Complex Reasoning Tasks. Each category is further divided into three contribution subsections: Method (**M**), Database (**D**), and Analysis Framework For Reasoning (**A**). A checkmark (✓) indicates the contribution's relevance to the respective area. The table highlights the distribution of efforts, showcasing where each work has made significant contributions.

While such approaches boast a high degree of faithfulness, they usually require a model trained for code generation and struggle with tasks that are ambiguous or hard to formalise strictly. I introduce **F**aithful **L**ogic-**A**ided **R**easoning and **E**xploration (**FLARE**), a novel interpretable approach for traversing the problem space using task decompositions, seen in fig. 1.11. I use the LLM to plan a solution, formalise the query into facts and predicates, which form the problem space, using a logic programming code and simulate that code execution using an exhaustive multi-hop search over the defined space. Our method allows us to compute the faithfulness of the reasoning process w.r.t. the generated code and explicitly trace the steps of the multi-hop search without relying on external solvers. Our methods achieve SOTA results on **7** out of **9** diverse reasoning benchmarks. I also show that model faithfulness positively correlates with overall performance and further demonstrate that **FLARE** allows pinpointing the decisive factors sufficient for and leading to the correct answer with optimal reasoning during the multi-hop search. Our findings reveal that successful traces exhibit, on average, a 18.1% increase in unique emergent facts, a 8.6% higher overlap between code-defined and execution-trace relations, and a 3.6% reduction in unused code relations.

1.3 Discussion and Future Work

The research publications within this thesis contribute to the field of deep learning by expanding our understanding of reasoning inconsistencies and suggesting novel ways to mitigate them across various domains such as NLP, image processing and reasoning over KGs. In particular, we identify and analyse three potential causes for inconsistency: internal processes, data imbalances, and task complexity. The contributions can be segmented into novel methods, datasets and analysis frameworks for detecting, measuring and mitigating reasoning inconsistencies in deep learning models, as seen in table 1.1.

1.3.1 Measuring and Mitigating Reasoning Inconsistencies

The thesis establishes reasoning inconsistencies related to internal representations and transitions that deep learning models learn. The papers suggest novel adversarial setups that directly expose the susceptibility of deep learning models to shallow heuristics, semantic misalignments, and adversarial vulnerabilities. For instance, the semantic sensitivity of NLI models (Arakelyan et al., 2024b, *Paper 1*) highlights a critical limitation in their robustness and generalization across diverse settings. Adversarial transferability studies (Arakelyan et al., 2024a, *Paper 2*) reveal how shared backbones escalate model vulnerabilities, emphasizing the need for more robust fine-tuning techniques and vigilance in the current model-sharing practices. In both of the publications, we provide a framework for directly detecting and measuring the extent of reasoning inconsistencies that numerous deep learning models possess.

The thesis also includes a new method for data-efficient topic-based sampling (Arakelyan et al., 2023a, *Paper 3*), which allows to circumvent the complications with biased and inconsistent reasoning in deep learning models, arising from dataset imbalances described in section 1.1.2.2. We directly measure the impact of mitigating these imbalances with our method on the predictive capabilities of the model, showing a significant performance boost both for in-domain and out-of-domain evaluations. The models trained using our sampling method are also less susceptible to erroneous behaviour that arises because of a dominating overrepresentation of specific topics or semantic features. We also show that an emergent property of the contrastive learning method we propose is that the internal representations of the model become more segmented w.r.t. different topics, thus overall boosting the model's effective capacity.

The thesis also contributes a method for synthetically generating question-answering datasets in low-resource settings (Ghazaryan et al., 2024, *Paper 4*) with a mechanism for automated sample verification and diversification. I constructed a human evaluation for each aspect of the method and the final output and showed that the method works for the Armenian language, which has almost no available machine learning resources for training and evaluation. Some would argue that allegorically, the biggest data imbalance is the availability of no data for even evaluating the reasoning capabilities of the deep learning models, which this publication addresses. Maybe some would be wrong in this assessment, but that, my dear reader, is a discussion for a different thesis.

To tackle the reasoning inconsistencies that emerge because of task complexity and the limited *effective capacity*/expressivity of the model, I dive into two complex query answering tasks over knowledge graphs and natural language. I introduce a new approach for handling complex queries over knowledge graphs (Arakelyan et al., 2023b, *Paper 5*), incorporating learnable adaptation layers that optimize intermediate answers and representations during the reasoning process. This enhances the model's ability to generalize to unseen query types, increases its effective capacity, and maintains data efficiency as an added advantage. This allows for circumventing prior limitations present because of task complexity and adds an ingrained tool for verifying and adapting the results of intermediate reasoning answers.

The other aspects I explore in this thesis are the predictive inconsistencies and sub-optimal explorations that LLMs have when reasoning in natural language. While LLMs exhibit strong performance on numerous language reasoning tasks, they often lack a structured and faithful inference mechanism when answering complex queries, which does not allow the model to formalise and explore the problem efficiently. Moreover, many prompting paradigms in natural language lack explicit verifiability because the text is inherently freeform. We create a novel method to mitigate these issues. **F**aithful **L**ogic-**A**ided **R**easoning and **E**xploration (**FLARE**) (Arakelyan et al., 2024c, *Paper 6*), is a novel interpretable approach for traversing the problem space using task decompositions. The method enhances reasoning interpretability and faithfulness by combining task decomposition, Prolog-like logical formalization, and LLM simulated search. Critically, FLARE addresses task complexity by enhancing the reasoning capacity of LLMs without solely relying on deterministic algorithms and allows for deeper model explorations within the problem. It supports multi-hop reasoning, task decomposition, and logical consistency verification. The results highlight FLARE's state-of-the-art performance on several datasets, achieving significant improvements in reasoning faithfulness and task accuracy.

1.3.2 Future Work

Currently, there is limited work on ingraining LLMs with test time compute capabilities and how it impacts the effective capacity of the model and the verifiability of the suggested reasoning lines. One natural extension for papers (Arakelyan et al., 2023b, 2024c, *5 and 6*) would be to directly use test time compute mechanisms, such as self-refinement and others for further enriching the predictive capabilities of the models. An intriguing alternative in a similar vein would involve training a reward model using the formalizations and reasoning paths generated by FLARE. This approach leverages the ability to directly execute the generated code, allowing us to sample Prolog search paths that yield correct answers as positive examples while treating the LLM-generated incorrect traversal simulations as negative examples. Training this type of model would allow to further tune other LLMs with a differentiable oracle that is capable of assessing the correctness and completeness of the search paths. This would also allow some notion of verifiability to be ingrained into the generated search paths. A direct extension of this can be using a strict logical decomposition and iterative multi-hop reasoning for the query, similar to the approach in (Arakelyan et al., 2023b, *5*). This would add the capability to adaptably search over intermediate answers and prune unlikely search directions.

Part II

Reasoning Inconsistencies from Internal Processes

2 Semantic Sensitivities and Inconsistent Predictions: Measuring the Fragility of NLI Models

2.1 Introduction

Transformer-based (Vaswani et al., 2017b) Language Models (LMs) have shown solid performance across various NLU tasks (Wang et al., 2018a, 2019a). These advances have led to suggestions regarding the emergent capabilities of the models in terms of syntactic (Sinha et al., 2020; Hewitt and Manning, 2019; Jawahar et al., 2019; Warstadt and Bowman, 2020), logic (Wei et al., 2022a,b) and semantic (Kojima et al., 2022a; Dasgupta et al., 2022) understanding. However, we present novel evidence that indicates that these models are prone to inconsistent predictions induced by inherent susceptibility towards semantic sensitivities.

To probe the models for these discrepancies, we formalise *semantic comprehension* as the ability to distinguish logical relations within sentences through identifying compositional semantics (Jacobson, 2014; Carnap, 1959). This means that negligible semantic variations should not impact the inherent relations implied between the texts, e.g. *"There were beads of perspiration on his brow."* entails both *"Sweat built up upon his face."* and the slight variation *"The sweat had built up on his face."* Authentic comprehension of semantics does allow for such understanding through discovering semantic structures and the inherent relations induced by them (Cicourel, 1991; Schiffer, 1986; Rommers et al., 2013). This means that analysing the emergent semantic understanding within a model should minimally involve testing for sensitivity towards semantics-preserving surface-form variations.

We particularly focus on the task of textual entailment (Dagan et al., 2005), otherwise referred to as Natural Language Inference (Bowman et al., 2015, NLI), which has been widely used to probe how well the models understand language (Condoravdi et al., 2003; Williams et al., 2017; Nie et al., 2019). This is a pairwise input task, where given a premise p and a hypothesis h, the objective is to predict if the premise *entails, contradicts* or is *neutral* towards the hypothesis.

We propose a framework for testing semantic sensitivity within transformer-based models trained for NLI, by creating semantics-preserving surface-form variations of the

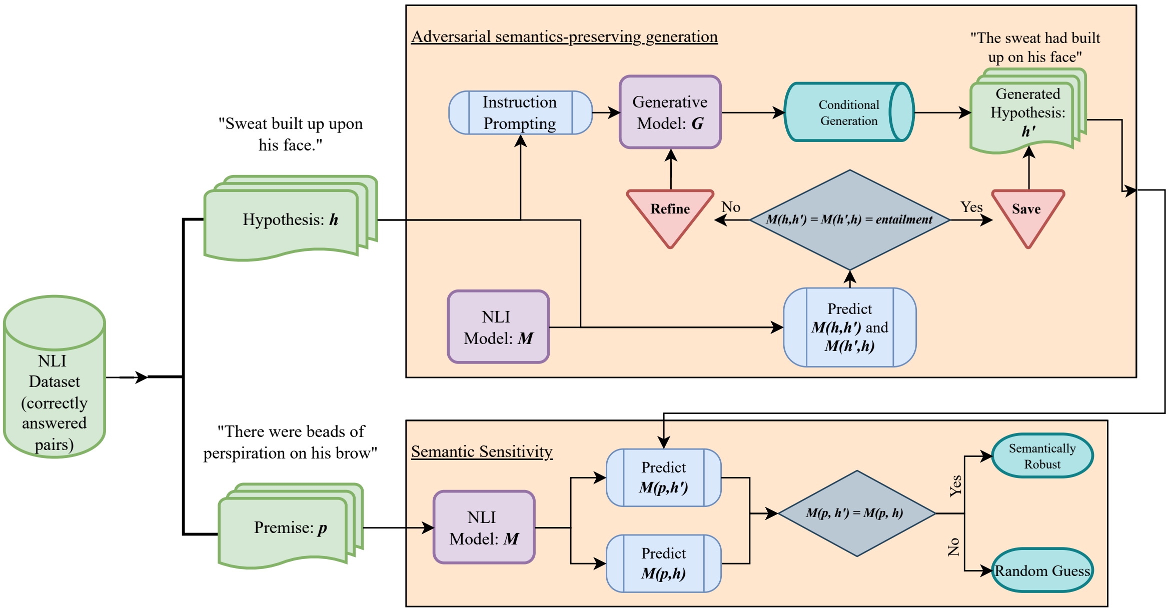

Adversarial semantics-preserving generation
"The sweat had built up on his face"
Instruction Prompting
Generative Model: *G*
Conditional Generation
Generated Hypothesis: *h'*
"Sweat built up upon his face."
Hypothesis: *h*
Refine
No
M(h,h') = M(h',h) = entailment
Yes
Save
NLI Dataset (correctly answered pairs)
NLI Model: *M*
Predict M(h,h') and M(h',h)
"There were beads of perspiration on his brow"
Semantic Sensitivity
Predict M(p,h')
Semantically Robust
Yes
Premise: *p*
NLI Model: *M*
M(p, h') = M(p, h)
No
Predict M(p,h)
Random Guess

Figure 2.1: The proposed framework is comprised of two components. (i) a module for generating semantics-preserving surface-form hypothesis variations and (ii) using the generated surface for measuring semantic sensitivity and predictive inconsistency.

hypothesis (see Figure 2.1). These variations are created using conditional generation with Large Language Models (LLMs). We show that proposed candidates do not alter the core meaning or the truth value compared to the original statement. The original and generated sentences maintain denotative equivalence, where two sentences or phrases might be interpreted as having the same truth value or factual content but may carry minor variations of nuances or connotations. To ensure that the relations are preserved within the candidates during conditional generation, we assert that the NLI model predicts the original and generated hypothesis to symmetrically entail each other. This indicates that the model perceives both the generated and original hypothesis as equivalent. After introducing these variations, we evaluate the NLI model by replacing the original hypothesis with the generated candidates. As the candidates are indicated to be equivalent by the same NLI model, this evaluation will indicate whether the model can recover the existent relation between the premise hypothesis pair in the presence of minor semantic-preserving noise. We use the samples where the model identifies the existing relation correctly from the original premise hypothesis pair. This ensures that assessing for semantic sensitivity would not be hindered by the discrepancies in model performance.

We systematically study the semantic sensitivity across transformers that achieve state-of-the-art or similar results when trained on NLI datasets, namely RoBERTa (Liu et al., 2019c), BART (Lewis et al., 2019a), DeBERTa (He et al., 2020) and DistilBart (Sanh et al., 2019; Lewis et al., 2019a) with different parametrizations. To measure the effect of the phenomenon on the inconsistency of the predictions, we use three popular English datasets - MultiNLI (Williams et al., 2017, MNLI), SNLI (Bowman et al., 2015) and ANLI (Nie et al., 2019). The models are fine-tuned using MNLI, which we choose for *in-domain* testing, as it covers a wide range of topics and is frequently used for zero-shot and few-shot textual classification (Yin et al., 2019). We use the same models for *out-of-domain* evaluation across the other NLI datasets.

Our contributions are as follows: (i) we propose a novel framework for assessing semantic sensitivity within transformer-based language models (ii) we systematically study the influence of this phenomenon on inconsistent predictions across various transformer variants (iii) we show that the effect is persistent and pronounced across both *in-* and *out-of-domain* evaluations (iv) we further complete ablations to assess the severity of the inconsistent predictions caused by semantic sensitivity.

2.2 Related Work

Semantic comprehension is considered a fundamental building block for language understanding (Allen, 1995). Although attempts have been made to probe language models in terms of compositional semantic capabilities, the conclusions regarding their emergence remain to be discussed.

2.2.1 Models appear to understand semantics

Recently a wide suite of tasks has been proposed for testing models for language understanding (Wang et al., 2019a; Zellers et al., 2018; Ribeiro et al., 2020) with the credence that a model with strong performance should be able to utilise semantic relations when completing the tasks. In light of these, it has been shown that transformer-based language models can be directly trained (Zhang et al., 2020; Rosset et al., 2020) to utilise semantic structure to gain distributional information within the task. Specifically, NLI models have also been shown to be capable of pragmatic inferences (Jeretic et al., 2020a) with a perception of implicature (Grice, 1975) and presupposition (Stalnaker et al., 1977; Grice, 1975).

2.2.2 Models struggle with semantics

Directly probing for a specific aspect of semantic understanding has shown that transformer-based language models tend to struggle with semantics (Belinkov, 2022). It has been indicated that pretraining the language models does not exploit semantic information for entity labeling and coreference resolution (Liu et al., 2019b). Furthermore, transformer attention heads only minimally capture semantic relations (Kovaleva et al., 2019) from FrameNet (Baker et al., 1998). Studies have also shown that NLI models, in particular, tend to struggle with lexical variations, including word

	bart-l	roberta-l	distilbart	deberta-b	deberta-l	deberta-xl
MNLI$_{(n=10000)}$	90.10%	90.56%	87.17%	88.77%	91.32%	91.44%
SNLI$_{(n=10000)}$	87.55%	86.44%	84.37%	84.39%	88.87%	88.54%
ANLI_r1$_{(n=1000)}$	46.20%	46.40%	41.40%	35.10%	49.70%	53.00%
ANLI_r2$_{(n=1000)}$	31.60%	27.00%	32.80%	29.80%	32.70%	35.40%
ANLI_r3$_{(n=1200)}$	33.08%	26.75%	32.75%	30.50%	35.92%	38.75%

Table 2.1: The original accuracy on testing/dev sets for various transformers (b-base, l-large, xl-extra large) on *in-domain* MNLI experiments and zero-shot transfers to *out-of-domain* SNLI and ANLI. The number near the dataset name designates the exact amount of original samples in the testing set.

replacements (Glockner et al., 2018; Ivan Sanchez Carmona et al., 2018; Geiger et al., 2020), and sequence permutations (Sinha et al., 2021).

2.2.3 Sensitivity in NLI models

Probing NLI models for language understanding has been a hallmark testing ground for measuring their emerging capabilities (Naik et al., 2018a; Wang and Jiang, 2015; Williams et al., 2017). A wide range of tests indicates that models trained for NLI are prone to struggling with syntax and linguistic phenomena (Dasgupta et al., 2018; Naik et al., 2018b; An et al., 2019; Ravichander et al., 2019; Jeretic et al., 2020b). It has also been shown that NLI models heavily rely on lexical overlaps (Ivan Sanchez Carmona et al., 2018; McCoy et al., 2019b; Naik et al., 2018b) and are susceptible to over-attending to particular words for prediction (Gururangan et al., 2018a; Clark et al., 2019). Our line of work is associated with evaluating NLI models for monotonicity reasoning (Yanaka et al., 2019) and sensitivity towards specific semantic phenomenon (Richardson et al., 2020), such as boolean coordination, quantification, etc. However, we systematically test NLI models for their compositional semantic abilities and measuring the degree of inconsistence of their predictions influenced by the phenomenon.

2.3 Methodology

We aim to create a framework for assessing semantic sensitivity within NLI models and measure its impact on the inconsistence of model predictions. The first part of the pipeline we propose is an adversarial semantics-preserving generation for introducing variations within the original samples. The second part of the pipeline involves assessment using the acquired generations.

2.3.1 Semantics Preserving Surface-Form Variations

We formalise NLI as a pairwise input classification task. Given a dataset of premise hypothesis pairs $\mathcal{D} = (p_1, h_1), \dots (p_n, h_n)$, where $\forall p_i \in P \;\; \& \;\; h_i \in H$ are a set of textual tokens $P, H \; \subseteq \mathcal{T}$, the goal is to classify the pairs as *entailment, contradiction* or *neutrality*, i.e. $\mathcal{C} = \{E, C, N\}$. We are also given a pre-trained language model (PLM) $\mathcal{M}$ that is trained for textual entailment. Before introducing semantic variations, only the samples where model $\mathcal{M}$ predicted the label correctly are filtered, i.e. $D_{correct} = \{\forall (p_i, h_i) \in \mathcal{D} : \mathcal{M}(p_i, h_i) = \hat{y} = y\}$, where $\hat{y}$ is the prediction and y is the original label. This is completed to ensure that the evaluation of semantic sensitivity is not hindered or inflated by the predictive performance and confidence of the model $\mathcal{M}$. This type of filtering is used when probing for emergent syntactic (Sinha et al., 2021), lexical (Jeretic et al., 2020b), and numerical (Wallace et al., 2019) reasoning capabilities. We can see the original accuracy of NLI models and the number of samples used in the study in Table 2.1.

To introduce semantics preserving noise within chosen samples, we complete a two-fold refinement process. We utilise a generative LLM $\mathcal{G}$, which has been fine-tuned on natural language instructions (Wei et al., 2021; Chung et al., 2022), and prompt it to paraphrase the original hypothesis h_i, with the following prompt: *Rephrase the following sentence while preserving its original meaning:* $<h_i>$. This is not sufficient to produce semantics-preserving variations as generative models are prone to hallucinations (Ji et al., 2023) and not assured to produce an equivalent paraphrase. To ensure that the generation h_i' is logically equivalent to the original sample and thus semantics-preserving, we impose the condition that the NLI model should infer the relation between the original and generated hypothesis as a symmetric entailment:

$$\mathcal{M}(h_i, h_i') = \hat{y}_{\mathcal{C}=E} = \mathcal{M}(h_i', h) \tag{2.1}$$

The bidirectional nature of entailment allows us to claim that sentences are logically equivalent (Angell, 1989; Clark, 1967). We refine the proposed variation candidates using the generator $\mathcal{G}$ until k candidates that satisfy the condition are produced.

2.3.2 Human Evaluation of Surface-Form Variations

To further ensure the validity of this variation generation method, we conduct a human evaluation of the generated samples. We randomly sample 100 examples of generated and original hypothesis pairs across all datasets and employ two annotators to assess whether the sentences are semantically and logically equivalent within the pair. Our results show that in 99% of the cases, the annotators marked the samples as equivalent with an inter-annotator agreement measure of Cohen's $\kappa = 0.94$. This

r_s/r_r	bart-large	roberta-large	distilbart	deberta-base	deberta-large	deberta-xlarge
MNLI	6.64%/12.35%	5.71%/11.56%	9.20%/ **16.80%**	6.66%/13.81%	5.38%/11.54%	5.89%/11.49%
SNLI	10.11%/15.52%	8.38%/14.98%	15.67%/**23.68%**	9.96%/17.01%	7.83%/13.39%	9.50%/14.69%
ANLI_r1	31.51%/42.89%	28.45%/35.01%	31.48%/**52.30%**	40.0%/48.99%	25.66%/37.88%	22.71%/30.73%
ANLI_r2	34.39%/51.91%	24.62%/42.80%	36.09%/**57.49%**	34.92%/48.47%	28.44%/44.04%	29.46%/46.46%
ANLI_r3	29.11%/51.39%	21.88%/45.00%	29.26%/52.42%	33.88%/**53.17%**	24.88%/44.65%	23.23%/42.37%

Table 2.2: The strict and relaxed fooling rates of different transformer models across *in-domain* (MNLI) and *out-of-domain* (SNLI, ANLI) evaluations. On average more than half of the labels change towards their logically contrasting counterpart.

further shows the reliability of the method for generating semantics-preserving surface form variations. We provide further token overlap level analysis in section 2.7.

2.3.3 Evaluating Semantic Sensitivity

After obtaining k semantic variations for each hypothesis, we test the semantic sensitivity of the model by replacing the original hypothesis h_i with the candidates $\{h_i'^1, \dots h_i'^k\}$ and making a prediction with the NLI model $\mathcal{M}$. As the proposed variations are logically equivalent to the original, we want to test if the new model prediction would vary compared to the original.

$$\mathcal{R}(p_i, h_i, h_i'^j, \mathcal{O}) \quad = \quad \begin{cases} 1, \mathcal{O}(\mathcal{M}(p_i, h_i), \mathcal{M}(p_i, h_i'^j)) = 0 \\ 0, \mathcal{O}(\mathcal{M}(p_i, h_i), \mathcal{M}(p_i, h_i'^j)) = 1 \end{cases} \tag{2.2}$$

Here $\mathcal{O} : \mathcal{C} \times \mathcal{C} \rightarrow \{0, 1\}$ is a boolean matching operator between the labels predicted with original hypothesis h_i and the surface-form variations $h_i'^j$. A change in the label would imply that the model is semantically sensitive and the original correct prediction is inconsistent with the label produced for the semantics preserving surface-form variation. A graphical representation can be seen in Figure 2.5. We use two metrics to measure semantic sensitivity within NLI models, both of which are derivative formulations of a Fooling Rate (Moosavi-Dezfooli et al., 2017), which is used for assessing the success of adversarial attacks (Chakraborty et al., 2018). Given k possible surface-form variations for the hypothesis, we test if at least one of the candidates would be able to cause a label change compared to the original prediction, which can be formalised as:

$$r_r = \frac{\sum_i^{n'} \mathbb{1}\left[\exists j \in [1, k], \mathcal{R}(p_i, h_i, h_i'^j, =) \neq 1\right]}{n'}. \tag{2.3}$$

Here n' is the number of correctly answered original samples, and the matching operator $\mathcal{O}$ is a simple equality checking operator "=". We refer to this metric as a relaxed Fooling Rate. To measure more drastic label changes, i.e. *entailment* to *contradiction* and vice versa, we also define a stricter version of Equation 2.3.

$$r_s = \frac{\sum_i^{n'} \mathbb{1}\left[\exists j \in [1,k], \mathcal{R}(p_i, h_i, h_i'^j, =^s) \neq 1\right]}{n'}. \tag{2.4}$$

We replace standard equality for the operator $\mathcal{O}$ in Equation 2.3 with a strict counterpart that matches only if the predictions are direct opposites, i.e. *entailment* ↔ *contradiction*. It must be noted that the *neutral* class does not have a direct opposite; thus, the metric for this label remains unchanged. It can be concluded that the inequality $r_s \leq r_r \leq 1$ trivially holds when using these metrics.

2.4 Experimental Setup

2.4.1 Model Details

2.4.2 Semantics preserving Generation

To generate and refine semantic variations of the original hypothesis, we chose *flan-t5-xl* as the generation model $\mathcal{G}$. It is an instruction-tuned LLM that has shown close state-of-the-art performance in tasks such as paraphrasing, zero and few shot generation, chain of thought reasoning (CoT), and multi-task language understanding (Chung et al., 2022). For each of the selected hypotheses, we produce $k = 5$ unique semantics-preserving variations. To ensure diversity and consistency of the generated text while avoiding computationally expensive exhaustive search, we use a group beam search (Vijayakumar et al., 2016) with a temperature $t \in [0.3, 0.6]$ and a maximum output of 40 tokens throughout the generation and refinement procedure. We also further diversify the generation by using the recipe from Li et al. (2016).

2.4.3 NLI models

We systematically experiment with transformer architectures that are fine-tuned on MNLI, which exhibit state-of-the-art or close predictive accuracy on the dataset. We specifically choose *bart-large* (Lewis et al., 2019a), *roberta-large* (Liu et al., 2019c), *deberta-base, deberta-large, deberta-xlarge* (He et al., 2020) and *distilbart* (Sanh et al., 2019). These PLMs are taken without change from their original studies through the Transformers library (Wolf et al., 2020), ensuring the complete reproducibility of the

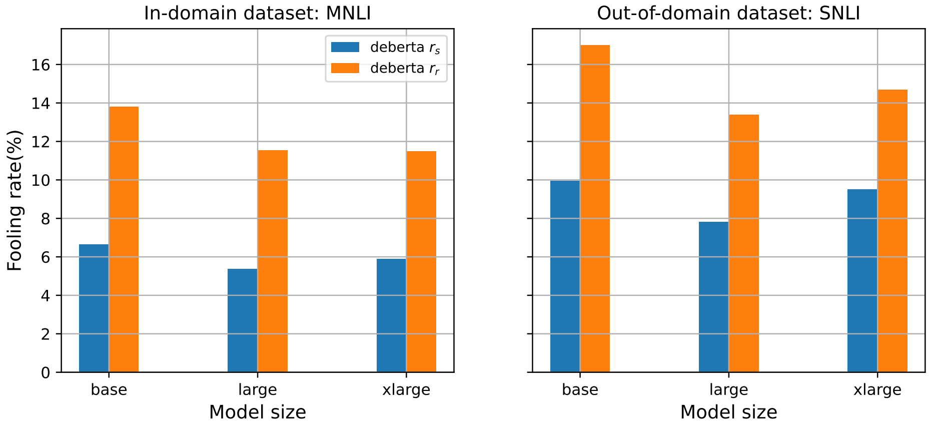


Figure 2.2: In- and out-of-domain fooling rate of DeBERTa of varied sizes, which are measured on MNLI (left) and SNLI (right). Similarly, r_s and r_r represent the strict and relaxed fooling rates, respectively.

results. To observe the effect in an *out-of-domain* setup, we also evaluate these models on SNLI and ANLI in a zero-shot transfer setting.

2.5 Results and Analysis

This section presents the results and analyses of our semantic sensitivity evaluation framework along with a suite of ablations analysing the phenomenon across various transformer sizes, domains, and label space. Furthermore, we measure the impact of the phenomenon on the inconsistent predictive behaviour of NLI models.

2.5.1 Semantic Sensitivity

2.5.2 In-domain

We evaluate several PLMs trained on MNLI using our experiments presented in Table 2.2. The results show that models are limited in their comprehension of compositional semantics as the relaxed fooling rate on *in-domain* experimentation averages at $r_r = 12.9\%$. This is further reinforced by the fact that more than half, $r_s = 6.58\%$ of the label changes occur with strict inequality. This means that minor semantics-preserving changes lead to a sizable shift in model predictions, even prompting towards the opposite decision edge half the time. The behaviour is consistent across all the transformers and leads us to believe that samples that changed labels after surface-form variations showcase the inconsistent predictive nature of the models. We further elaborate on this in the next section. Consequently, semantically equivalent variations evidently hinder the decision-making of the NLI models, prompting us to believe that models have limited understanding w.r.t. semantic structure and logical relation, even when the model is trained on texts from the same distribution.

2.5.3 Out-of-domain

We also probe the NLI models in an *out-of-domain* zero-shot setting to assess the transferability of compositional semantic knowledge. Our results in Table 2.2 show that the discrepancies and limitations in semantic comprehension are even more pronounced in this setting. We see an averaged relaxed fooling rate of $r_r = 23.7\%$, with the maximum at 57.49%, which is only marginally better than a majority voting baseline. It must be noted that because different datasets have varying numbers of samples, the average is weighted w.r.t. the number of sampled instances from the particular dataset in the experiment. The results on *out-of-domain* evaluation once again follow the pattern that more than half, $r_s = 15.8\%$ of the samples switch the labels to their logically contrasting counterparts. This shows that zero-shot transfer further amplifies the limitations that NLI models have for using semantic structures and preserving logical relations. This further suggests that the semantic variations where a label change occurs are likely to be originally predicted correctly as an inconsistent guess. It follows, that although PLMs fine-tuned on MNLI are widely used for zero-shot classification, their effectiveness diminishes if the classification tasks require syntactic understanding. Indeed, model effectiveness declines and the fooling rates rise as the tasks become more challenging, requiring greater syntactic knowledge, as we can see from the comparison of the results from SNLI to ANLI.

2.5.4 Effects of distillation

Next, we want to probe if the susceptibility towards semantic noise is transferred during model distillation. Thus, we use *DistilBart* that is distilled from a larger pre-trained BART model. While model accuracy remains comparable to the original model in Table 2.1, the distilled version struggles sizeably more with surface-form variations. On average, across *in-* and *out-of-* domain evaluation, the distilled NLI model is more sensitive than the original in terms of relaxed fooling rate by $\triangle r_r = 18.4\%$. The effect of supposed inconsistence is amplified when observing the strict fooling rate, where on average $\frac{r_r}{r_s} \leq 1.5$. This indicates that during distillation, models are bound to forget the knowledge regarding compositional semantics making it harder to preserve the logical equivalence during inference.

2.5.5 Effects of model size

We also test how semantics-preserving noise affects models of different sizes and parametrization (see Figure 2.2). Although for *in-domain* setup, the relaxed fooling rate metrics marginally drop as the models get bigger, the same cannot be observed in *out-of-domain* setup. It is evident that bigger PLMs from our study are almost as restricted in semantic comprehension as their smaller counterparts. This indicates that

	$r_s/r_r\,(y=E)$	$r_s/r_r\,(y=N)$	$r_s/r_r\,(y=C)$	r_s/r_r
MNLI	2.78%/13.41%	14.33%/14.33%	3.69%/11.17%	6.58%/12.92%
SNLI	9.54%/18.73%	19.42%/19.42%	2.92%/11.82%	10.24%/16.54%
ANLI_r1	21.64%/41.97%	38.62%/38.62%	29.17%/44.57%	29.97%/41.30%
ANLI_r2	20.84%/46.28%	49.41%/49.41%	21.89%/50.80%	31.32%/48.53%
ANLI_r3	11.65%/52.00%	47.18%/47.18%	16.42%/46.50%	27.04%/48.17%

Table 2.3: Fooling rate averaged over all models. r_s represents the strict fooling rate, in which case the predicted label of the evaluation pair is opposite to the original label y. r_r measures the proportion of label change. $y \in \{E, N, C\}$ group the (p, h) pairs by their semantic relation, representing entailment, neutrality, and contradiction, respectively.

emergent semantic capabilities are not only tied to model size, but also widely depend upon the choice of the training dataset.

2.5.6 Severity of Inconsistent Predictions

2.5.7 Consistency across label space

To analyse the extent of semantic sensitivities within NLI models we test the effect across all the classes in the label spaces, presented in Table 2.3. The per-class breakdown of the strict and relaxed fooling rate indicates that the effect is consistent across the whole label space. This allows us to conclude that the observed limitations in compositional semantic understanding are not caused by class imbalances and are not specific to a particular set of examples. We see the increased fooling rate across all of the labels when comparing *in-domain* and *out-of-domain* experiments. This reinforces the prior indications regarding models' inability to use semantic structure to preserve inherent relations within the data, as all logical relations attain rather similar amounts of fooling rate during direct evaluation.

2.5.8 Distribution shift in decision making

Recall that we want to measure the impact of semantics-preserving surface-form variations on NLI models. We study the predictive distributional shift within the samples that cause a changed model prediction. To do this, we initially split the samples into two categories considering whether the sample induced a change of the original prediction within the NLI model. We further average the probability distribution of labels obtained from the final softmax layer of the model for these two categories. We measure the differences between the two distributions with two statistical tests. To evaluate the relative entropy between them, we use Jensen-Shanon Divergence (Fuglede and Topsoe, 2004), a symmetric, non-negative, and

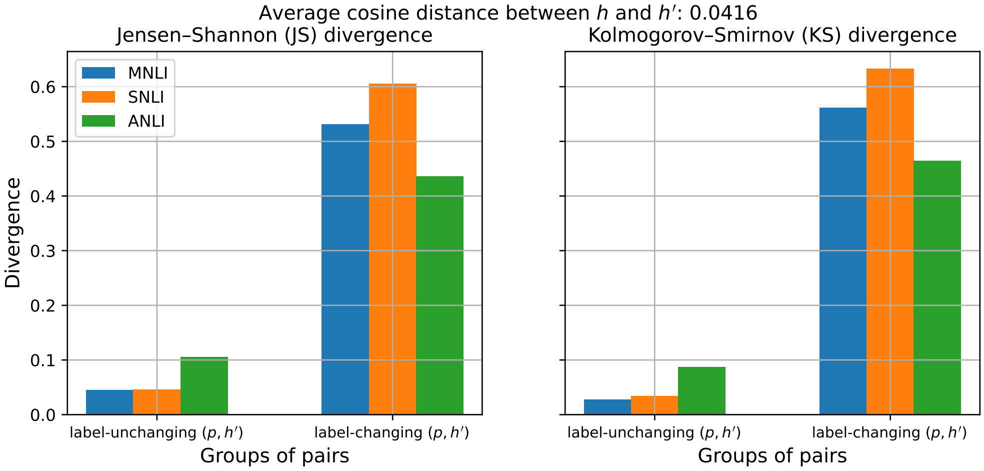


Figure 2.3: Divergence of predictive probability distribution between (p, h) and (p, h') measured across the datasets (ANLI is averaged over the rounds) and averaged over all models. All evaluation pairs are split into two groups based on whether they manage to flip the original label. Two divergence metrics are shown – JS divergence (left) and KS divergence (right).

bounded metric for assessing the similarity between two distributions, $\mathrm{JSD}(P\|Q) = \frac{1}{2}D(P\|M) + \frac{1}{2}D(Q\|M)$, where D is the Kullback–Leibler divergence (Joyce, 2011). We verify the statistical significance of our findings with the Kolmogorov–Smirnov test (Berger and Zhou, 2014), which shows if the two sets of samples are likely to come from the same distribution.

Our results in Figure 2.3 show a significant distribution shift when assessing semantics-preserving surface-form variations. The cosine distance in the sentence embedding space between the generated and original samples is negligible at 0.04. As the absolute cosine similarity values possess limited interpretable meaning, we further explore the distributions of cosine distances towards original samples for the examples that do and do not induce label changes. We measure the Jansen-Shannon divergence of these two distributions at 0.001, implying they are strongly similar. This reinforces the hypothesis that surface-form variations produce logically equivalent samples with minor distance in the embedding space regardless of the induced label changes. However, despite minor changes in the semantic composition, we see a sizable change in the final predictive distribution of the NLI models. We see a significant rise both in Jensen-Shannon divergence and Kalmogorov-Smirnov metric, $\triangle$JSD $= 0.51$ and $\triangle$K-S $= 0.54$, when comparing the examples where the model prediction has changed compared to the original. This indicates that the generated variations do not cause negligible change within model prediction, but rather can be considered adversarial for the model. It shows that the limited capabilities to utilise syntactic information cause the model to significantly change the final prediction given

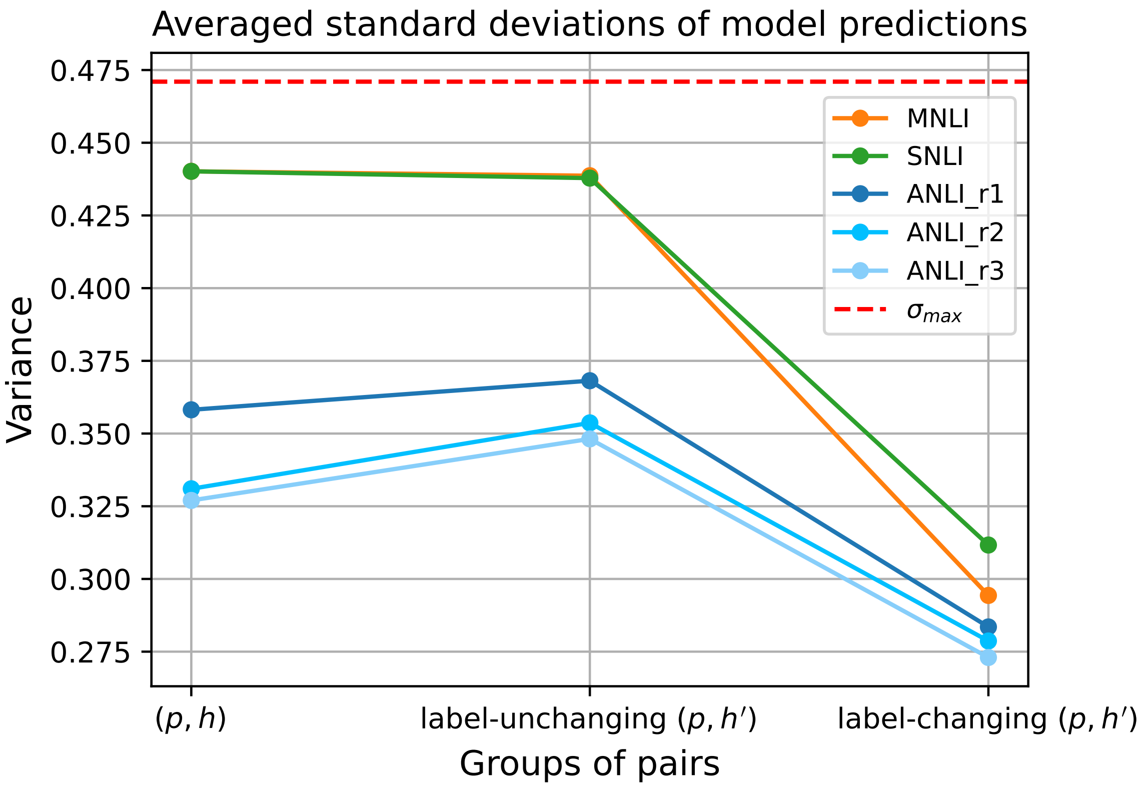


Figure 2.4: Standard deviation σ of predicted label probabilities (obtained from the final softmax layer of the model) averaged for original premise-hypothesis pair (left), surface-form variations that did not cause label changes (mid) and did induce label change (right). The bigger σ, the more confident the model is w.r.t. the predictions. The results are averaged over all models.

minuscule variations, which is an inconsistent predictive behaviour. Given that we initially sampled examples that the models answered correctly, these results assert our belief that the models do not display consistent predictive behaviour despite having equivalent inputs. This shows that albeit the strong model performance presented in Table 2.1, there is masked degeneration and discrepancies within the NLI models stemming from semantic sensitivity. Our method allows for explicitly quantifying the degree of semantic sensitivity within PLMs and allows to measure the impact of that sensitivity on the decision-making process of the model.

2.5.9 Semantic-Sensitivity and decision variations

We lastly analyse the standard deviation within the predicted label distribution produced from the softmax of the model. We compute the standard deviation for the distribution of original premise hypothesis predictions and compare it with a replacement that does not and does cause label changes in PLM classification, see Figure 2.4. For reference, the upper bound for standard deviation in this 3 class setting happens when the model is greatly confident in one of the classes, i.e. $softamx = [1, 0, 0] \rightarrow \sigma_{max} = 0.471$. Bigger σ on average implies more confident answers by the PLM. It can be observed that the average predictions with the original samples have a great degree of confidence. We see an interesting phenomenon where the predictive confidence slightly rises across most of the datasets for the cases where the model is able to recover the inherent textual relations. However, when faced with examples that cause label changes, there is a significant drop of $\triangle\sigma = 0.1$ in the standard deviation averaged across the datasets. This signifies that predictive confidence sizably degrades when the model struggles to recover the existent relations because of slight semantics-preserving variations. That further indicates that NLI models are susceptible to semantic sensitivity and have limited knowledge of compositional semantics, which can lead to the degradation of predictive confidence and incidentally inconsistent predictions.

2.6 Conclusion

We present a novel framework for assessing semantic sensitivity in NLI models through generating semantics-preserving variations. Our systematic study of the phenomenon across various datasets and transformer-based PLMs shows that the models consistently struggle with variations requiring knowledge of compositional semantics. This performance deterioration happens across the whole label space, almost regardless of model size. We measure the impact of semantic-sensitivity and show that it diminishes models' predictive confidence and can lead to predictive inconsistency.

Limitations

In our work, we cover the semantic-sensitivity that can be found within NLI models. However, the framework can be applied to a wider range of classification tasks. The benchmark can be extended with more datasets and further enhanced with larger human evaluation. Also, we covered PLMs specifically trained for NLI; however, it would be great to cover bigger LLMs, in particular w.r.t. their emergent zero-shot capabilities. Another limitation is that we only cover English-based language models and do not test in multi-lingual or cross-lingual settings.

Ethics Statement

Our work completes an analysis of numerous models w.r.t. their decision inconsistency induced by semantic surface form variations. We show that models are somewhat unable to handle logically and semantically equivalent sentences, which would lead to an inconsistent use across various domains and applications. Our generation method does not induce any further exploitation threat and can only be used for measuring the above-mentioned inconsistencies. We exclusively use open source publicly accessible data and models within our experimentations.

Acknowledgements

Erik is partially funded by a DFF Sapere Aude research leader grant under grant agreement No 0171-00034B, as well as by a NEC PhD fellowship. This work is further supported by the Pioneer Centre for AI, DNRF grant number P1.

Dataset	Fuzzy token match %	average length h	average length $h\prime$	average token overlap
mnli	84.83	14.31	14.14	13.25
snli	81.55	10.81	11.21	10.38
anli_r1	87.59	17.3	17.02	13.73
anli_r2	86.49	15.99	15.84	12.8
anli_r3	85.17	14.32	14.29	11.27

Table 2.4: Percentages of token matches and other statistics.

2.7 Appendices

2.7.1 Evaluation under Label change

To assess the extent of the impact of semantic sensitivity, we employ an evaluation under label change. This means we consider the examples that changed the original prediction of the model after a surface-form variation replaced the original hypothesis. A graphical representation of this can be seen in Figure 2.5. It must be noted that we use only the samples that the model originally predicted correctly to avoid incorrect assessment regarding the reasoning behind the false predictions. Our primary aim is to measure the semantic sensitivity within the model predictions and the extent of inconsistency it causes.

2.7.2 Token Level-Differences of the generated variations

We further explore the difference between surface-form variations and original examples by conducting a token-level analysis for each pair $(h, h\prime)$. We compute the average amount of tokens present for the original and generated hypothesis and use fuzzy and exact matching to assess the overlap of tokens on average for each dataset. The results can be seen in Table 2.4. The results show that the generated and original examples have a high token level overlap which further reinforces the idea that surface form variations are close both syntactically, in the embedding space and logically.

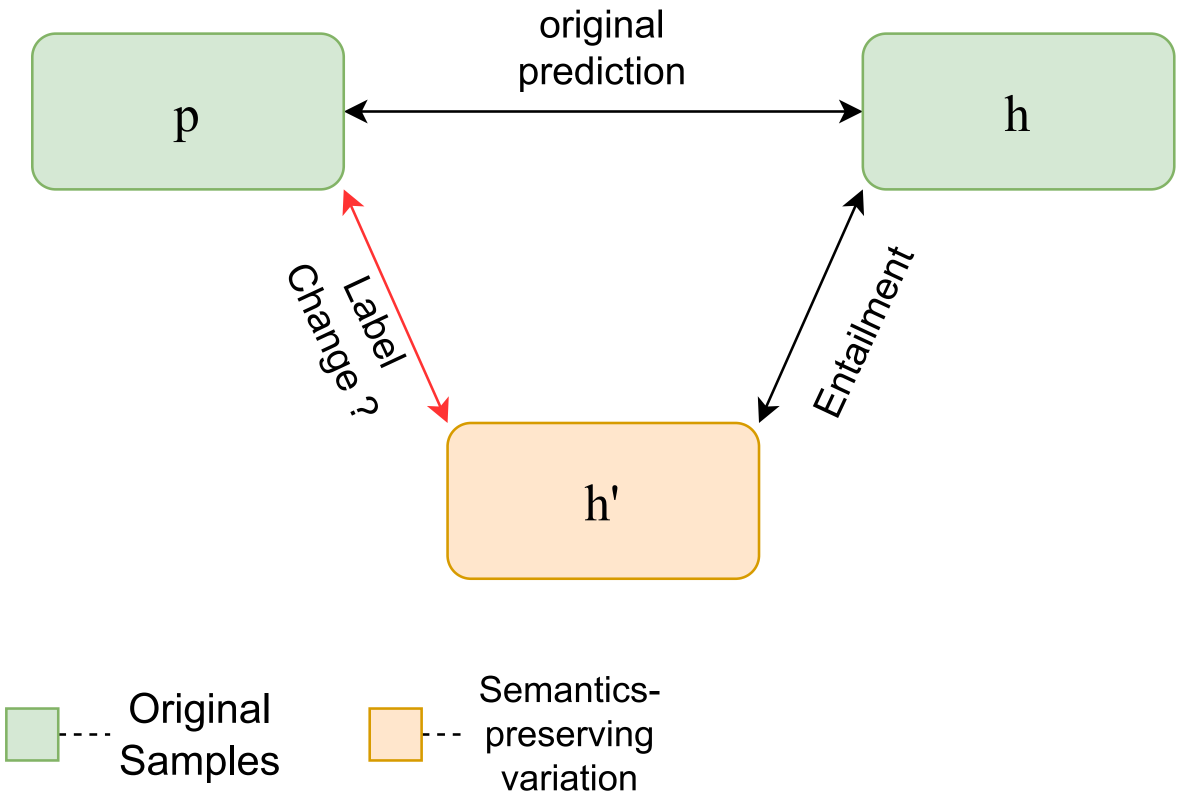


Figure 2.5: A diagram for assessing semantic similarity. Given the generated semantics-preserving surface-form variation h', we evaluate if a label change occurs when replacing the hypothesis in accordance with Equation 2.1

.

With Great Backbones Comes Great Adversarial Transferability

3

With Great Backbones Comes Great Adversarial Transferability

3.1 Introduction

Machine vision models pre-trained with massive amounts of data and using self-supervised techniques (Newell and Deng, 2020) are shown to be robust and highly performing(Goyal et al., 2021a; Goldblum et al., 2024) feature-extracting backbones (Elharrouss et al., 2022; Han et al., 2022), which are further used in a variety of tasks, from classification (Atito et al., 2021; Chen et al., 2020b) to semantic segmentation (Ziegler and Asano, 2022). However, creating such backbones incurs substantial data annotation (Jing and Tian, 2020) and computational costs (Han et al., 2022), consequently rendering the use of such publicly available pre-trained backbones the most common and efficient solution for researchers and engineers alike. Prior works have focused on analysing safety and adversarial robustness with complete, i.e. *white-box* (Porkodi et al., 2018) or no, i.e. *black-box* (Bhambri et al., 2019) knowledge of the target model weights, fine-tuning data, fine-tuning techniques and other tuning meta-information. Although, in practice, an attacker can access partial knowledge (Lord et al., 2022; Zhu et al., 2022a; Carlini et al., 2022) of how the targeted model was produced, i.e. original backbone weights, tuning recipe, etc., the adversarial robustness of models tuned on a downstream task from a given pre-trained backbone remains largely underexplored. We refer to settings with partial knowledge of target model constructions meta-information as *grey-box*. This is important both for research and production settings because with an increased usage (Goldblum et al., 2023) of publically available pre-trained backbones for downstream applications, we are incapable of assessing the potential exploitation susceptibility and inherent risks within models tuned on top of them and subsequently enhance future pre-trained backbone sharing practices.

In this work, we systematically explore the safety towards adversarial attacks within the models tuned on a downstream classification task from a known publically available backbone pre-trained with a self-supervised objective. We further explicitly measure the effect of the target model construction meta-information by simulating

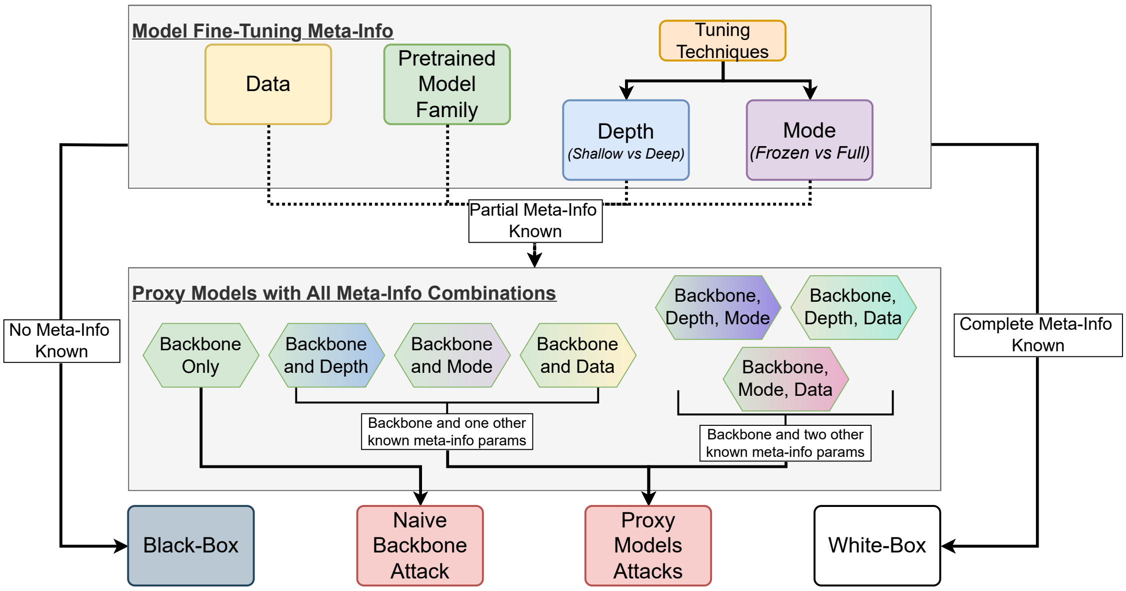


Figure 3.1: The figure depicts all of the settings used to evaluate adversarial vulnerabilities given different information of the target model construction. From left to right, we simulate exhaustive varying combinations of meta-information available about the target model during adversarial attack construction. All of the created proxy models are used separately to assess adversarial transferability.

different levels of its availability during the adversarial attack. For this purpose, we initially train 352 diverse models from 21 families of commonly used pre-trained backbones using 4 different fine-tuning techniques and 4 datasets. We fix each of these networks as a potential target model and transfer adversarial attacks using all of the other models produced from the same backbones as proxy surrogates (Qin et al., 2023; Lord et al., 2022) for adversarial attack construction. Each surrogate model simulates varying levels of knowledge availability w.r.t. target model construction on top of the available backbone during adversarial attack construction. This constitutes approximately 20000 adversarial transferability comparisons between target and proxy pairs across all model families and meta-information variations. By assessing the adversarial transferability of attacks from these surrogate models, we are able to explicitly measure the impact of the availability of each meta-information combination about the final target model during adversarial sample generation.

We further introduce a naive exploitation method referred to as *backbone attacks* that utilizes only the pre-trained feature extractor for adversarial sample construction. The attack uses projected gradient descent over the representation space to disentangle the features of similar examples. Our results show that both proxy models and even simplistic *backbone attacks* are capable of surpassing strong query-based *black-box* methods and closing to *white-box* performance. The findings indicate that *backbone at-*

tacks, where the attacker lacks meta-information about the target model, are generally more effective than attempts to generate adversarial samples with limited knowledge. This highlights the vulnerability of models built on publicly available backbones.

Our ablations show that *having access to the weights of the pre-trained backbone is functionally equivalent to possessing all other meta-information about the target model when performing adversarial attacks*. We compare these two scenarios and show that both lead to similar vulnerabilities, highlighting the interchangeable nature of these knowledge types in attack effectiveness. Our results emphasize the risks in sharing and deploying pre-trained backbones, particularly concerning the disclosure of meta-information. Our experimental framework can be seen in fig. 3.1.

Toward this end, our contributions are as follows:

- We introduce, formalize and systematically study the **grey-box** adversarial setting, which reflects realistic scenarios where attackers have partial knowledge of target model construction, such as access to pre-trained backbone weights and/or fine-tuning meta-information.

- We simulate over $20,000$ adversarial transferability comparisons, evaluating the impact of varying levels of meta-information availability about target models during attack construction.

- We propose a naive attack method, *backbone attacks*, which leverages the pre-trained backbone's representation space for adversarial sample generation, demonstrating that even such a simplistic approach can achieve stronger performance compared to a query-based black-box method and often approaches white-box attack effectiveness.

- We show that access to pre-trained backbone weights alone enables adversarial attacks as effectively as access to the full meta-information about the target model, emphasizing the inherent vulnerabilities in publicly available pre-trained backbones.

3.2 Related Work

3.2.1 Self Supervised Learning

With the emergence of massive unannotated datasets in machine vision, such as YFCC100M(Thomee et al., 2016), ImageNet(Deng et al., 2009), CIFAR (Krizhevsky et al., 2009) and others Self Supervised Learning (SSL) techniques (Jing and Tian, 2021) became increasingly more popular for pre-training the models (Newell and Deng, 2020). This prompted the creation of various families of SSL objectives, such as colorization prediction (Zhang et al., 2016), jigsaw puzzle solving (Noroozi and Favaro, 2016) with further invariance constraints (Misra and van der Maaten, 2020, PIRL), non-parametric instance discrimination (Wu et al., 2018, NPID, NPID++), unsupervised clustering (Caron et al., 2018), rotation prediction (Gidaris et al., 2018, RotNet), sample clustering with cluster assignment constraints(Caron et al., 2020, SwAV), contrastive representation entanglement (Chen et al., 2020a, SimCLR), self-distillation without labels (Caron et al., 2021, DINO) and others (Jing and Tian, 2021). Numerous architectures, like AlexNet (Krizhevsky et al., 2012), variants of ResNet(He et al., 2016) and visual transformers (Dosovitskiy et al., 2021; Touvron et al., 2021; Ali et al., 2021) were trained using these SSL methods and shared for public use, thus forming the set of widely used pre-trained backbones. We obtain all of these models trained with different self-supervised objectives from their original designated studies summarised in VISSL (Goyal et al., 2021b). An exhaustive list of all models can be seen in table 3.1.

3.2.2 Adversarial Attacks

The availability of pre-trained backbones allows to test them for vulnerabilities towards adversarial attacks, which are learnable imperceptible perturbations generated to mislead models into making incorrect predictions (Szegedy et al., 2014; Goodfellow et al., 2015). Several attack strategies have been studied, including single-step fast gradient descent (Goodfellow et al., 2014; Kurakin et al., 2017, FGSM), and computationally more expensive optimization-based attacks, such as projected gradient descent based attacks (Madry et al., 2018, PGD), CW (Carlini and Wagner, 2017), JSMA (Papernot et al., 2017), and others (Dong et al., 2018; Moosavi-Dezfooli et al., 2016; Madry et al., 2018). All of these attacks assume complete access to the target model, which is known as the *white-box* (Papernot et al., 2017) setting. These attacks can be *targeted* toward confusing the model to infer a specific wrong class or *untargeted* with the desire that it infers any incorrect label. However, an opposite setting with no information, referred to as *black-box* (Papernot et al., 2017), has also been explored as a more practical setting. The methods involve attempts at gradient estimation (Chen et al., 2017b; Ilyas et al., 2018; Bhagoji et al., 2018), adversarial transferability (Papernot et al., 2017; Chen et al., 2020c), local search (Narodytska and Kasiviswanathan, 2016; Brendel et al., 2018; Li et al., 2019; Moon et al., 2019), combinatorial perturbations (Moon et al., 2019) and others (Bhambri et al., 2019). However, these methods also require massive sample query budgets ranging from $[10^3, 10^5]$ queries or computational resources creating each adversarial sample (Bhambri et al., 2019). Compared to these, we introduce a novel setup with the knowledge of the pre-trained backbone and varying levels of partially known target model tuning meta-information during adversarial attack construction, which we call *grey-box*. We show that even simple naive attacks are capable of exploiting better than black-box attacks without the need for significantly querying the target model.

3.2.3 Adversarial Transferability

Our work is also aligned with adversarial transferability, where adversarial examples generated for one model can mislead other models, even without access to the target model weights or training data. This property poses significant security concerns, as it allows for effective black-box attacks on systems with no direct access (Papernot et al., 2017; Ilyas et al., 2018). Efforts can be divided into *generation-based* and *optimisation* methods. Generative methods have emerged as an alternative approach to iterative attacks, where adversarial generators are trained to produce transferable perturbations. For instance, Poursaeed et al. (2018) employed autoencoders trained on white-box models to generate adversarial examples. Most of the attacks aiming for adversarial transferability strongly depend on the availability of data from the target domain (Carlini and Wagner, 2017; Papernot et al., 2017). However, although current adversarial transferability methods claim to produce massive vulnerabilities in machine vision models, Katzir and Elovici (2021) examines the practical implications of adversarial transferability, which are frequently overstated. That study demonstrates that it is nearly impossible to reliably predict whether a specific adversarial example will transfer to an unseen target model in a black-box setting. This perspective underscores the importance of systematically evaluating transferability in realistic settings, including scenarios where attackers are sensitive to the cost of failed attempts. In our study, we offer a novel systematic approach to explicitly assess the adversarial transferability with varying levels of meta-information knowledge.

3.3 Methodology

3.3.1 Preliminaries

For consistency, we employ the following notation. We denote each Dataset $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\}$. Where $\mathcal{X} = \{x_1, \ldots, x_{|\mathcal{D}|}\}$ is a set of images, with $x_i \in \mathcal{R}^{H \times W \times C}$, where H,W and C are the height, width and the channels of the image accordingly and $\mathcal{Y} = \{y_1 \ldots y_n\}$ is used as the set of ground truth labels. We denote the training, validation and testing splits per task as $\mathcal{D} = \{\mathcal{D}_{train}, \mathcal{D}_{val}, \mathcal{D}_{test}\}$. A *model* is defined as the following tuple $\mathcal{M} = \mathcal{M}(\mathcal{D}, \mathcal{W}, \mathcal{B}, \mathcal{F})$, where $\mathcal{D}$ contains the dataset used for training, $\mathcal{W}$ are the weights of the trained model and $\mathcal{B}$ is the pre-trained back-bone $\mathcal{B}(\mathcal{W}_{\mathcal{B}})$ with available weights $\mathcal{W}_{\mathcal{B}}$. The notation $\mathcal{F}(\mathcal{T}, \mathcal{Z})$, where $\mathcal{T}$ encodes the *mode* of tuning (e.g., full fine-tuning, partial fine-tuning, etc.) and $\mathcal{Z}$ the *depth* of tuning of the final classifier on top of the backbone.

3.3.2 Meta-Information variations

We define the variations of the available meta-information about the target model $\mathcal{M}$ during an adversarial attack as a *unit of release* $\mathcal{R} = \mathcal{R}(\mathcal{M}(\mathcal{D}, \mathcal{W}, \mathcal{B}(\mathcal{W}_{\mathcal{B}}), \mathcal{F}(\mathcal{T}, \mathcal{Z})))$. For example, if the target fine-tuning mode $\mathcal{Z}^{target}$ and dataset $\mathcal{D}^{target}$ are not known, the unit of release will be $\mathcal{R} = \mathcal{R}(\mathcal{M}(*, \mathcal{W}, \mathcal{B}(\mathcal{W}_{\mathcal{B}}), \mathcal{F}(\mathcal{T}, *)))$. Note that the *black-box* setting will correspond to the unit of release $\mathcal{R}(\mathcal{M}(*, *, *, *, *))$ and the *white-box* setting to $\mathcal{R}(\mathcal{M}(\mathcal{D}, \mathcal{W}, \mathcal{B}(\mathcal{W}_{\mathcal{B}}), \mathcal{F}(\mathcal{T}, \mathcal{Z})))$, all the variations between these are considered *grey-box*. When discussing any experiments within the *gery-box* setup, we assume the minimal unit of release contains knowledge about at least the pre-trained backbone i.e. $\mathcal{R}(\mathcal{M}(*, *, \mathcal{B}(\mathcal{W}_{\mathcal{B}}), *)$.

3.3.3 Adversarial Attacks with Proxy Models

To test the adversarial robustness of the models trained from the same pre-trained backbone, we create a set of proxy models $\mathcal{M}^{proxy} = \{\mathcal{M}_1^{proxy} \dots \mathcal{M}_v^{proxy}\}$ given the pre-trained backbone $\mathcal{B}$, where v is the number of all possible units of release between *black-box* and *white-box* settings that include the backbone. For each proxy model $\mathcal{M}_i^{proxy}$ with its designated meta-information unit of release $\mathcal{R}_i$, we use an adversarial attack $\mathcal{A}$ to generate adversarial noise and further transfer it to the target model $\mathcal{M}^{target}$. This means that given an example image x with a label y, target and proxy models $\mathcal{M}^{target}$, $\mathcal{M}^{proxy}$ we want to produce a sample x' that would fool the target model, such that $\arg\max \mathcal{M}^{target}(x') \neq y$. If we are using a targeted attack then we want $\mathcal{M}^{target}(x') = t$ where t is the targeted class different from the ground truth $t \neq c_{gt}$. After creating the adversarial attack for each sample in $\mathcal{D}_{test}^{proxy}$ and $\mathcal{D}_{test}^{target}$ we evaluate the success rate of the attack and the success rate of the transferability onto the target model. To measure the success and robustness of the adversarial attack and its transferability, we define the following metrics:

- **Attack Success Rate (ASR):** This is the proportion of adversarial examples successfully fooling the proxy model $\mathcal{M}_i^{proxy}$, defined as:

$$\text{ASR}_i = \frac{1}{|\mathcal{D}_{test}^{proxy}|} \sum_{x \in \mathcal{D}_{test}^{proxy}} \mathbb{I}\left[\arg\max \mathcal{M}_i^{proxy}(x') \neq y\right], \tag{3.1}$$

  where $\mathbb{I}[\cdot]$ is the indicator function.

Algorithm 1 Backbone Attack

```
Input: Model backbone $\mathcal{B}$, clean image $x_0$, perturbation bound $\epsilon$, step size $\alpha$, number
       of steps $T$, distance function $\mathcal{L}_{\text{cosine}}$, random start flag
Output: Adversarial image $x_{\text{adv}}$
Initialization:
 $x_{\text{adv}} \leftarrow x_0$
if random start then
    $x_{\text{adv}} \leftarrow x_{\text{adv}} + \text{Uniform}(-\epsilon, \epsilon)$
      $x_{\text{adv}} \leftarrow \text{Clip}(x_{\text{adv}}, 0, 1)$
end
Fixed Original Image Representation:
 $z_0 \leftarrow StopGrad(\mathcal{B}(x_0))$

for $t = 1$ to $T$ do
    Forward Pass:
        $z_{\text{adv}} \leftarrow \mathcal{B}(x_{\text{adv}})$ // Adversarial image representation
    Compute Loss and Gradient:
        $\mathcal{L} \leftarrow 1 - \cos(z_{\text{adv}}, z_0)$ // Distance loss
    $g \leftarrow \nabla_{x_{\text{adv}}} \mathcal{L}$ // Gradient w.r.t $x_{\text{adv}}$
    Update Adversarial Image:
        $x_{\text{adv}} \leftarrow x_{\text{adv}} + \alpha \cdot \text{sign}(g)$ // PGD step
    Projection:
        $\delta \leftarrow \text{Clip}(x_{\text{adv}} - x_0, -\epsilon, \epsilon)$ // Project perturbation into $\ell_\infty$-ball
    $x_{\text{adv}} \leftarrow \text{Clip}(x_0 + \delta, 0, 1)$ // pixel range
end
return $x_{\text{adv}}$
```

- **Transfer Success Rate (TSR):** To evaluate the transferability of adversarial examples generated using the proxy model $\mathcal{M}_i^{proxy}$ to the target model $\mathcal{M}^{target}$, we compute the fooling rate on the target model as:

$$\text{TSR}_i = \frac{1}{|\mathcal{D}_{test}^{target}|} \sum_{x \in \mathcal{D}_{test}^{target}} \mathbb{I}\left[\arg\max \mathcal{M}^{target}(x') \neq y\right]. \tag{3.2}$$

This setup allows us to explicitly quantify how the availability of diverse meta-information combinations explicitly impacts the adversarial transferability of the given model, thus highlighting the risks in the model-sharing practices. A visual depiction of this can be seen in fig. 3.1.

3.3.4 Backbone Attack

To test the vulnerabilities associated with publicly available pre-trained feature extractors, we designed a naive *backbone attack*, which only utilises the known backbone $\mathcal{B}$ of the model $\mathcal{M}^{target}$. The aim, similar to the prior paragraph, is to create an adversarial attack from the $\mathcal{B}$ to transfer towards the target model $\mathcal{M}^{target}$. To do this, we utilise a Projected Gradient Descent (, PGD)-based method, where the attack iteratively perturbs the input images in order to maximise the distance between the feature representations of the clean input and the adversarial input, as derived from the backbone $\mathcal{B}$. More formally, let x and $\tilde{x}$ represent the clean input and adversarial input, respectively. The attack iteratively refines $\tilde{x}$ such that:

$$\tilde{x}_{t+1} = \text{Proj}_{\mathcal{S}} \left(\tilde{x}_t + \alpha \cdot \text{sign} \left(\nabla_{\tilde{x}_t} \mathcal{L}_{\mathcal{B}}(x, \tilde{x}_t) \right) \right), \tag{3.3}$$

where $\mathcal{L}_{\mathcal{B}}$ is the loss function defined to measure the distance between the feature representations of the clean and adversarial inputs. The backbone representations $f_{\mathcal{B}}$ are extracted as $f_{\mathcal{B}}(x) = \mathcal{B}(x)$, and the differentiable loss can be formulated as:

$$\mathcal{L}_{\mathcal{B}}(x, \tilde{x}) = 1 - \cos \left(f_{\mathcal{B}}(x), f_{\mathcal{B}}(\tilde{x}) \right), \tag{3.4}$$

where $\cos(\cdot, \cdot)$ represents the cosine similarity between the two feature vectors. To prevent gradient computation from propagating to the clean representation $f_{\mathcal{B}}(x)$, we utilize a stop-gradient operation $\tilde{f}_{\mathcal{B}}(x) = SG(f_{\mathcal{B}}(x))$. The adversarial input $\tilde{x}$ is initialized with a random perturbation within the ℓ_∞ ball of radius ϵ, and the updates are iteratively projected back onto this ball using the $\text{Proj}_{\mathcal{S}}$ operator:

$$\begin{aligned} \text{Proj}_{\mathcal{S}}(\tilde{x}) &= \text{clip} \left(x + \delta, 0, 1 \right), \\ \text{where} \quad \delta &= \text{clip} \left(\tilde{x} - x, -\epsilon, \epsilon \right). \end{aligned} \tag{3.5}$$

The pseudo-code of the complete process can bee seen in algorithm 1. In summary, the backbone attack focuses solely on the backbone $\mathcal{B}$, without requiring any knowledge of the full target model $\mathcal{M}^{target}$, thereby revealing vulnerabilities inherent to publicly available feature extractors.

3.4 Experimental Setup

3.4.1 Image classification datasets

Through our study, we use 4 datasets covering both classical and domain-specific classification benchmarks, such as CIFAR-10 and CIFAR-100 (Beyer et al., 2020) and Oxford-IIIT Pets (Parkhi et al., 2012), Oxford Flowers-102 (Nilsback and Zisserman, 2008). We train the proxy and target model variation on each one of the datasets using the recipe from (Kolesnikov et al., 2020), reproducing the state-of-the-art model performance results (Dosovitskiy et al., 2020; Yu et al., 2022; Bruno et al., 2022; Foret et al., 2020).

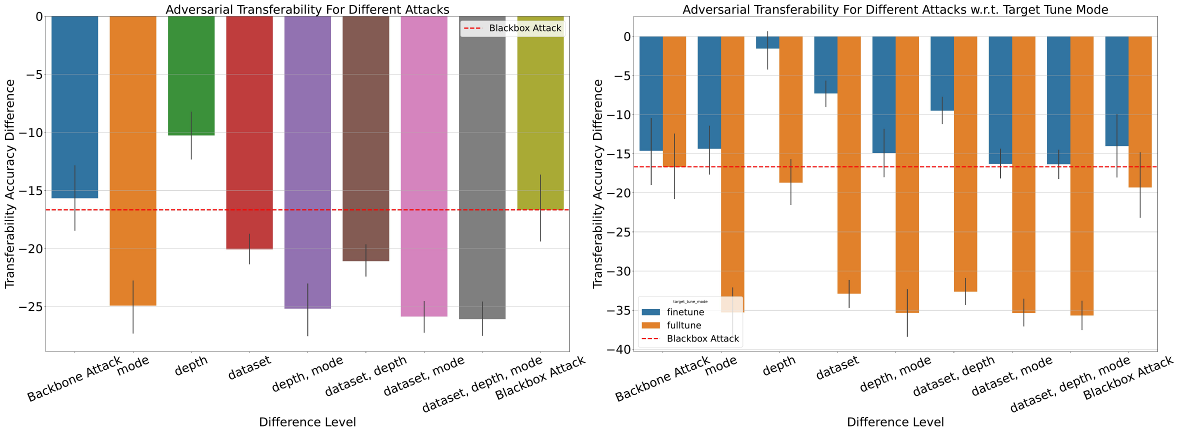


Figure 3.2: The figure depicts the impact of the **unavailability**, i.e. difference from the target model, with each possible meta-information combination on adversarial transferability during proxy attack construction and the backbone attack. The results show the average difference from the *white-box* in transferability using PGD with a higher budget (left) and the segmentation w.r.t. in the target training mode (right).

3.4.2 Model variations

We use 21 different models tuned from 5 architectures, 9 self-supervised objectives and 3 pre-training datasets. A detailed overview of these can be seen in table 3.1.

3.4.3 Model Fintuning Variations

For training the proxy and target models, we employ two *modes* of training $\mathcal{T}$, with full-tuning of the weights and with fine-tuning only the last added classification layers on top of the pre-trained backbone. We also define the depth of tuning $\mathcal{Z}$ as the number of classification layers added on top of the pre-trained backbone. We use $\{1, 3\}$ final layers corresponding to *shallow* and *deep* tuning settings.

3.4.4 Adversarial Attacks

To assess the *white-box* adversarial attack success rate and the adversarial transferability from the proxy models, we employ FGSM (Goodfellow et al., 2015) and PGD (Madry et al., 2018). We use standard attack hyper-parameters introduced in parallel adversarial transferability studies (Waseda et al., 2023; Naseer et al., 2022). For a fair comparison, we also use the same values for our *backbone-attack*. To show that our results are consistent even with a higher computational budget, we report the results of PGD with 4 times more iterations per sample for *white-box*, proxy and *backbone* attack experiments. For *black-box* experiments, we use the Square attack (Andriushchenko et al., 2020), which is a query-efficient method that uses a random search through adversarial sample construction. To standardise the query budget for

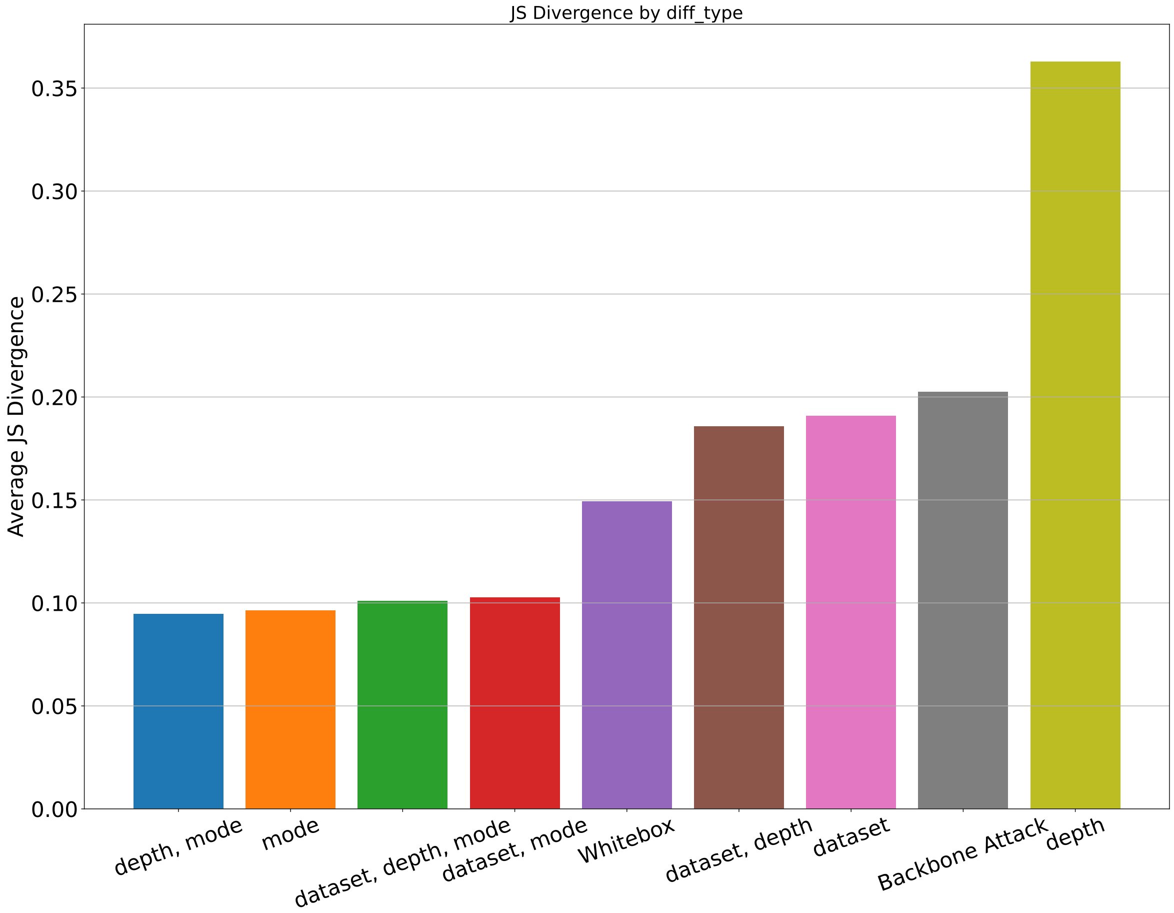


Figure 3.3: The figure breaks down impact of the **unavailability**, i.e. difference from the target model, of each possible meta-information combination on the change in the final decision-making of the model. Higher JS divergence implies a bigger change in the final classification of the sample.

all architectures and simulate real-world constraints, we allow 10 queries of the target model per sample.

3.5 Results

3.5.1 What meta-information matters

To quantify the impact of each possible meta-information availability along with the backbone knowledge during adversarial attack construction, we compute the difference between the adversarial attack success rate (ASR) for the target model and the transferability success rate (TSR) from a proxy model, trained from the same backbone, with partial information. We report the results obtained with the PGD attack trained with higher iteration steps per sample as that is more representative for measuring the adversarial attack success in *white-box* and *grey-box* settings. Our results are consistent across the other attack types and variations.

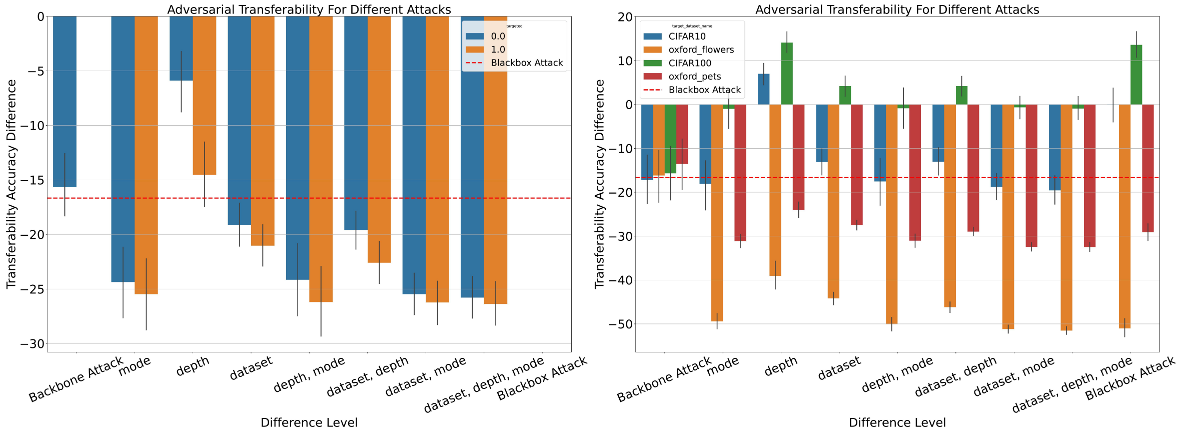


Figure 3.4: The figure depicts the impact of the **unavailability**, i.e. difference from the target model, of each possible meta-information combination on adversarial transferability during proxy attack construction and the backbone attack. The results show the average transferability for PGD with a higher budget for targeted vs untargeted attacks (left) and the segmentation w.r.t. the target training dataset (right).

3.5.2 Which meta-information is important?

Our results in fig. 3.2 show that the most significant performance decay compared to a *white-box* attack performance occurs when the attacker is unaware of the *mode* of the training of the target model, i.e. if it is trained with complete parameters or only tunes the last classification layers. The second most impactful knowledge for attack construction is the availability of the target tuning *dataset*. The *depth* of the tuning is the least important knowledge for obtaining a transferable attack. We further show in the right part of fig. 3.2 that models that finetune the last classification layers can be trivially exploited with transferable attacks, achieving results significantly better than strong black-box exploitation and closing white-box attack performance. It is, however, apparent that training all of the model weights substantially decreases the efficiency of proxy attacks, with almost no correlation towards meta-information availability. We further show that our results remain consistent w.r.t. the choice of the dataset, and regardless if the adversarial attack is targeted or untargeted as seen in fig. 3.4. It is interesting to note that for datasets with more domain-specific content, such as Oxford-IIIT Pets and Oxford Flowers-102, the effectiveness of the proxy attack dwindles, although these datasets are much less diverse compared to CIFAR-100.

3.5.3 Meta-information impacts the quality of adversarial attacks

We also want to measure the effectiveness of the adversarial attack and the impact of meta-information on it by quantifying how the generated adversarial sample has sifted the decision-making of the model. To do this, we compute the entropy of the final softmax layer for each original sample and its adversarial counterpart and complete ANOVA variance analysis (St et al., 1989) of entropy distribution. This analysis, presented in table 3.2, tests whether the means of entropies from original and adversarial images differ significantly across the groups of available meta-information. A perfect attack would produce a sample that does not majorly impact the entropy from the model. The analysis reveals that the target dataset, and tuning mode significantly influence entropy, particularly in adversarial scenarios. This finding suggests that while this meta-information aids in crafting effective adversarial samples, it also plays a critical role in amplifying entropy shifts, thereby making these adversarial samples more detectable.

To quantify the impact of the meta-information availability during attack construction on the decision-making of the model, we also compute the Jensen-Shannon Divergence (Menéndez et al., 1997) between the output softmax distributions of the model produced for original samples and their adversarial counterparts. High JS divergence suggests a strong attack, as the adversarial example causes a significant shift in the model's predicted probabilities, with minimal changes to the input sample. Our results show that not knowing the *mode* of the target model training causes the most degradation in constructing successful adversarial samples with proxy attacks. The second most important fact is the choice of the target *dataset*, while the *depth* of the final classification layers does not seem to be impactful for creating adversarial samples. This reaffirms our findings from fig. 3.2 and fig. 3.3, while also revealing a critical insight: proxy attacks, even when constructed without knowledge of the target model's *dataset* or *depth*, can generate adversarial samples that induce more pronounced distribution shifts than *white-box* attacks. In other words, attackers do not require access to the training dataset or model classification depth to craft adversarial samples capable of significantly disrupting the target model's decision-making process.

3.5.4 Backbone-attacks

To test the extent of the vulnerabilities that the knowledge of the pre-trained backbone can cause, we evaluate our naive exploitation method, *backbone attack*, that utilizes only the pre-trained feature extractor for adversarial sample construction. Our results in fig. 3.2 and fig. 3.4 show that *backbone attacks* are highly effective at producing transferable adversarial samples regardless of the target model tuning *mode*, *dataset* or classification layer *depth*. This naive attack shows significantly higher transferability compared to a strong *black-box* attack with a sizeable query and iteration budget and almost all *proxy attacks*. The results are consistent across all meta-information variations, showing that even a naive attack can exploit the target model vulnerabilities closely to a *white-box* setting, given the knowledge of the pre-trained backbone. Moreover, from fig. 3.3, we see that the adversarial samples produced from this attack, on average, cause a bigger shift in the model's decision-making compared to *white-box attacks*. This indicates that backbone attacks amplify the uncertainty in the target model's predictions, making them more disruptive than conventional *white-box* attacks, highlighting the inherent risks of sharing pre-trained backbones for public use. A concerning aspect of backbone attacks is their effectiveness in resource-constrained environments. Unlike black-box attacks, which often require extensive computation or iterative querying, backbone attacks can be executed with minimal resources, leveraging pre-trained models freely available in public repositories. This ease of implementation raises concerns, as it lowers the barrier for malicious actors to exploit adversarial vulnerabilities.

3.5.5 Knowing weights vs Knowing everything but the weights

To isolate the impact of pre-trained backbone knowledge in adversarial transferability, we train two sets of models from the same ResNet-50 SwAV backbone with identical meta-information variations but different batch sizes. This allows the production of two sets of models with matching training meta-information but varying weights; one set is chosen as the target, and the other as the proxy model. We aim to compare the adversarial transferability of the attacks from the set of proxies towards their matching targets with the backbone attacks. This allows us to simulate conditions where adversaries either know all meta-information but lack the weights or have access to the backbone weights alone. Our results in fig. 3.5 show that the knowledge of the pre-trained backbone is, on average, a stronger or at least an equivalent signal for producing adversarially transferable attacks compared to possessing all of the training meta-information without the knowledge of the weights. The results are consistent across all of the datasets, with domain-specific datasets showing marginal differences in adversarial transferability between the two scenarios. This means that possessing information about only the target model backbone is equivalent to knowing all of the training meta-information for constructing transferable adversarial samples.

3.6 Conclusions

In this paper, we investigated the vulnerabilities of machine vision models fine-tuned from publicly available pre-trained backbones under a novel *grey-box* adversarial setting. Through an extensive evaluation framework, including over 20,000 adversarial

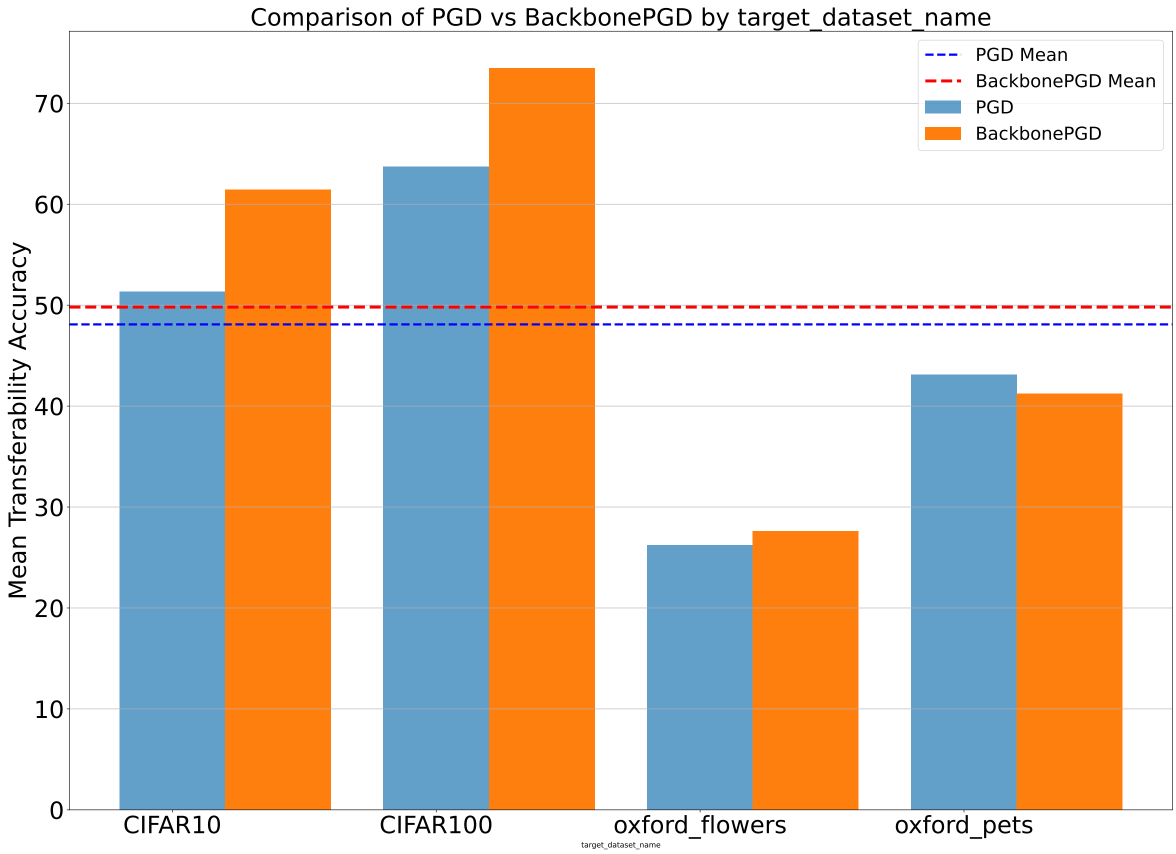


Figure 3.5: The figure shows scenarios where adversaries either know all meta-information but lack the weights or have access to the backbone weights (SwaV ResNet-50) alone. Knowledge of only the backbone is highlighted as *BackbonePGD*.

transferability comparisons, we measured the effect of varying levels of training meta-information availability for constructing transferable adversarial attacks. We also introduced a naive *backbone attack* method, showing that access to backbone weights is sufficient for obtaining adversarial attacks significantly better than query-based *black-box* settings and approaching white-box performance. We found that attacks crafted using only the backbone weights often induce more substantial shifts in the model's decision-making than traditional white-box attacks. We demonstrated that access to backbone weights is equivalent in effectiveness to possessing all meta-information about the target model, making public backbones a critical security concern. Our results highlight significant security risks associated with sharing pre-trained backbones, as they enable attackers to craft highly effective adversarial samples, even with minimal additional information. These findings underscore the need for stricter practices in sharing and deploying pre-trained backbones to mitigate the inherent vulnerabilities exposed by adversarial transferability.

SSL Method	Pretraining Dataset	Architecture
Colorization (Zhang et al., 2016)		
Colorization	YFCC100M	AlexNet
Colorization	ImageNet-1K	AlexNet
Colorization	ImageNet-1K	ResNet-50
Colorization	ImageNet-21K	AlexNet
Colorization	ImageNet-21K	ResNet-50
Jigsaw Puzzle(Noroozi and Favaro, 2016)		
Jigsaw Puzzle	ImageNet-21K	ResNet-50
Jigsaw Puzzle	ImageNet-1K	ResNet-50
Jigsaw Puzzle	ImageNet-21K	ResNet-50
Jigsaw Puzzle	ImageNet-21K	AlexNet
Jigsaw Puzzle	ImageNet-1K	AlexNet
Jigsaw Puzzle	ImageNet-1K	ResNet-50
PIRL (Jigsaw-based)(Misra and van der Maaten, 2020)		
PIRL	ImageNet-1K	ResNet-50
Rotation Prediction (Gidaris et al., 2018)		
RotNet	ImageNet-1K	ResNet-50
DINO(Caron et al., 2021)		
DINO	ImageNet-1K	DeiT-Small
DINO	ImageNet-1K	XCiT-Small
SimCLR(Chen et al., 2020a)		
SimCLR	ImageNet-1K	ResNet-50
SimCLR	ImageNet-1K	ResNet-101
SwAV (Caron et al., 2020)		
SwAV	ImageNet-1K	ResNet-50
SwAV	ImageNet-1K	ResNet-50
DeepCluster V2 (Caron et al., 2018)		
DeepCluster V2	ImageNet-1K	AlexNet
Instance Discrimination (NPID) (Wu et al., 2018)		
NPID	ImageNet-1K	ResNet-50

Table 3.1: Summary of Self-Supervised Learning Methods, Pretraining Datasets, and Architectures used in our study.

	Original Entropy		**Adversarial Entropy**	
Metadata type	**F-Statistic**	**P-Value**	**F-Statistic**	**P-Value**
Target Tune Mode	0.00	0.96	1238.7	0.0
Proxy Tune Mode	0.02	0.88	0.5	0.4
Target Dataset	2812.25	0.00	1184.1	0.0
Proxy Dataset	8.31	0.00	5.0	0.0
Target Tune Depth	5.64	0.01	0.36	0
Proxy Tune Depth	0.08	0.77	0.00	0

Table 3.2: Variance analysis of entropy values across categorical variables. The table shows F-statistics and p-values for both original and adversarial entropy means. Significant p-values (p <0.05) show notable variations in entropy across meta-information.

Part III

Reasoning Inconsistencies from Data

4 Topic-Guided Sampling For Data-Efficient Multi-Domain Stance Detection

4.1 Introduction

The goal of stance detection is to identify the viewpoint expressed by an author within a piece of text towards a designated topic (Mohammad et al., 2016). Such analyses can be used in a variety of domains ranging from identifying claims within political or ideological debates (Somasundaran and Wiebe, 2010; Thomas et al., 2006), identifying mis- and disinformation (Hanselowski et al., 2018; Hardalov et al., 2022a), public health policymaking (Glandt et al., 2021; Hossain et al., 2020; Osnabrügge et al., 2023), news recommendation (Reuver et al., 2021) to investigating attitudes voiced on social media (Qazvinian et al., 2011; Augenstein et al., 2016; Conforti et al., 2020). However, in most domains, and even more so for cross-domain stance detection, the exact formalisation of the task gets blurry, with varying label sets and their corresponding definitions, data collection protocols and available annotations. Furthermore, this is accompanied by significant changes in the topic-specific vocabulary (Somasundaran and Wiebe, 2010; Wei and Mao, 2019), text style (Pomerleau and Rao, 2017; Ferreira and Vlachos, 2016) and topics mentioned either explicitly (Qazvinian et al., 2011; Walker et al., 2012) or implicitly (Hasan and Ng, 2013; Derczynski et al., 2017). Recently, a benchmark of 16 datasets (Hardalov et al., 2021a) covering a variety of domains and topics has been proposed for testing stance detection models across multiple domains. It must be noted that these datasets are highly imbalanced, with an imbalanced label distribution between the covered topics, i.e. inter-topic and within each topic, i.e. per-topic, as can be seen in Figure 4.2 and Figure 4.3. This further complicates the creation of a robust stance detection classifier.

Given the inherent skew present within the dataset and variances within each domain, we propose a topic-guided diversity sampling method, which produces a data-efficient representative subset while mitigating label imbalances. These samples are used for fine-tuning a Pre-trained Language Model (PLM), using a contrastive learning objective to create a robust stance detection model. These two components form our **T**opic **E**fficient **S**tanc**E** **D**etection (TESTED) framework, as seen in Figure 4.1, and are analysed separately to pinpoint the factors impacting model performance and robustness. We test our method on the multi-domain stance detection benchmark by

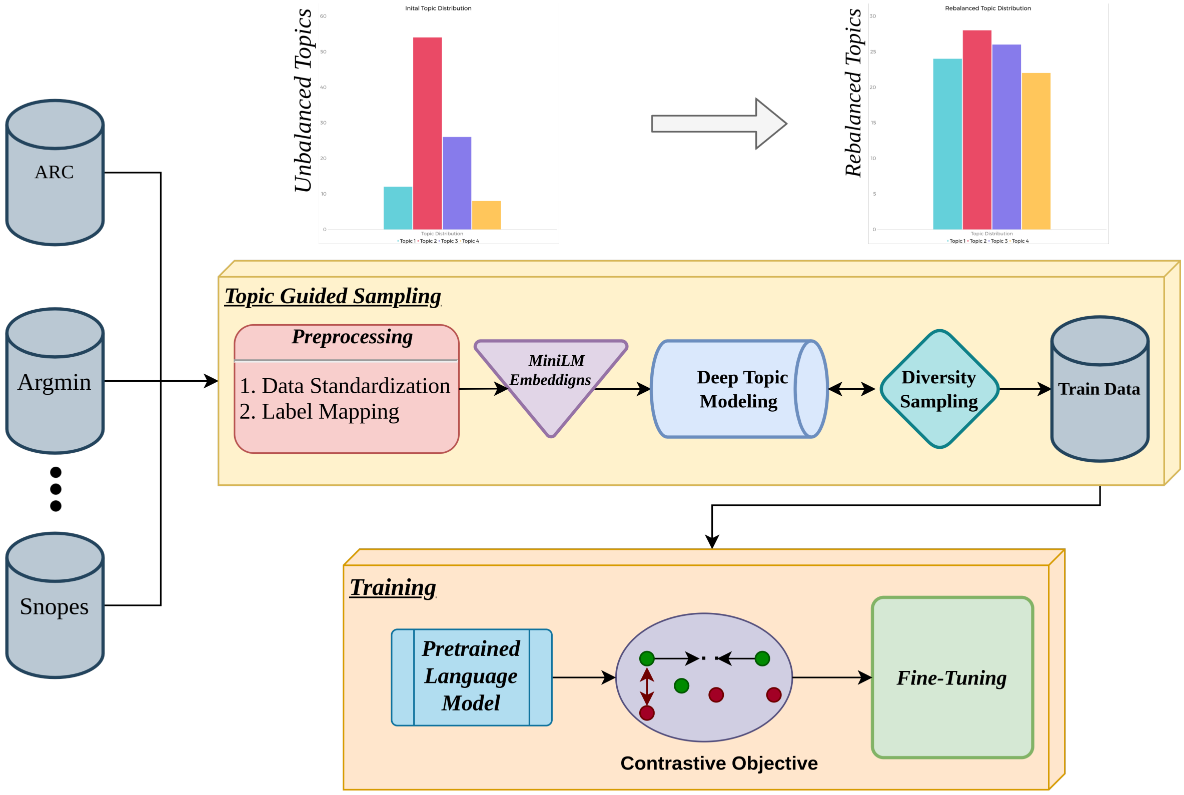


Figure 4.1: The two components of TESTED: Topic Guided Sampling (top) and training with contrastive objective (bottom).

Hardalov et al. (2021a), achieving state-of-the-art results with both in-domain, i.e. all topics seen and out-of-domain, i.e. unseen topics evaluations. Note though that TESTED could be applied to any text classification setting.

In summary, our **contributions** are:

- We propose a novel framework (TESTED) for predicting stances across various domains, with data-efficient sampling and contrastive learning objective;
- Our proposed method achieves SOTA results both in-domain and out-of-domain;
- Our analysis shows that our topic-guided sampling method mitigates dataset imbalances while accounting for better performance than other sampling techniques;
- The analysis shows that the contrastive learning objective boosts the ability of the classifier to differentiate varying topics and stances.

4.2 Related Work

4.2.1 Stance Detection

is an NLP task which aims to identify an author's attitude towards a particular topic or claim. The task has been widely explored in the context of mis- and disinformation detection (Ferreira and Vlachos, 2016; Hanselowski et al., 2018; Zubiaga et al., 2018b; Hardalov et al., 2022a), sentiment analysis (Mohammad et al., 2017; Aldayel and Magdy, 2019) and argument mining (Boltužić and Šnajder, 2014; Sobhani et al., 2015; Wang et al., 2019c). Most papers formally define stance detection as a pairwise sequence classification where stance targets are provided (Küçük and Can, 2020). However, with the emergence of different data sources, ranging from debating platforms (Somasundaran and Wiebe, 2010; Hasan and Ng, 2014; Aharoni et al., 2014) to social media (Mohammad et al., 2016; Derczynski et al., 2017), and new applications (Zubiaga et al., 2018a; Hardalov et al., 2022a), this formal definition has been subject to variations w.r.t. the label dictionary inferred for the task.

Previous research has predominantly focused on a specific dataset or domain of interest, outside of a few exceptions like multi-target (Sobhani et al., 2017; Wei et al., 2018) and cross-lingual (Hardalov et al., 2022b) stance detection. In contrast, our work focuses on multi-domain stance detection, while evaluating in- and out-of-domain on a 16 dataset benchmark with state-of-the-art baselines (Hardalov et al., 2021a).

4.2.2 Topic Sampling

Our line of research is closely associated with diversity (Ren et al., 2021) and importance (Beygelzimer et al., 2009) sampling and their applications in natural language processing (Zhu et al., 2008; Zhou and Lampouras, 2021). Clustering-based sampling approaches have been used for automatic speech recognition (Syed et al., 2016), image classification (Ranganathan et al., 2017; Yan et al., 2022) and semi-supervised active learning (Buchert et al., 2022) with limited use for textual data (Yang et al., 2014b) through topic modelling (Blei et al., 2001). This research proposes an importance-weighted topic-guided diversity sampling method that utilises deep topic models, for mitigating inherent imbalances present in the data, while preserving relevant examples.

4.2.3 Contrastive Learning

has been used for tasks where the expected feature representations should be able to differentiate between similar and divergent inputs (Liu et al., 2021; Rethmeier and Augenstein, 2023). Such methods have been used for image classification (Khosla et al., 2020), captioning (Dai and Lin, 2017) and textual representations (Giorgi et al., 2021; Jaiswal et al., 2020; Ostendorff et al., 2022). The diversity of topics (Qazvinian et al., 2011; Walker et al., 2012; Hasan and Ng, 2013), vocabulary (Somasundaran and Wiebe, 2010; Wei and Mao, 2019) and expression styles (Pomerleau and Rao, 2017) common for stance detection can be tackled with contrastive objectives, as seen for similar sentence embedding and classification tasks (Gao et al., 2021; Yan et al., 2021).

4.3 Datasets

Our study uses an existing multi-domain dataset benchmark (Hardalov et al., 2021a), consisting of 16 individual datasets split into four source groups: *Debates, News, Social Media, Various*. The categories include datasets about debating and political claims including arc (Hanselowski et al., 2018; Habernal et al., 2018), iac1 (Walker et al., 2012), perspectum (Chen et al., 2019), poldeb (Somasundaran and Wiebe, 2010), scd (Hasan and Ng, 2013), news like emergent (Ferreira and Vlachos, 2016), fnc1 (Pomerleau and Rao, 2017), snopes (Hanselowski et al., 2019), social media like mtsd (Sobhani et al., 2017), rumour (Qazvinian et al., 2011), semeval2016t6 (Mohammad et al., 2016), semeval2019t7 (Derczynski et al., 2017), wtwt (Conforti et al., 2020) and datasets that cover a variety of diverse topics like argmin (Stab et al., 2018), ibmcs (Bar-Haim et al., 2017) and vast (Allaway and McKeown, 2020). Overall statistics for all of the datasets can be seen in section 4.11.

4.3.1 Data Standardisation

As the above-mentioned stance datasets from different domains possess different label inventories, the stance detection benchmark by Hardalov et al. (2021a) introduce a mapping strategy to make the class inventory homogeneous. We adopt that same mapping for a fair comparison with prior work, shown in Appendix 4.11.

4.4 Methods

Our goal is to create a stance detection method that performs strongly on the topics known during training and can generalize to unseen topics. The benchmark by Hardalov et al. (2021a) consisting of 16 datasets is highly imbalanced w.r.t the inter-topic frequency and per-topic label distribution, as seen in Figure 4.2.

These limitations necessitate a novel experimental pipeline. The first component of the pipeline we propose is an importance-weighted topic-guided diversity sampling method that allows the creation of supervised training sets while mitigating the inherent imbalances in the data. We then create a stance detection model by fine-tuning a Pre-trained Language Model (PLM) using a contrastive objective.

4.4.1 Topic-Efficient Sampling

We follow the setting in prior work on data-efficient sampling (Buchert et al., 2022; Yan et al., 2022), framing the task as a selection process between multi-domain examples w.r.t the theme discussed within the text and its stance. This means that given a set of datasets $\mathcal{D} = (\mathcal{D}_1, \dots \mathcal{D}_n)$ with their designated documents $\mathcal{D}_i = (d_i^1, \dots d_i^m)$, we wish to select a set of diverse representative examples $\mathcal{D}_{train}$, that are balanced w.r.t the provided topics $\mathcal{T} = (t_1, \dots t_q)$ and stance labels $L = (l_1, \dots l_k)$.

4.4.2 Diversity Sampling via Topic Modeling

We thus opt for using topic modelling to produce a supervised subset from all multi-domain datasets. Selecting annotated examples during task-specific fine-tuning is a challenging task (Shao et al., 2019), explored extensively within active learning research (Hino, 2020; Konyushkova et al., 2017). Random sampling can lead to poor generalization and knowledge transfer within the novel problem domain (Das et al., 2021; Perez et al., 2021). To mitigate the inconsistency caused by choosing suboptimal examples, we propose using deep unsupervised topic models, which allow us to sample relevant examples for each topic of interest. We further enhance the model with an importance-weighted diverse example selection process (Shao et al., 2019; Yang et al., 2015a) within the relevant examples generated by the topic model. The diversity maximisation sampling is modeled similarly to Yang et al. (2015a).

The topic model we train is based on the technique proposed by Angelov (2020) that tries to find topic vectors while jointly learning document and word semantic embeddings. The topic model is initialized with weights from the *all-MiniLM-L6* PLM, which has a strong performance on sentence embedding benchmarks (Wang et al., 2020). It is shown that learning unsupervised topics in this fashion maximizes the total information gained, about all texts $\mathcal{D}$ when described by all words $\mathcal{W}$.

$$\mathcal{I}(\mathcal{D}, \mathcal{W}) = \sum_{d \in \mathcal{D}} \sum_{w \in \mathcal{W}} P(d, w) \log \left(\frac{P(d, w)}{P(d)P(w)} \right)$$

This characteristic is handy for finding relevant samples across varying topics, allowing us to search within the learned documents d_i. We train a deep topic model $\mathcal{M}_{topic}$ using multi-domain data $\mathcal{D}$ and obtain topic clusters $\mathcal{C} = (\mathcal{C}_i, \dots \mathcal{C}_t)$, where $|\mathcal{C}| = t$ is the number of topic clusters. We obtain the vector representation for $\forall d_i$ from the tuned PLM embeddings $\mathcal{E} = (e_1, \dots e_m)$ in $\mathcal{M}_{topic}$, while iteratively traversing through the clusters $\mathcal{C}_i \in \mathcal{C}$.

Our sampling process selects increasingly more diverse samples after each iteration. This search within the relevant examples is presented in algorithm 2. This algorithm selects a set of diverse samples from the given multi-domain datasets $\mathcal{D}$, using the clusters from a deep topic model $\mathcal{M}_{topic}$ and the sentence embeddings $\mathcal{E}$ of the sentences as a basis for comparison. The algorithm starts by selecting a random sentence as the first diverse sample and uses this sentence to calculate a "centroid"

Algorithm 2 Topic Efficient Sampling

```
Input: $S \geq 0$ // Sampling Threshold
Input: $Avg \in \{moving, exp\}$
Output: $|\mathcal{C}| > 0$
$\mathcal{D}_{train} \leftarrow \{\}$
$I \leftarrow \{\frac{|\mathcal{C}_1|}{\sum_{\mathcal{C}_i \in \mathcal{C}} \mathcal{C}_i}, \dots, \frac{|\mathcal{C}_t|}{\sum_{\mathcal{C}_i \in \mathcal{C}} \mathcal{C}_i}\}$ // Cluster Importances
for $\mathcal{C}_i \in \mathcal{C}$ do
    $\mathcal{E}_i \leftarrow \{PLM(d_i^1), \dots\} = \{\mathbf{e}_i^1, \dots, \mathbf{e}_i^m\}$
    $s_i \leftarrow \max(1, S \cdot I_i)$ // Threshold per cluster
    $j \leftarrow 0$
    $cent_0 \leftarrow \frac{\sum_{\mathbf{e}_i \in \mathcal{E}} \mathbf{e}_i}{|\mathcal{E}|}$ // Centroid of the cluster
    while $j \leq s_i$ do
        $sim \leftarrow \frac{\langle \mathcal{E}, cent \rangle}{\|\mathcal{E}\| \|cent\|}$ // Similarity Ranking
        $sample \leftarrow \arg\operatorname{sort}(sim, Ascending)[0]$ // Take the sample most diverse
         from the centroid
        $\mathcal{D}_{train} \leftarrow \mathcal{D}_{train} \cup sample$
        $j \leftarrow j + 1$
        $cent_j \leftarrow \begin{cases} \alpha \cdot \mathbf{e}_{sample} + (1 - \alpha) \cdot cent_{j-1} & \text{if } exp \\ \frac{(j-1)}{j} \cdot cent_{j-1} + \frac{\mathbf{e}_{sample}}{j} & \text{if } moving \end{cases}$ // Centroid update
         w.r.t. sampled data
    end
end
return $\mathcal{D}_{train}$
```

embedding. It then iteratively selects the next most dissimilar sentence to the current centroid, until the desired number of diverse samples is obtained.

4.4.3 Topic-Guided Stance Detection

4.4.4 Task Formalization

Given the topic, t_i for each document d_i in the generated set $\mathcal{D}_{train}$ we aim to classify the stance expressed within that text towards the topic. For a fair comparison with prior work, we use the label mapping from the previous multi-domain benchmark (Hardalov et al., 2021a) and standardise the original labels L into a five-way stance classification setting, $S = \{\text{Positive, Negative, Discuss, Other, Neutral}\}$. Stance detection can be generalized as pairwise sequence classification, where a model learns a mapping $f : (d_i, t_i) \rightarrow S$. We combine the textual sequences with the stance labels to learn this mapping. The combination is implemented using a simple prompt commonly used for NLI tasks (Lan et al., 2020; Raffel et al., 2020; Hambardzumyan et al., 2021), where the textual sequence becomes the premise and the topic the hypothesis.

[CLS] premise: premise

hypothesis: topic [EOS]

The result of this process is a supervised dataset for stance prediction $\mathcal{D}_{train} = ((Prompt(d_1, t_1), s_1) \dots (Prompt(d_n, t_n), s_n))$ where $\forall s_i \in S$. This method allows for data-efficient sampling, as we at most sample 10% of the data while preserving the diversity and relevance of the selected samples. The versatility of the method allows *TESTED* to be applied to any text classification setting.

4.4.5 Tuning with a Contrastive Objective

After obtaining the multi-domain supervised training set $\mathcal{D}_{train}$, we decided to leverage the robustness of PLMs, based on a transformer architecture (Vaswani et al., 2017b) and fine-tune on $\mathcal{D}_{train}$ with a single classification head. This effectively allows us to transfer the knowledge embedded within the PLM onto our problem domain. For standard fine-tuning of the stance detection model $\mathcal{M}_{stance}$ we use cross-entropy as our initial loss:

$$\mathcal{L}_{CE} = -\sum_{i \in S} y_i \log \left(\mathcal{M}_{stance}(d_i)\right) \tag{4.1}$$

Here y_i is the ground truth label. However, as we operate in a multi-domain setting, with variations in writing vocabulary, style and covered topics, it is necessary to train a model where similar sentences have a homogeneous representation within the embedding space while keeping contrastive pairs distant. We propose a new contrastive objective based on the *cosine* distance between the samples to accomplish this. In each training batch $B = (d_1, \dots d_b)$, we create a matrix of contrastive pairs $\mathcal{P} \in \mathcal{R}^{b \times b}$, where $\forall i, j = \overline{1, b}$, $\mathcal{P}_{ij} = 1$ if i-th and j-th examples share the same label and -1 otherwise. The matrices can be precomputed during dataset creation, thus not adding to the computational complexity of the training process. We formulate our pairwise contrastive objective $\mathcal{L}_{CL}(x_i, x_j, \mathcal{P}_{ij})$ using matrix $\mathcal{P}$.

$$\mathcal{L}_{CL} = \begin{cases} e(1 - e^{\cos(x_i, x_j) - 1}), \mathcal{P}_{ij} = 1 \\ e^{\max(0, \cos(x_i, x_j) - \beta)} - 1, \mathcal{P}_{ij} = -1 \end{cases} \tag{4.2}$$

Here x_i, x_j are the vector representations of examples d_i, d_j. The loss is similar to cosine embedding loss and soft triplet loss (Barz and Denzler, 2020; Qian et al., 2019);

	F_1 avg.	arc	iac1	perspectrum	poldeb	scd	emergent	fnc1	snopes	mtsd	rumor	semeval16	semeval19	wtwt	argmin	ibmcs	vast
Majority class baseline	27.60	21.45	21.27	34.66	39.38	35.30	21.30	20.96	43.98	19.49	25.15	24.27	22.34	15.91	33.83	34.06	17.19
Random baseline	35.19	18.50	30.66	50.06	48.67	50.08	31.83	18.64	45.49	33.15	20.43	31.11	17.02	20.01	49.94	50.08	33.25
MoLE	65.55	63.17	38.50	85.27	50.76	**65.91**	**83.74**	75.82	75.07	**65.08**	**67.24**	**70.05**	57.78	68.37	**63.73**	79.38	38.92
TESTED (Our Model)	**69.12**	**64.82**	**56.97**	**83.11**	**52.76**	64.71	82.10	**83.17**	**78.61**	63.96	66.58	69.91	**58.72**	**70.98**	62.79	**88.06**	**57.47**
Topic → Random Sampling	61.14	53.92	42.59	77.68	44.08	52.54	67.55	75.60	72.67	56.35	59.08	66.88	57.28	69.32	52.02	76.93	53.80
Topic → Stratified Sampling	64.01	50.27	51.57	77.78	46.67	62.13	79.00	77.90	76.44	61.50	64.92	68.45	51.96	69.47	56.76	78.30	51.16
- Contrastive Objective	65.63	61.11	55.50	81.85	43.81	63.04	80.84	79.05	73.43	62.18	61.57	60.17	56.06	68.79	59.51	86.94	56.35
Topic Sampling → Stratified - Contrastive Loss	63.24	60.98	49.17	77.85	45.54	58.23	77.36	75.80	74.77	60.85	63.69	62.59	54.74	62.85	53.67	86.04	47.72

Table 4.1: In-domain results reported with macro averaged F1, averaged over experiments. In lines under TESTED, we replace (for Sampling) (→) or remove (for loss) (−), the comprising components.

	F_1 avg.	arc	iac1	perspectrum	poldeb	scd	emergent	fnc1	snopes	mtsd	rumor	semeval16	semeval19	wtwt	argmin	ibmcs	vast
MoLE w/ Hard Mapping	32.78	25.29	35.15	29.55	22.80	16.13	58.49	47.05	29.28	23.34	32.93	37.01	21.85	16.10	34.16	72.93	22.89
MoLE w/ Weak Mapping	49.20	**51.81**	38.97	58.48	47.23	53.96	**82.07**	51.57	56.97	40.13	**51.29**	36.31	31.75	22.75	50.71	75.69	37.15
MoLE w/Soft Mapping	46.56	48.31	32.21	62.73	54.19	51.97	46.86	57.31	53.58	37.88	44.46	36.77	28.92	28.97	57.78	72.11	30.96
TESTED	**59.41**	50.80	**57.95**	**78.95**	**55.62**	**55.23**	80.80	**72.51**	**61.70**	**55.49**	39.44	**40.54**	**46.28**	**42.77**	**72.07**	**86.19**	**54.33**
Topic Sampling → Stratified	50.38	38.47	46.54	69.75	50.54	51.37	68.25	59.41	51.64	48.24	28.04	29.69	34.97	38.13	63.83	83.20	44.06
- Contrastive Loss	54.63	47.96	50.09	76.51	47.49	51.93	75.22	68.69	56.53	49.47	33.95	37.96	44.10	39.56	63.09	83.59	48.03

Table 4.2: Out-of-domain results with macro averaged F1. In lines under TESTED, we replace (for Sampling) (→) or remove (for loss) (−), the comprising components. Results for MoLE w/Soft Mapping are aggregated across with best per-embedding results present in the study (Hardalov et al., 2021a).

however, it penalizes the opposing pairs harsher because of the exponential nature, but does not suffer from computational instability as the values are bounded in the range $[0, e - \frac{1}{e}]$. The final loss is:

$$\mathcal{L} = \mathcal{L}_{CE} + \mathcal{L}_{CL} \tag{4.3}$$

We use the fine-tuning method from Mosbach et al. (2021); Liu et al. (2019c) to avoid the instability caused by catastrophic forgetting, small-sized fine-tuning datasets or optimization difficulties.

4.5 Experimental Setup

4.5.1 Evaluation

We evaluate our method on the 16 dataset multi-domain benchmark and the baselines proposed by Hardalov et al. (2021a). To directly compare with prior work, we use the same set of evaluation metrics: macro averaged F1, precision, recall and accuracy.

4.5.2 Model Details

We explore several PLM transformer architectures within our training and classification pipelines in order to evaluate the stability of the proposed technique. We opt to finetune a pre-trained *roberta-large* architecture (Liu et al., 2019c; Conneau et al., 2020). For fine-tuning, we use the method introduced by Mosbach et al. (2021), by adding a linear warmup on the initial 10% of the iteration raising the learning rate to $2e^{-5}$ and decreasing it to 0 afterwards. We use a weight decay of $\lambda = 0.01$ and train for 3 epochs with global gradient clipping on the stance detection task. We further show that learning for longer epochs does not yield sizeable improvement over the initial fine-tuning. The optimizer used for experimentation is an AdamW (Loshchilov and Hutter, 2019) with a bias correction component added to stabilise the experimentation (Mosbach et al., 2021).

4.5.3 Topic Efficiency

Recall that we introduce a topic-guided diversity sampling method within ***TESTED***, which allows us to pick relevant samples per topic and class for further fine-tuning. We evaluate its effectiveness by fine-tuning PLMs on the examples it generates and comparing it with training on a random stratified sample of the same size.

4.6 Results and Analysis

In this section, we discuss and analyze our results, while comparing the performance of the method against the current state-of-the-art (Hardalov et al., 2021a) and providing an analysis of the topic efficient sampling and the contrastive objective.

4.6.1 Stance Detection

4.6.2 In-domain

We train on our topic-efficient subset $\mathcal{D}_{train}$ and test the method on all datasets $\mathcal{D}$ in the multi-domain benchmark. Our method TESTED is compared to MoLE (Hardalov et al., 2021a), a strong baseline and the current state-of-the-art on the benchmark. The results, presented in Table 4.1, show that TESTED has the highest average performance on in-domain experiments with an increase of 3.5 F1 points over MoLE, all while using $\leq 10\%$ of the amount of training data in our subset $\mathcal{D}_{train}$ sampled from the whole dataset $\mathcal{D}$. Our method is able to outperform all the baselines on 10 out of 16 datasets. On the remaining 6 datasets the maximum absolute difference between TESTED and MoLE is 1.1 points in F1. We also present ablations for TESTED, by replacing the proposed sampling method with other alternatives, removing the contrastive objective or both simultaneously. Replacing Topic Efficient sampling with either *Random* or *Stratified* selections deteriorates the results for all datasets with an average decrease of 8 and 5 F1 points, respectively. We attribute this to the inability of other sampling techniques to maintain inter-topic distribution and per-topic label distributions balanced while selecting diverse samples. We further analyse how our sampling technique tackles these tasks in section 4.6.4. We also see that removing the contrastive loss also results in a deteriorated performance across all the datasets with an average decrease of 3 F1 points. In particular, we see a more significant decrease in datasets with similar topics and textual expressions, i.e. *poldeb* and *semeval16*, meaning that learning to differentiate between contrastive pairs is essential within this task. We analyse the effect of the contrastive training objective further in section 4.6.9.

4.6.3 Out-of-domain

In the out-of-domain evaluation, we leave one dataset out of the training process for subsequent testing. We present the results of TESTED in Table 4.2, showing that it is able to overperform over the previous state-of-the-art significantly. The metrics in each column of Table 4.2 show the results for each dataset held out from training and only evaluated on. Our method records an increased performance on 13 of 16 datasets, with an averaged increase of 10.2 F1 points over MoLE, which is a significantly more pronounced increase than for the in-domain setting, demonstrating that the strength of TESTED lies in better out-of-domain generalisation. We can also confirm that replacing the sampling technique or removing the contrastive loss results in lower performance across all datasets, with decreases of 9 and 5 F1 points respectively. This effect is even more pronounced compared to the in-domain experiments, as adapting to unseen domains and topics is facilitated by diverse samples with a balanced label distribution.

4.6.4 Imbalance Mitigation Through Sampling

4.6.5 Inter-Topic

To investigate the inter-topic imbalances, we look at the topic distribution for the top 20 most frequent topics covered in the complete multi-domain dataset $\mathcal{D}$, which accounts for $\geq 40\%$ of the overall data. As we can see in Figure 4.2, even the most frequent topics greatly vary in their representation frequency, with $\sigma = 4093.55$, where σ is the standard deviation between represented amounts. For the training dataset $\mathcal{D}_{train}$, by contrast, the standard deviation between the topics is much smaller $\sigma = 63.59$. This can be attributed to the fact that $\mathcal{D}_{train}$ constitutes $\leq 10\%$ of $\mathcal{D}$, thus we also show the aggregated data distributions in Figure 4.2. For a more systematic

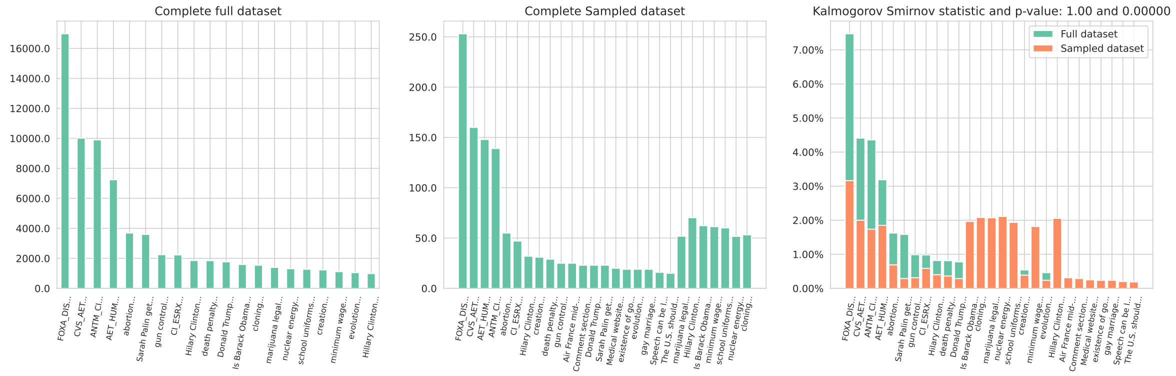


Figure 4.2: Distributions of top 20 most frequent topics in complete dataset $\mathcal{D}$ (left), Sampled dataset $\mathcal{D}_{train}$ (mid) and their aggregated comparison (right). The distribution of top 20 topics in $\{\mathcal{D}\} - \{\mathcal{D}_{train}\}$ is added to the tail of the figure (mid).

dataset	stat	p-value
fnc-1-ours	1.00	0.007937
arc	0.40	0.873016
emergent	0.80	0.079365
wtwt	0.20	1.000000
rumor	0.40	0.873016
snopes	0.40	0.873016
perspectrum	0.60	0.357143
vast	0.60	0.357143
semeval2016task6	0.40	0.873016
iac	0.40	0.873016
mtsd	0.25	1.000000
argmin	0.40	0.873016
scd	1.00	0.007937
ibm_claim_stance	0.80	0.079365
politicaldebates	0.50	1.000000

Table 4.3: KS test for topic distributions. The topics in bold designate a rejected null-hypothesis (criteria: $p \leq 0.05$ or $stat \geq 0.4$), that the topics in $\mathcal{D}$ and $\mathcal{D}_{train}$ come from the same distribution.

analysis, we employ the two sample Kolmogorov-Smirnov (KS) test (Massey, 1951), to compare topic distributions in $\mathcal{D}$ and $\mathcal{D}_{train}$ for each dataset present in $\mathcal{D}$. The test compares the cumulative distributions (CDF) of the two groups, in terms of their maximum-absolute difference, stat $= \sup_x |F_1(x) - F_2(x)|$.

The results in Table 4.3 show that the topic distribution within the full and sampled data $\mathcal{D}$, $\mathcal{D}_{train}$, cannot be the same for most of the datasets. The results for the maximum-absolute difference also show that with at least 0.4 difference in CDF, the sampled dataset $\mathcal{D}_{train}$ on average has a more balanced topic distribution. The analysis in Figure 4.2 and Table 4.3, show that the sampling technique is able to mitigate the inter-topic imbalances present in $\mathcal{D}$. A more in-depth analysis for each dataset is provided in section 4.9.

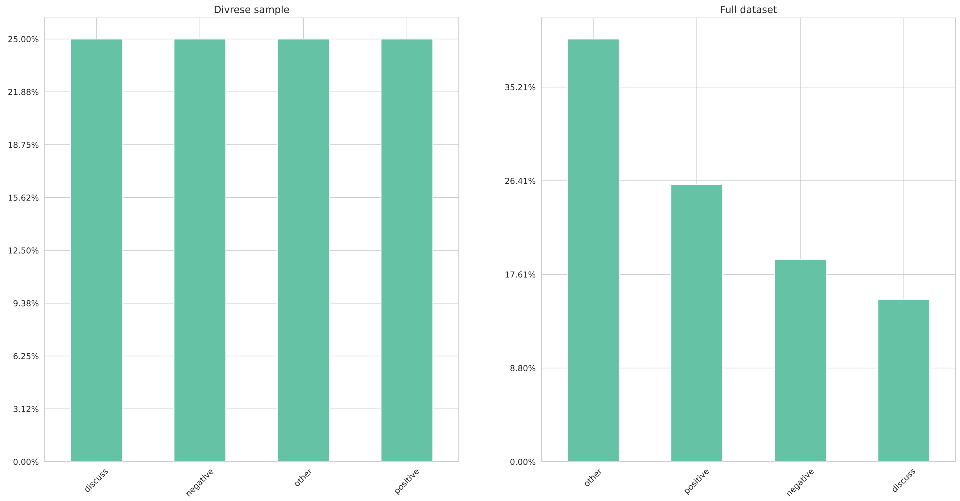


Figure 4.3: Label distribution in $\mathcal{D}$ (right) and $\mathcal{D}_{train}$ (left).

4.6.6 Per-topic

For the per-topic imbalance analysis, we complete similar steps to the inter-topic analysis, with the difference that we iterate over the top 20 frequent topics looking at *label* imbalances within each topic. We examine the label distribution for the top 20 topics for a per-topic comparison. The standard deviation in label distributions averaged across those 20 topics is $\sigma = 591.05$ for the whole dataset $\mathcal{D}$ and the sampled set $\mathcal{D}_{train}$ $\sigma = 11.7$. This can be attributed to the stratified manner of our sampling technique. This is also evident from Figure 4.3, which portrays the overall label distribution in $\mathcal{D}$ and $\mathcal{D}_{train}$.

To investigate the difference in label distribution for each of the top 20 topics in $\mathcal{D}$, we use the KS test, presented in Table 4.4. For most topics, we see that the label samples in $\mathcal{D}$ and $\mathcal{D}_{train}$ cannot come from the same distribution. This means that the per-topic label distribution in the sampled dataset $\mathcal{D}_{train}$, does not possess the same imbalances present in $\mathcal{D}$.

We can also see the normalized standard deviation for the label distribution within $\mathcal{D}_{train}$ is lower than in $\mathcal{D}$, as shown in Figure 4.4. This reinforces the finding that per-topic label distributions in the sampled dataset are more uniform. For complete per-topic results, we refer the reader to section 4.9.

4.6.7 Performance

Using our topic-efficient sampling method is highly beneficial for in- and out-of-domain experiments, presented in Table 4.1 and Table 4.2. Our sampling method can select diverse and representative examples while outperforming *Random* and

topic	p-values
FOXA_DIS	0.028571
CVS_AET	0.028571
ANTM_CI	0.028571
AET_HUM	0.047143
abortion	0.100000
Sarah Palin getting divorced?	0.028571
gun control	0.001879
CI_ESRX	0.028571
Hilary Clinton	0.001468
death penalty	0.100000
Donald Trump	0.002494
Is Barack Obama muslim?	0.028571
cloning	0.333333
marijuana legalization	0.032178
nuclear energy	0.333333
school uniforms	0.333333
creation	0.003333
minimum wage	0.333333
evolution	0.100000
lockdowns	0.000491

Table 4.4: KS test for label distributions. The topics in bold designate a rejected null-hypothesis (criteria: $p \leq 0.05$), that the label samples in $\mathcal{D}$ and $\mathcal{D}_{train}$ averaged per top 20 topics come from the same distribution.

Stratified sampling techniques by 8 and 5 F1 points on average. This performance can be attributed to the mitigated inter- and per-topic imbalance in $\mathcal{D}_{train}$.

4.6.8 Data Efficiency

TESTED allows for sampling topic-efficient, diverse and representative samples while preserving the balance of topics and labels. This enables the training of data-efficient models for stance detection while avoiding redundant or noisy samples. We analyse the data efficiency of our method by training on datasets with sizes $[1\%, 15\%]$ compared to the overall data size $|\mathcal{D}|$, sampled using our technique. Results for the in-domain setting in terms of averaged F1 scores for each sampled dataset size are shown in Figure 4.5. One can observe a steady performance increase with the more selected samples, but diminishing returns from the 10% point onwards. This leads us to use 10% as the optimal threshold for our sampling process, reinforcing the data-efficient nature of TESTED.

4.6.9 Contrastive Objective Analysis

To analyse the effect of the contrastive loss, we sample 200 unseen instances stratified across each dataset and compare the sentence representations before and after training. To compare the representations, we reduce the dimension of the embeddings with t-SNE and cluster them with standard K-means. We see in Figure 4.6 that using the objective allows for segmenting contrastive examples in a more pronounced way. The cluster purity also massively rises from 0.312 to 0.776 after training with the

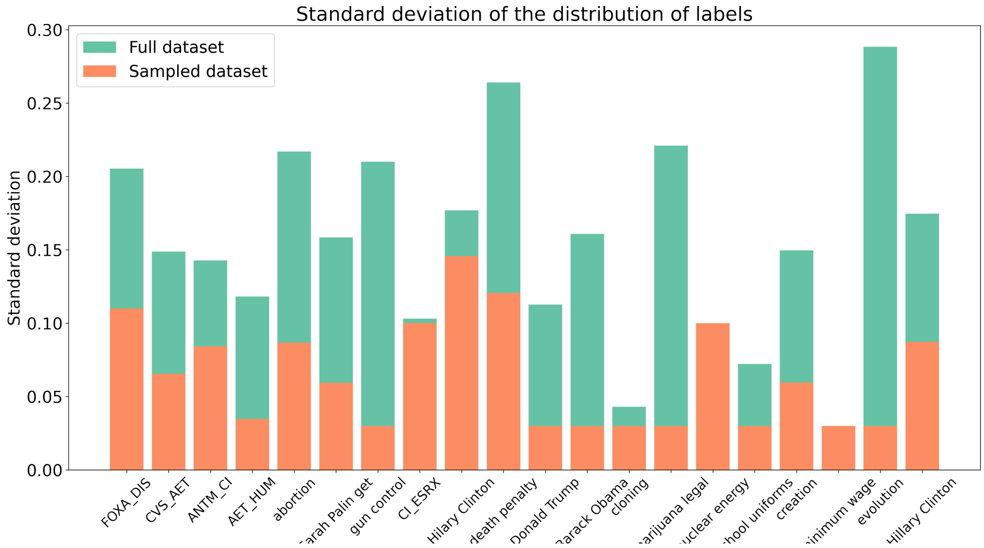


Figure 4.4: Normalized Standard Deviation in label distribution for top 20 topics.

contrastive loss. This allows the stance detection model to differentiate and reason over the contrastive samples with greater confidence.

4.7 Conclusions

We proposed TESTED, a novel end-to-end framework for multi-domain stance detection. The method consists of a data-efficient topic-guided sampling module, that mitigates the imbalances inherent in the data while selecting diverse examples, and a stance detection model with a contrastive training objective. TESTED yields significant performance gains compared to strong baselines on in-domain experiments, but in particular generalises well on out-of-domain topics, achieving a 10.2 F1 point improvement over the state of the art, all while using $\leq 10\%$ of the training data. While in this paper, we have evaluated TESTED on stance detection, the method is applicable to text classification more broadly, which we plan to investigate in more depth in future work.

Limitations

Our framework currently only supports English, thus not allowing us to complete a cross-lingual study. Future work should focus on extending this study to a multilingual setup. Our method is evaluated on a 16 dataset stance benchmark, where some domains bear similarities. The benchmark should be extended and analyzed further to find independent datasets with varying domains and minimal similarities, allowing for a more granular out-of-domain evaluation.

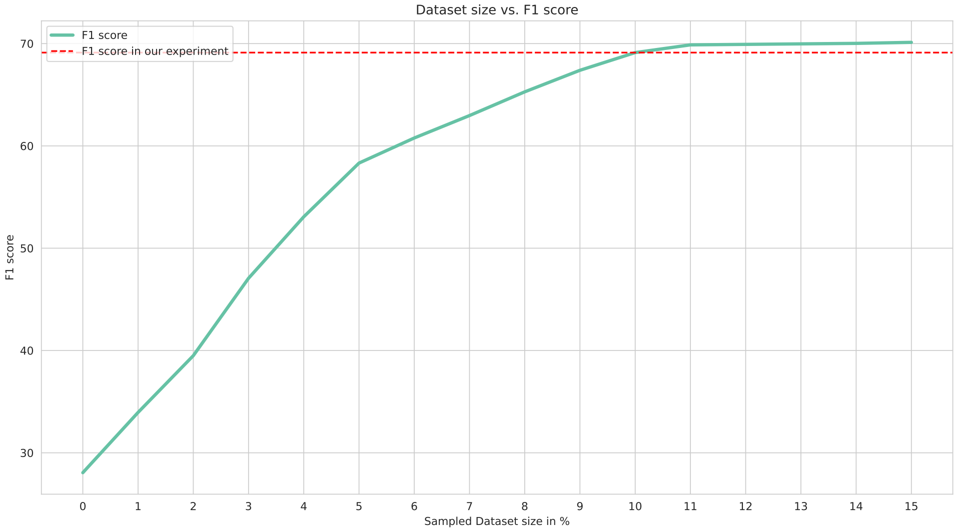


Figure 4.5: Sampled Data size vs Performance. Performance increases with a bigger sampled selection.

Acknowledgements

This research is funded by a DFF Sapere Aude research leader grant under grant agreement No 0171-00034B, as well as supported by the Pioneer Centre for AI, DNRF grant number P1.

4.8 Appendix

4.9 Imbalance analysis

4.9.1 Inter-topic

To complement our inter-topic imbalance mitigation study, we complete an ablation on all topics in $\mathcal{D}$ and report them on a per-domain basis in Figure 4.7. The trend is similar to the one in Figure 4.2, where the dataset with imbalanced distributions is rebalanced, and balanced datasets are not corrupted.

4.9.2 Per-topic

We show that our topic-efficient sampling method allows us to balance the label distribution for unbalanced topics, while not corrupting the ones distributed almost uniformly. To do this, we investigate each of the per-topic label distributions for the top 20 most frequent topics while comparing the label distributions for $\mathcal{D}$ and $\mathcal{D}_{train}$, presented in Figure 4.8.

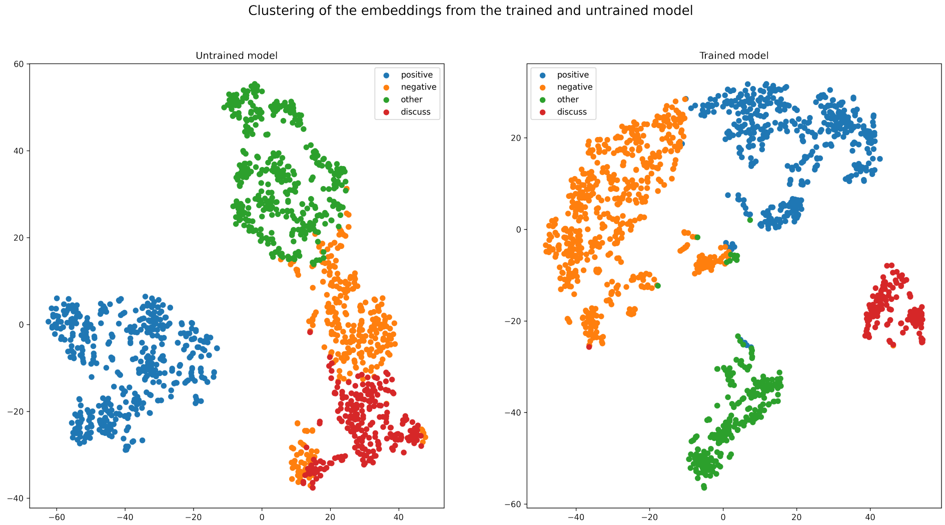


Figure 4.6: Sample Representation before (left) and after (right) contrastive training.

fnc-1-ours
arc
emergent
wtwt
rumor
snopes
perspectrum
vast
semeval2016task6
iac
mtsd
argmin
scd
ibm_claim_stance
politicaldebates

Figure 4.7: Distributions of top 20 most frequent topics for each dataset (left), Sampled dataset $\mathcal{D}_{train=dataset}$ (mid) and their aggregated comparison (right).

Figure 4.8: Distributions of labels for top 20 most frequent topics for $\mathcal{D}$ (left), Sampled dataset $\mathcal{D}_{train=dataset}$ (mid) and their aggregated comparison (right).

4.10 Evaluation Metrics

To evaluate our models and have a fair comparison with the introduced benchmarks we use a standard set of metrics for classification tasks such as macro-averaged F1, precision, recall and accuracy.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.4}$$

$$Prec = \frac{TP}{TP + FP} \tag{4.5}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.6}$$

$$F1 = \frac{2 * Prec * Recall}{Prec + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \tag{4.7}$$

4.11 Dataset Statistics

We use a stance detection benchmark (Hardalov et al., 2021a) whose data statistics are shown in Table 4.5. The label mapping employed is shown in Table 4.6.

4.12 TESTED with different backbones

We chose to employ different PLM's as the backbone for TESTED and report the results in the Table 4.7. The PLMs are taken from the set of *roberta-base, roberta-large, xlm-roberta-base, xlm-roberta-large.* The differences between models with a similar

Dataset	Train	Dev	Test	Total
arc	12,382	1,851	3,559	17,792
argmin	6,845	1,568	2,726	11,139
emergent	1,770	301	524	2,595
fnc1	42,476	7,496	25,413	75,385
iac1	4,227	454	924	5,605
ibmcs	935	104	1,355	2,394
mtsd	3,718	520	1,092	$5,330$
perspectrum	6,978	2,071	2,773	11,822
poldeb	4,753	1,151	1,230	7,134
rumor	6,093	471	505	$7,276$
scd	3,251	624	964	4,839
semeval2016t6	2,497	417	1,249	4,163
semeval2019t7	5,217	1,485	1,827	8,529
snopes	14,416	1,868	3,154	19,438
vast	13,477	2,062	3,006	18,545
wtwt	25,193	7,897	18,194	51,284
Total	154,228	30,547	68,495	253,270

Table 4.5: Dataset statistics of the stance detection benchmark by Hardalov et al. (2021a) also used in this paper. Note that the rumour and mtsd datasets are altered in that benchmark as some of the data was unavailable.

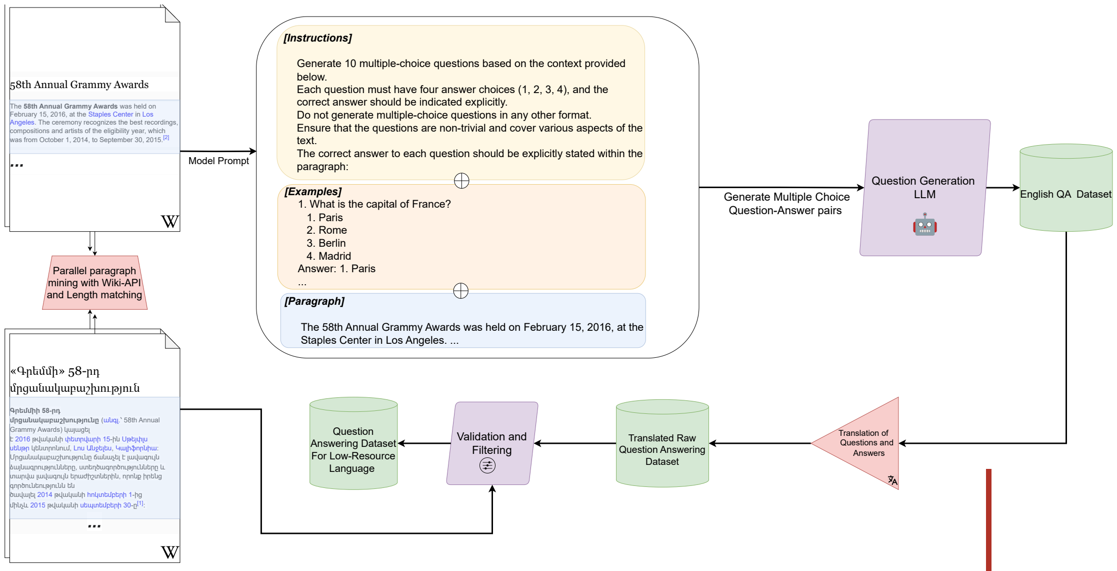
number of parameters are marginal. We can see a degradation of the F1 score between the *base* and *large* versions of the models, which can be attributed to the expressiveness the models possess. We also experiment with the distilled version of the model and can confirm that in terms of the final F1 score, it works on par with the larger models. This shows that we can utilise smaller and more computationally efficient models within the task with marginal degradation in overall performance.

Label	Description
Positive	agree, argument for, for, pro, favor, support, endorse
Negative	disagree, argument against, against, anti, con, undermine, deny, refute
Discuss	discuss, observing, question, query, comment
Other	unrelated, none, comment
Neutral	neutral

Table 4.6: Hard stance label mapping employed in this paper, following the stance detection benchmark by Hardalov et al. (2021a).

	F_1 avg.	arc	iac1	perspectrum	poldeb	scd	emergent	fnc1	snopes	mtsd	rumor	semeval16	semeval19	wtwt	argmin	ibmcs	vast
TESTED$_{reberta\text{-}large}$	69.12	64.82	56.97	83.11	52.76	64.71	82.10	83.17	78.61	63.96	66.58	69.91	58.72	70.98	62.79	88.06	57.47
TESTED$_{xlm\text{-}reberta\text{-}large}$	68.86	64.35	57.0	82.71	52.93	64.75	81.72	82.71	78.38	63.66	66.71	69.76	58.27	71.29	62.73	87.75	57.2
TESTED$_{reberta\text{-}base}$	65.32	59.71	51.86	76.75	50.23	61.35	78.84	82.09	73.31	62.87	65.46	63.89	58.3	67.28	58.28	83.81	51.09
TESTED$_{xlm\text{-}reberta\text{-}base}$	65.05	60.26	51.96	76.2	51.82	58.74	74.68	77.9	72.61	62.71	66.08	69.74	53.27	65.83	59.09	87.92	52.08
TESTED$_{distilroberta\text{-}base}$	68.86	61.78	56.94	80.36	46.29	64.1	79.26	81.37	73.44	62.6	63.4	63.75	56.53	68.35	57.27	81.93	56.3

Table 4.7: In-domain results reported with macro averaged F1, with varying backbones when using TESTED.


58th Annual Grammy Awards
...
W
Model Prompt
Parallel paragraph mining with Wiki-API and Length matching
«Գրեմմի» 58-րդ մրցանակաբաշխություն
...
W
[Instructions]
[Examples]
[Paragraph]
Generate Multiple Choice Question-Answer pairs
Question Generation LLM
English QA Dataset
Translation of Questions and Answers
Translated Raw Question Answering Dataset
Validation and Filtering
Question Answering Dataset For Low-Resource Language

The 58th Annual Grammy Awards was held on February 15, 2016, at the Staples Center in Los Angeles. The ceremony recognizes the best recordings, compositions and artists of the eligibility year, which was from October 1, 2014, to September 30, 2015.[2]

Գրեմմի 58-րդ մրցանակաբաշխությունը (անգլ.՝ 58th Annual Grammy Awards) կայացել է 2016 թվականի փետրվարի 15-ին Սթեյփլս սենթր կենտրոնում, Լոս Անջելես, Կալիֆորնիա։ Մրցանակաբաշխությունը ճանաչել է լավագույն ձայնագրությունները, ստեղծագործությունները և տարվա լավագույն երաժիշտներին, որոնք իրենց գործունեությունն են ծավալել 2014 թվականի հոկտեմբերի 1-ից մինչև 2015 թվականի սեպտեմբերի 30-ը[1]։

Generate 10 multiple-choice questions based on the context provided below.
Each question must have four answer choices (1, 2, 3, 4), and the correct answer should be indicated explicitly.
Do not generate multiple-choice questions in any other format.
Ensure that the questions are non-trivial and cover various aspects of the text.
The correct answer to each question should be explicitly stated within the paragraph:

1. What is the capital of France?
1. Paris
2. Rome
3. Berlin
4. Madrid
Answer: 1. Paris
...

The 58th Annual Grammy Awards was held on February 15, 2016, at the Staples Center in Los Angeles. ...

Figure 5.1: The proposed framework is comprised of three components: (i) a module for mining parallel paragraphs using wiki-API and length matching; (ii) generating a synthetic question-answering dataset with an LLM using the mined English paragraphs; (iii) translating the question-answer pairs and Filtering/Validating them for obtaining a high-quality synthetic QA dataset in the low-resource language.

5 SynDARin: Synthesising Datasets for Automated Reasoning in Low-Resource Languages

5.1 Introduction

Question Answering (QA) has been a hallmark task for testing reading comprehension and reasoning capabilities in NLP systems. The availability of numerous English benchmarks that frame the problem as extractive, cloze-style or open-domain (Yang et al., 2015b; Rajpurkar et al., 2016; Chen et al., 2017a) reasoning tasks, along with novel pre-trained language models (PLMs) (Devlin et al., 2018; Lewis et al., 2019a) and LLMs (Touvron et al., 2023b; Jiang et al., 2023; Achiam et al., 2023) allowed for the development and granular evaluation of QA systems that occasionally boast human-like or better performance (Devlin et al., 2018; Min et al., 2023; Rogers et al., 2023). Cross-lingual alignment through translation-following has improved performance on multilingual benchmarks like XQUAD and MLQA (Ranaldi and Pucci, 2023). Although some concentrated effort has been made to create multilingual QA resources (Lewis et al., 2019b; Asai et al., 2018; Liu et al., 2019a), the datasets remain rather scarce and usually cover a small selected set of languages due to the labour-intensive annotation costs. The proposed methods suggest using direct machine translation (Lewis et al., 2019b; Carrino et al., 2019) or multilingual synthetic data generation (Riabi et al., 2020; Agrawal et al., 2023; Shakeri et al., 2020). However, these approaches are directly bound to introduce biases and hallucinations during translation (Artetxe et al., 2020), cross-lingual transfer (Lauscher et al., 2020; Guerreiro et al., 2023) or generation (Ahuja et al., 2023). These limitations directly hinder the possibility to *develop* and *evaluate* the multilingual QA capabilities of language models in low-resource languages.

In this work, we propose **S**yn**DAR**in, a novel method for synthesising datasets for automated reasoning in low-resource languages that circumvents the above-mentioned obstacles and test it by creating a QA dataset for the Armenian language, which has virtually no presence of structured NLP datasets (Avetisyan and Broneske, 2023). We mine parallel English and Armenian introductory paragraphs from the same diverse set of Wikipedia articles, ensuring that the contents match by comparing their relative length. Similar mining approaches have been shown to be efficient for this task (Lewis et al., 2021; Artetxe and Schwenk, 2019). This allows us to obtain human-curated text from diverse topics while bypassing a wide chunk of direct content translation and annotation. Given the English subset of this data, we generate MC question-answer pairs by prompting an LLM to produce queries with an answer explicitly mentioned within the paragraph. Following Lewis et al. (2019b), we filter out examples that do not contain the answer substring verbatim in the paragraph and additionally perform a human evaluation on a subset of 50 examples and show that 98% of these question-answer pairs are answerable and maintain quality. The produced question-answers are subsequently translated using an automated tool and further validated by answer substring and semantic matching in the parallel Armenian paragraph. This allows us to mitigate the likelihood of hallucinated, biased, and inconsistent entries in the final QA dataset. Our human evaluation with native Armenian speakers shows that 70% of such corrupted examples are removed. We use the dataset as a reasoning benchmark for Armenian and evaluate several LLMs in zero-shot, few-shot, and fine-tuned modes. We show that the dataset cannot be trivially solved, thus highlighting it as a useful resource for measuring model performance. In sum, our contributions are as follows: (i) a novel method for QA dataset construction in low-resource languages, (ii) a QA dataset in Armenian, (iii) ablations showing the quality of the generated samples, and (iv) an evaluation of several LLM families on the QA dataset.

Who	Where	What	When	Which	How	General	Why
304	128	1536	215	473	244	76	16

Table 5.1: Frequency of Question Types in the generated English question-answer pairs.

5.2 Methodology

An outline of **SynDARin** can be seen in fig. 5.1.

5.2.1 Parallel Data Mining

Given parallel English and Armenian introductory paragraph tokens $\mathcal{P}_{\text{En}} = (T_1, \dots T_n)$, $\mathcal{P}_{\text{Arm}} = (T_1, \dots T_m)$ obtained from a diverse set of Wiki articles, we want to save the segments that contain the same content. As the introductory paragraphs in Wikipedia contain highly similar information (Lewis et al., 2019b), we found that filtering out the paragraph pairs based on their relative view count and the number of tokens, i.e. length, is sufficient. To do this, we simply define a conditional rejection process on Wikipedia pages that have been viewed more than 1000 and edited more than 5 times $|||\mathcal{P}_{\text{En}}|| - ||\mathcal{P}_{\text{Arm}}||| \leq K_{\text{DM}}$, where K_{DM} is the threshold for the length difference. A higher length difference would imply that the contents of the paragraphs are misaligned, thus making us reject such samples. Consequently, we are able to obtain naturally written human-curated parallel paragraphs that cover a diverse set of topics.

5.2.2 QA Generation

After obtaining the parallel data, we prompt an LLM $\mathcal{M}$ with instructions $\mathcal{I} = (T_1, \dots T_{|\mathcal{I}|})$ and 10 in-context example demonstrations $\mathcal{E} = (E_1, \dots E_{10})$, where $\forall i, E_i = (T_1, \dots T_{|E_i|})$, to generate diverse English MC question-answer pairs $\mathcal{K}_{\text{Eng}} = \{(q_1, a_1) \dots (q_N, a_N)\}$ given an English context paragraph $\mathcal{P}_{\text{En}}$:

$$q_i, a_i \sim \prod_{t=1}^{|\mathcal{K}_i|} P_{\mathcal{M}} \left(T_t^{(i)} \mid T_1^{(i)}, ..., T_{t-1}^{(i)}, \mathcal{I}, \mathcal{E}, \mathcal{P}_{\text{En}} \right) \tag{5.1}$$

We filter out all repeating questions, $\forall \{i, j : i \neq j\}, q_i \neq q_j$, and question-answer pairs where the answer span is not exactly mentioned within the text, i.e. $a_i \not\subset \mathcal{P}_{\text{En}}$. An example input used for generation can be seen in fig. 5.1. This generation and validation pipeline resembles the ones in Lewis et al. (2021); Agrawal et al. (2023), which have shown successful question-generation results for the English language. Several examples of produced questions are available in section 5.6.

Problem type(%)	*Filtered*	*Unfiltered*
Partially Missing Info	38	77
Bad Translation	5	51
Partially Correct Answers	22	31
Several Correct Answers	27	45
Date Mismatch	13	17
Other	8	22

Table 5.2: Unanswerable sample analysis before(Unfiltered) and after(Filtered) the validation. Annotators can choose multiple reasons per sample.

5.2.3 Translation and Validation

We transfer the generated question-answer pairs $\mathcal{K}_{\text{Eng}}$ into Armenian by using the Google Translate API to obtain $\mathcal{K}_{\text{Arm}}$. To mitigate the inconsistencies introduced during the translation process, we save only the samples where the translated answer $a_i \in \mathcal{K}_{\text{Arm}}$ is contained within and semantically related to the paragraph $\mathcal{P}_{\text{Arm}}$. To do this, we use a fuzzy substring matching function $\mathcal{F} : \mathcal{T} \times \mathcal{T} \to [0, 1]$, along with a multilingual language model $\mathcal{M}_{\text{sim}} : \mathcal{T} \to \mathcal{R}^d$ to measure semantic similarity, where $\mathcal{T}$ is an arbitrary set of tokens and d is the dimensionality of the embedding space of the model. Samples below a certain threshold, $\mathcal{F}(a_i, \mathcal{P}_{\text{Arm}}) \leq K_{\text{Fuzz}}$ and $\cos(\mathcal{M}(a_i), \mathcal{M}(\mathcal{P}_{\text{Arm}})) \leq K_{\text{Sim}}$ are filtered out. Note that exact matching is insufficient, as the morphology of the translated answer tokens can vary in the low-resource language. The multiple-choice answers are balanced uniformly in the final dataset so as not to introduce a bias toward any particular answer ordering.

5.3 Experimental Setup

5.3.1 QA Generation

Our QA generation uses GPT-4 (Achiam et al., 2023), known for generating high-quality text (Zhou et al., 2023b) and synthetic data (Hämäläinen et al., 2023; Li et al., 2023).

5.3.2 Substring Matching and Semantic Similarity

We employ Levenshtein distance for fuzzy substring matching ($\mathcal{F}$) and multilingual sentence embeddings (Reimers and Gurevych, 2019) ($\mathcal{M}_{\text{sim}}$) for semantic similarity using cosine distance.

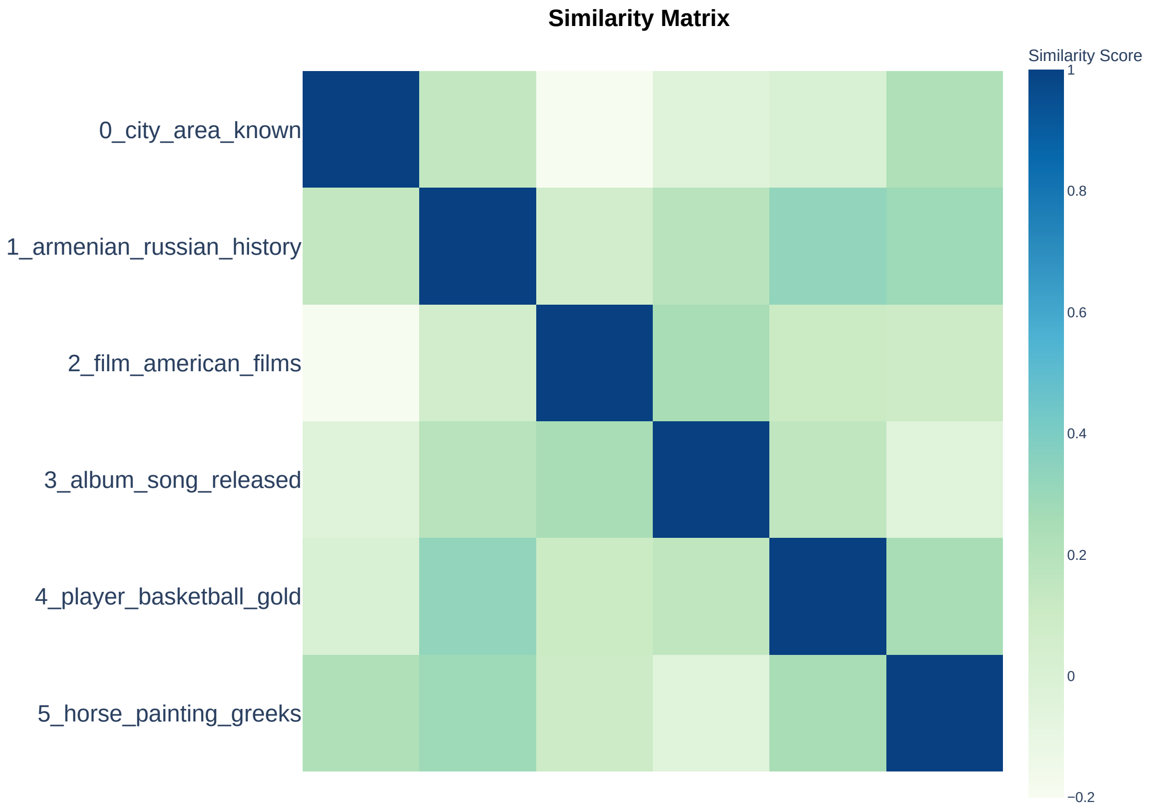


Figure 5.2: BERTopic embeddings similarity heatmap for the top 6 frequent topics in the mined English paragraphs.

5.3.3 Armenian QA Benchmarking

We benchmark GPT-3.5 (Achiam et al., 2023), CMD-R, and CMD-R+ (Cohere, 2024) using $\{0, 2, 4, 6\}$ in-context examples with few-shot prompting (Brown et al., 2020b) on the Armenian QA dataset. We further frame the task as classification with multiple-choice answers and perform supervised fine-tuning with a recipe (Mosbach et al., 2021) on XLM-RoBERTa-base (Conneau et al., 2020), with $\{32, 64, \ldots, 980\}$ training samples and benchmark it on the same testing set. Following Poliak et al. (2018), we analyze model performance on *question-only* and *paragraph-only* inputs for bias detection.

5.4 Results

	Accuracy			
Filter	128	256	512	987
Complete	30.1%	33.5%	38.7%	39.5%
paragraph-only	26.7%	28.3%	23.9%	28.3%
question-only	22.1%	22.7%	19.4%	23.5%
Random performance	**25.0%**			

Table 5.3: The results of fine-tuning XLM-Roberta on the Armenian QA dataset with a varying number of training samples in different degeneracy testing scenarios.

5.4.1 English QA Dataset Generation

We mined 300 parallel English-Armenian Wikipedia paragraphs and generated 10 diverse questions with 4 MC answers each, resulting in 3000 English QA pairs.

5.4.2 Dataset Diversity

We assessed question diversity (table 5.1) and found meaningful variation consistent with prior human-curated datasets (Lewis et al., 2019b; Rajpurkar et al., 2016). Topic modeling using BERTopic (Grootendorst, 2022) validated the subject diversity (fig. 5.2). A granular diversity analysis within the dataset is presented in section 5.6.

5.4.3 Human Evaluation

To assess the data quality, we follow Lewis et al. (2021) and ask two English-speaking human annotators to manually inspect 50 randomly chosen samples from the English QA dataset regarding the captured contextual information and answerability of the sample question. The results show, with an inter-annotator agreement score of Cohen's $\kappa = 0.99$, that 98% of examples contain sufficient details to answer the question while accurately capturing contextual information.

5.4.4 Automatic Translation and Validation

We translate the obtained 3000 QA samples and pass the results through our validation pipeline to produce 1235 filtered Armenian examples.

5.4.5 Armenian QA dataset

We use these samples and their designated Armenian paragraphs to form the QA dataset. We split the data into $80/20$ *train/test* buckets with 987 samples in training and 247 in testing. We ensure that the paragraphs in the testing set are not contained in the train set to avoid any data leakage. We maintain a uniform distribution of MC questions within the answers, avoiding bias towards any answer ordering.

	Accuracy			
Model Name	0	2	4	6
Command-R	58.7%	**68.4%**	64.8%	64.0%
Command-R+	59.3%	67.2%	69.6%	**70.9%**
GPT-3.5	56.3%	56.3%	**59.1%**	54.3%

Table 5.4: Model Accuracy with a varying number of provided in-context samples before generation.

5.4.6 Human Evaluation

We assessed the translation validation pipeline and datasets using two native-speaking annotators. They reviewed the *test* set, which was mixed with 100 randomly flagged poor samples from automatic validation. Annotators either answered the samples or marked them as unanswerable, citing reasons from a predefined set, see in table 5.2. Results showed that 87% of the flagged examples were unanswerable due to insufficient context, translation errors, or hallucinations. The error breakdown in table 5.2 highlights the quality improvement in filtered samples w.r.t. to the abovementioned discrepancies, where annotators answered correctly in 75% of cases. We measure the inter-annotator agreement using Cohen's $\kappa = 0.8$. These confirm the ability of our validation pipeline to maintain the dataset quality.

5.4.7 Benchmarks

To show the value of the created dataset, we investigate if it suffers from statistical biases or degenerate solutions by training an XLM-RoBERTa model on inputs that contain only the paragraph or the question, excluding everything else from the sample. The results in table 5.3 show that regardless of the number of training samples, the models trained with question and paragraph-only samples behave similarly to random chance, while training with complete data gradually increases the performance, highlighting that the dataset is unlikely to suffer from inconsistencies and degenerate solutions and can be used for developing QA capabilities for Armenian. We further benchmark several state-of-the-art LLMs on this dataset in supervised fine-tuning, *zero-shot* and *few-shot* settings. We see in table 5.4 that even the largest models do not trivially solve the dataset, showing its utility as a benchmarking tool.

5.5 Conclusion

We propose **S**yn**DAR**in, a novel method for constructing QA datasets for low-resource languages and producing a dataset for the Armenian language. Systematic studies of the reliability of the individual modules to produce diverse QA samples that maintain answerability and quality show the effectiveness of the method. We further use the produced Armenian QA dataset to benchmark state-of-the-art LLMs and show the value of the proposed resource in evaluating QA reasoning capabilities in the low-resource language.

Limitations

The proposed methods have currently been tested only for a smaller-scale QA dataset creation in Armenian, thus not allowing us to complete a wider cross-lingual study. The study benchmarks should be extended and analyzed further in more multilingual, low-resource languages. In the case of extremely rare low-resource languages, the automatic translation part within our pipeline would require either the development of such a translation method, robust cross-lingual transfer from a similar language, or direct manual effort, all of which are bound to introduce either qualitative or logistic complications while creating the final QA resource.

Acknowledgments

Erik is partially funded by a DFF Sapere Aude research leader grant under grant agreement No 0171-00034B, as well as by an NEC PhD fellowship, and is supported by the Pioneer Centre for AI, DNRF grant number P1. Pasquale was partially funded by ELIAI (The Edinburgh Laboratory for Integrated Artificial Intelligence), EPSRC (grant no. EP/W002876/1), an industry grant from Cisco, and a donation from Accenture LLP. Isabelle's research is partially funded by the European Union (ERC, ExplainYourself, 101077481), and is supported by the Pioneer Centre for AI, DNRF grant number P1. This work was supported by the Edinburgh International Data Facility (EIDF) and the Data-Driven Innovation Programme at the University of Edinburgh.

5.6 Appendix

OTHER	NORP	GPE	PERCENT	PERSON	DATE	ORG	WORK OF ART	LANGUAGE	QUANTITY	EVENT	MONEY	LOC	ORDINAL	TIME	FAC	PRODUCT
3178	172	223	8	397	335	327	14	10	25	21	9	52	38	9	9	3

Table 5.5: Distribution of Entities within question-answer pairs in the generated English QA dataset. The Entity labelling scheme follows Honnibal et al.

5.6.1 Generated Question-Answer pairs

We showcase examples of generated and validated question-answer pairs along with their designated English paragraph $\mathcal{P}_{\text{Eng}}$ in table 5.6. These are representative samples

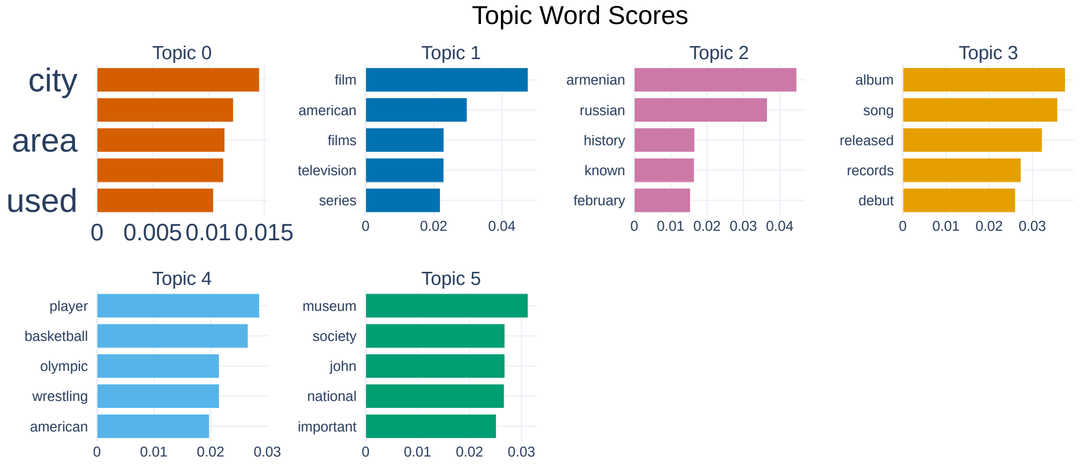


Figure 5.3: The usage of frequent words in the top 6 frequent topics present within the mined English paragraphs.

of the generation process, further reinforced by the fact that human evaluation of the quality of the generation showed that 98% of the examples are answerable and maintain quality.

5.6.2 What are the questions about?

To understand the type of inquiries asked within the questions, we employ a pre-trained model for Named Entity Recognition (NER) from spaCy[1] and detect all the entity types mentioned within the question-answer pairs. The results can be seen in table 5.5, showing that the object of the inquiries can vary massively from people (PERSON) and locations (LOC) to organization (ORG), numeric values (DATE, ORDINAL, TIME), etc. This further ensures that we are able to generate high-quality questions with diverse compositions and object of inquiry types.

5.6.3 Topic Distribution the parallel paragraphs

To estimate the overlap within the topics found in the mined paragraphs, we use unsupervised topic modeling BERTopic (Grootendorst, 2022) to segment the 5 most frequently occurring segments. We measure the overlap between these by calculating the averaged cosine distance of the topic embeddings obtained from BERTopic. The results can be seen in fig. 5.2 and fig. 5.3, validating our hypothesis that we are able to cover diverse themes using our parallel paragraph mining method.

[1] https://spacy.io/api/entityrecognizer

Example 1: UEFA Champions League
Since the rebranding of the European Champion Clubs' Cup as the UEFA Champions League in 1992, 107 different players from 37 countries have scored three goals or more in a single match (a hat-trick) on 152 occasions, representing 53 clubs from 17 leagues. The first player to achieve the feat was Juul Ellerman, who scored three times for PSV Eindhoven in a 6–0 victory over Žalgiris on 16 September 1992. Lionel Messi and Cristiano Ronaldo have scored three or more goals in a match eight times each in the Champions League, more than any other player, followed by Robert Lewandowski with six, and Karim Benzema with four. **Question:** What was the original name of the UEFA Champions League? **Answers:** 1. European Champion Clubs' Cup, 2. European Premier League, 3. UEFA Football Cup, 4. European Soccer Championship **Correct Answer:** 1. European Champion Clubs' Cup
Example 2: Sign Languages
Sign languages (also known as signed languages) are languages that use the visual-manual modality to convey meaning, instead of spoken words. Sign languages are expressed through manual articulation in combination with non-manual markers. Sign languages are full-fledged natural languages with their own grammar and lexicon. Sign languages are not universal and are usually not mutually intelligible, although there are also similarities among different sign languages. **Question:** What is the primary modality used to convey meaning in sign languages? **Answers:** 1. Auditory-vocal, 2. Visual-manual, 3. Tactile-kinesthetic, 4. Olfactory-gustatory **Correct Answer:** 2. Visual-manual

Table 5.6: Examples of English paragraphs along with their generated question-answer pairs

5.6.4 Benchmarking with Armenian QA dataset

To show the usefulness of the created dataset, we benchmark several SOTA LLMs on it in supervised fine-tuning, *zero-shot* and *few-shot* settings. We further investigate if the dataset suffers from statistical biases or degenerate solutions by training an XLM-RoBERTa model on inputs that contain only the paragraph or the question, excluding everything else from the sample. The results in fig. 5.5 show us that regardless of the amount of provided training samples, the question, and paragraph-only evaluations behave similarly to random chance, highlighting that the dataset is unlikely to suffer from inconsistencies and degenerate solutions.

We benchmark several LLMs, shown in fig. 5.4, using produced Armenian QA benchmark and show that while increasing the number of model parameters and in-context samples helps the overall model performance, still even very large models are unable to solve the dataset trivially, thus showing its value as a benchmarking resource.

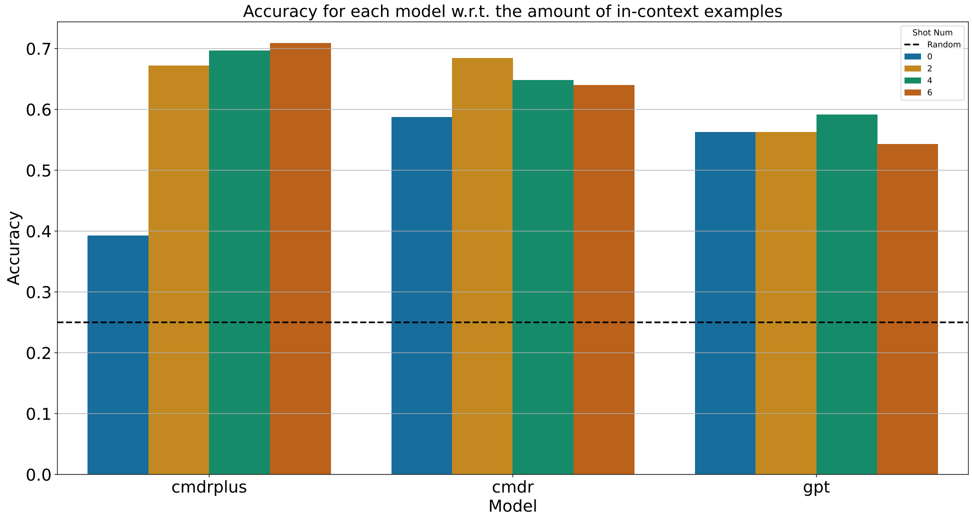


Figure 5.4: Accuracy of each model with a varying number of in-context examples given before generation.

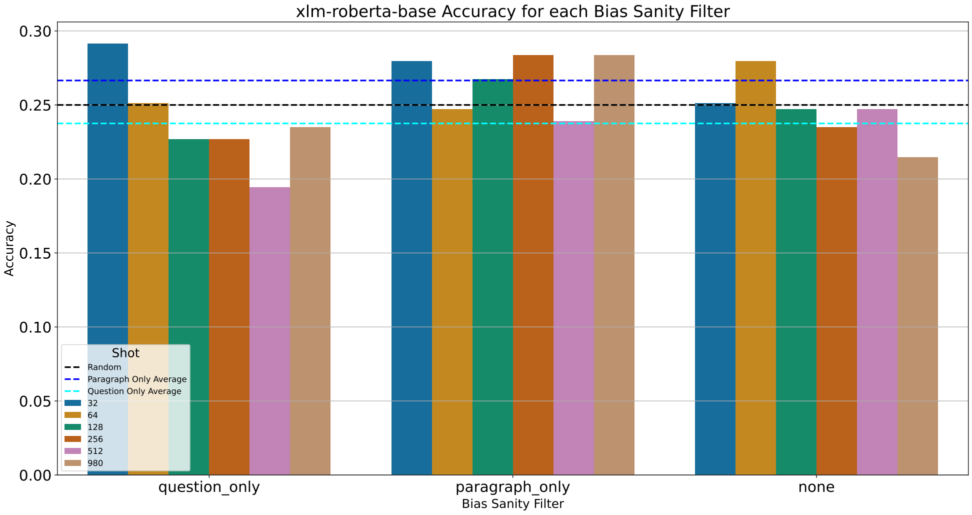


Figure 5.5: The results of fine-tuning XLM-Roberta on the Armenian QA dataset with a varying number of training samples while using only paragraphs, questions or random data.

Part IV

Reasoning Inconsistencies from Task Complexity

Adapting Neural Link Predictors for Complex Query Answering

6

6.1 Introduction

Knowledge Graphs (KGs) are knowledge bases representing the relationships between entities in a relational graph structure. The flexibility of this knowledge representation formalism allows KGs to be widely used in various domains. Examples of KGs include general-purpose knowledge bases such as Wikidata (Vrandečić and Krötzsch, 2014), DBpedia (Auer et al., 2007), Freebase (Bollacker et al., 2008), and YAGO (Suchanek et al., 2007); application-driven graphs such as the Google Knowledge Graph, Microsoft's Bing Knowledge Graph, and Facebook's Social Graph (Noy et al., 2019); and domain-specific ones such as SNOMED CT (Bodenreider et al., 2018), MeSH (Lipscomb, 2000), and Hetionet (Himmelstein et al., 2017) for life sciences; and WordNet (Miller, 1992) for linguistics. Answering complex queries over Knowledge Graphs involves a logical reasoning process where a conclusion should be inferred from the available knowledge.

Neural link predictors (Nickel et al., 2016) tackle the problem of identifying missing edges in large KGs. However, in many complex domains, it is challenging to develop techniques for answering complex queries involving multiple and potentially unobserved edges, entities, and variables rather than just single edges.

Example 6.1.1 (Complex Query)**.** Consider the question "*Which people are male and produced the music for the film Constantine?*", which corresponds to a query from the FB15K-237 dataset. It can be formalised as a complex query $\mathcal{Q} \equiv ?T : gender(male, T) \wedge producerOf(Constantine, T)$, where *male* and *Constantine* are anchor nodes and T is the target of the query, as presented in fig. 6.1. The answer set is denoted as $[\![\mathcal{Q}]\!]$ and corresponds to all the entities in the knowledge graph that are male composers for the film Constantine.

Continuous Query Decomposition (CQD, Arakelyan et al., 2021; Minervini et al., 2022) is a framework for answering First-Order Logic Queries, where the query is compiled in an end-to-end differentiable function, modelling the interactions between its atoms. Despite only being trained on atomic queries, CQD achieves significantly more accurate results than other complex query-answering methods while also being

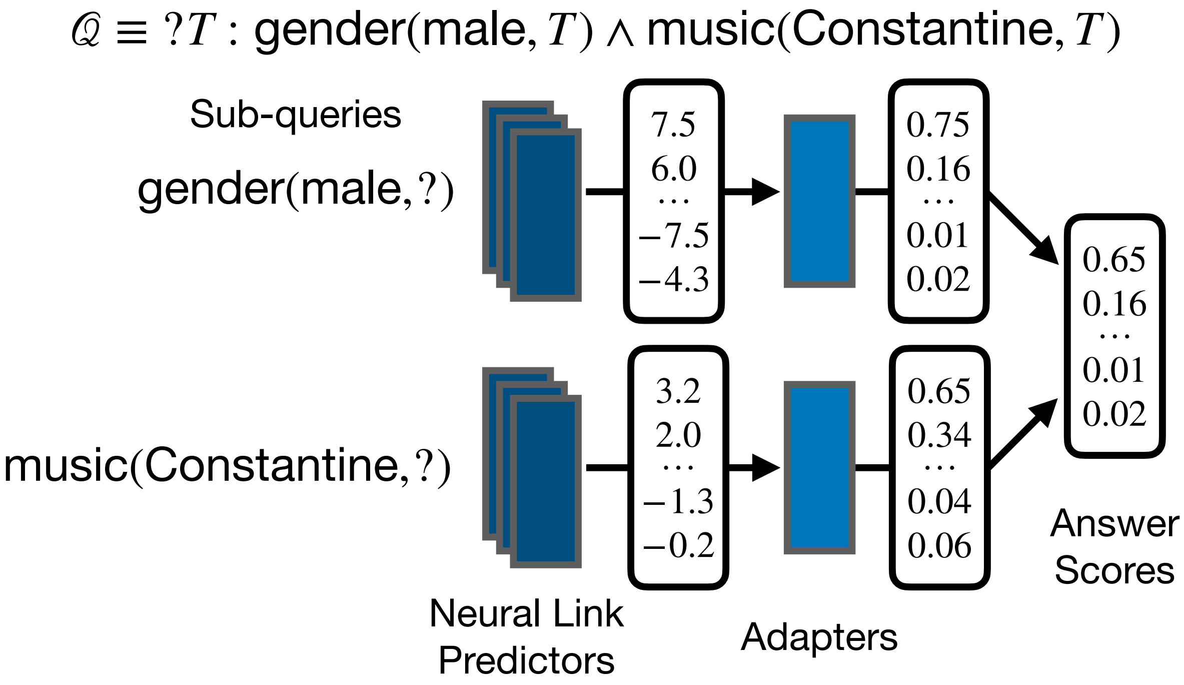


Figure 6.1: Outline of CQD$^{\mathcal{A}}$– given a complex query $\mathcal{Q}$, the neural link prediction scores for the sub-queries are further adapted by neural layers for improving the interaction between them. The adaptation layers are trained on the downstream complex query-answering task via back-propagation.

more explainable (Arakelyan et al., 2021; Minervini et al., 2022). A limitation of this method is that the model is only trained to answer simple atomic queries: the scores of atoms can be accurate when used individually for answering atomic queries but do not interact when answering more complex queries.

We propose a new method CQD$^{\mathcal{A}}$, introduced in section 6.4, for answering complex logical queries on knowledge graphs with the addition of negations. The main idea in CQD$^{\mathcal{A}}$ is to incorporate additional neural components that improve the interaction between atomic query scores; these components are trained end-to-end on complex query-answering tasks. An evaluation of the method on three benchmark datasets for complex query answering shows an increase from 34.4 to 35.1 MRR over the current state-of-the-art averaged across all datasets while using $\leq 35\%$ of the available training query types. The ablations show that the method is data-efficient by achieving results comparable to the state-of-the-art while using 1% of the training data. Our experiments reveal that CQD$^{\mathcal{A}}$ can generalise across unseen query types while using only 1% of the instances from a single query type during training.

6.2 Related Work

6.2.1 Link Predictors in Knowledge Graphs

Reasoning over KGs with missing nodes has been widely explored throughout the last few years. One can approach the task using latent feature models, such as neural link predictors (Bordes et al., 2013; Trouillon et al., 2016; Yang et al., 2014a; Dettmers et al., 2018; Sun et al., 2019; Balažević et al., 2019; Amin et al., 2020) which learn continuous representations for the entities and relation types in the graph and can answer atomic queries over incomplete KGs, and Graph Neural Networks (GNNs, Schlichtkrull et al., 2018; Vashishth et al., 2019a; Teru et al., 2020). Other lines of research tackle the link prediction problem through graph feature models (Xiong et al., 2017; Das et al., 2017; Hildebrandt et al., 2020; Yang et al., 2017; Sadeghian et al., 2019).

6.2.2 Complex Query Answering

Complex queries over knowledge graphs can be formalised by extending one-hop atomic queries with First Order Logic (FOL) operators, such as the existential quantifier ($\exists$), conjunctions ($\wedge$), disjunctions ($\vee$) and negations ($\neg$). These FOL constructs can be represented as directed acyclic graphs, which are used by embedding-based methods that represent the queries using geometric objects (Ren et al., 2020; Hamilton et al., 2018) or probabilistic distributions (Ren and Leskovec, 2020; Zhang et al., 2021; Choudhary et al., 2021) and search the embedding space for the answer set. It is also possible to enhance the properties of the embedding space using GNNs and Fuzzy Logic (Zhu et al., 2022b; Chen et al., 2022). Recent work (Daza and Cochez, 2020; Hamilton et al., 2018; Ren and Leskovec, 2020) suggests that such methods require a large dataset with millions of diverse queries during the training, and it can be hard to explain their predictions.

Our work is closely associated with CQD (Arakelyan et al., 2021; Minervini et al., 2022), which uses a pre-trained neural link predictor along with fuzzy logical t-norms and t-conorms for complex query answering. One limitation of CQD is that the pre-trained neural link predictors are not calibrated to interact during the complex query-answering process. This means that the final scores of the model are highly dependent on the choice of the particular t-(co)norm aggregation functions. This leads to discrepancies within the intermediate reasoning process and final predictions. CQD was also not designed to leverage complex queries during training and does not support logical negations.

Our method, CQD$^{\mathcal{A}}$, extends CQD so that it can be trained on complex queries via gradient-based optimisation and can answer a broader class of FOL queries – for example, queries with logical negations.

6.3 Background

In this section, we introduce the complex query types we consider and the CQD complex query answering method.

6.3.1 Existential Positive First-Order Logical Queries

A Knowledge Graph $\mathcal{G} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ can be defined as a set of subject-predicate-object $\langle s, p, o\rangle$ triples, where each triple encodes a relationship of type $p \in \mathcal{R}$ between the subject $s \in \mathcal{E}$ and the object $o \in \mathcal{E}$ of the triple, where $\mathcal{E}$ and $\mathcal{R}$ denote the set of all entities and relation types, respectively. One can think of a Knowledge Graph as a labelled multi-graph, where entities $\mathcal{E}$ represent nodes, and edges are labelled with relation types $\mathcal{R}$. Without loss of generality, a Knowledge Graph can be represented as a First-Order Logic Knowledge Base, where each triple $\langle s, p, o\rangle$ denotes an atomic formula $p(s, o)$, with $p \in \mathcal{R}$ a binary predicate and $s, o \in \mathcal{E}$ its arguments.

Conjunctive queries are a sub-class of First-Order Logical queries that use existential quantification (∃) and conjunction (∧) operations. We consider conjunctive queries $\mathcal{Q}$ in the following form:

$$
\begin{aligned}
\mathcal{Q}[A] \triangleq ?\, A : \quad & \exists V_1, \ldots, V_m . e_1 \wedge \ldots \wedge e_n \\
& \mathit{where}\ e_i = p(c, V), \\
& \quad \mathit{with}\ V \in \{A, V_1, \ldots, V_m\}, c \in \mathcal{E}, p \in \mathcal{R}, \\
& \mathit{or}\ e_i = p(V, V'), \\
& \quad \mathit{with}\ V, V' \in \{A, V_1, \ldots, V_m\}, V \neq V', p \in \mathcal{R}.
\end{aligned} \tag{6.1}
$$

In eq. (6.1), the variable A is the *target* of the query, $V_1, \ldots, V_m$ denote the *bound variable nodes*, while $c \in \mathcal{E}$ represent the *input anchor nodes*. Each e_i denotes a logical atom, with either one ($p(c, V)$) or two variables ($p(V, V')$), and $e_1 \wedge \ldots \wedge e_n$ denotes a conjunction between n atoms.

The goal of answering the logical query $\mathcal{Q}$ consists in finding a set of entities $[\![\mathcal{Q}]\!] \subseteq \mathcal{E}$ such that $a \in [\![\mathcal{Q}]\!]$ iff $\mathcal{Q}[a]$ holds true, where $[\![\mathcal{Q}]\!]$ is the *answer set* of the query $\mathcal{Q}$.

As illustrated in fig. 6.1, the *dependency graph* of a conjunctive query $\mathcal{Q}$ is a graph representation of $\mathcal{Q}$ where nodes correspond to variable or non-variable atom arguments in $\mathcal{Q}$ and edges correspond to atom predicates. We follow Hamilton et al. (2018) and focus on *valid* conjunctive queries – i.e. the dependency graph needs to be a directed acyclic graph, where anchor entities correspond to source nodes, and the query target A is the unique sink node.

6.3.2 Handling Disjunctions and Negations

We aim to answer a broader class First Order Logical queries (Dalvi and Suciu, 2004), which include the Existential Positive First-Order (EPFO) family, but in addition to existential quantification and conjunctions, also include disjunctions ($\vee$) and negations ($\neg$). We follow Ren et al. (2020) by transforming a given EPFO query into Disjunctive Normal Form (DNF, Davey and Priestley, 2002), i.e. a disjunction of conjunctive queries, along with the subsequent extension with negations in (Ren and Leskovec, 2020). See fig. 6.1 for an example.

6.3.3 Continuous Query Decomposition

CQD is a framework for answering EPFO logical queries in the presence of missing edges. Given a query $\mathcal{Q}$, CQD defines the score of a target node $a \in \mathcal{E}$ as a candidate answer for a query as a function of the score of all atomic queries in $\mathcal{Q}$, given a variable-to-entity substitution for all variables in $\mathcal{Q}$.

Each variable is mapped to an *embedding vector* that can either correspond to an entity $c \in \mathcal{E}$ or to a virtual entity. The score of each of the query atoms is determined individually using a neural link predictor (Nickel et al., 2016). Then, the query score for a given candidate answer $\mathcal{Q}[a]$ is computed by aggregating all of the atom scores using t-norms and t-conorms – continuous relaxations of the logical conjunction and disjunction operators.

6.3.4 T-norms and Negations

Fuzzy logic generalises over Boolean by extending the operators of conjunction ($\wedge$), disjunction ($\vee$) and negation ($\neg$) through the use of t-norms and fuzzy negations. A *t-norm* $\top : [0,1] \times [0,1] \mapsto [0,1]$ is a generalisation of conjunction in fuzzy logic (Klement et al., 2000, 2004). Some examples include the *Gödel t-norm* $\top_{min}(x,y) = \min\{x,y\}$, the *product t-norm* $\top_{prod}(x,y) = x \times y$, and the *Łukasiewicz t-norm* $\top_{Luk}(x,y) = \max\{0, x+y-1\}$.

Analogously, *t-conorms* are dual to t-norms for disjunctions – given a t-norm $\top$, the complementary t-conorm is defined by $\bot(x,y) = 1 - \top(1-x, 1-y)$. Fuzzy logic also encompasses negations $n : [0,1] \mapsto [0,1]$. The *standard* $n_{stand}(x) = 1 - x$ and *strict cosine* $n_{cos} = \frac{1}{2}(1 + \cos(\pi x))$ are common examples of fuzzy negations. In our experiments, we use Gödel t-norm and product t-norm with their corresponding conorms, and *standard* and *strict cosine* as fuzzy negations.

6.3.5 Neural Link Predictors

A neural link predictor is a differentiable model where atom arguments are first mapped into a k-dimensional embedding space and then used to produce a score for the atom. More formally, given a query atom $p(s, o)$, where $p \in \mathcal{R}$ and $s, o \in \mathcal{E}$, the score for $p(s, o)$ is computed as $\phi_p(\mathbf{e}_s, \mathbf{e}_o)$, where $\mathbf{e}_s, \mathbf{e}_o \in \mathbb{R}^k$ are the embedding vectors of s and o, and $\phi_p : \mathbb{R}^k \times \mathbb{R}^k \mapsto [0, 1]$ is a *scoring function* computing the likelihood that entities s and o are related by the relationship p. Following Arakelyan et al. (2021); Minervini et al. (2022), in our experiments, we use a regularised variant of ComplEx (Trouillon et al., 2016; Lacroix et al., 2018) as the neural link predictor of choice, due to its simplicity, efficiency, and generalisation properties (Ruffinelli et al., 2020). To ensure that the output of the neural link predictor is always in $[0, 1]$, following Arakelyan et al. (2021); Minervini et al. (2022), we use a sigmoid function or min-max re-scaling.

6.3.6 Continuous Query Decomposition

Let $\mathcal{Q}$ denote the following DNF query:

$$\begin{aligned} \mathcal{Q}[A] \triangleq ?\, A : \exists V_1, \ldots, V_m . & \left(e_1^1 \wedge \ldots \wedge e_{n_1}^1\right) \vee \ldots \\ & \ldots \vee \left(e_1^d \wedge \ldots \wedge e_{n_d}^d\right), \\ & \textit{where } e_i^j = p(c, V), \\ & \quad \textit{with } V \in \{A, V_1, \ldots, V_m\}, c \in \mathcal{E}, p \in \mathcal{R}, \\ & \textit{or } e_i^j = p(V, V'), \\ & \quad \textit{with } V, V' \in \{A, V_1, \ldots, V_m\}, V \neq V', p \in \mathcal{R}. \end{aligned} \tag{6.2}$$

CQD aims to find the variable assignments that render $\mathcal{Q}$ true. To achieve this, CQD casts the problem of answering the query in eq. (6.2) as an optimisation problem, where the aim is finding a mapping from variables to entities that *maximises* the score of $\mathcal{Q}$:

$$\begin{aligned} \underset{A, V_1, \ldots, m \in \mathcal{E}}{\arg\max} & \left(e_1^1 \top \ldots \top e_{n_1}^1\right) \perp \ldots \perp \left(e_1^d \top \ldots \top e_{n_d}^d\right) \\ & \textit{where } e_i^j = \phi_p(\mathbf{e}_c, \mathbf{e}_V), \\ & \quad \textit{with } V \in \{A, V_1, \ldots, V_m\}, c \in \mathcal{E}, p \in \mathcal{R} \\ & \textit{or } e_i^j = \phi_p(\mathbf{e}_V, \mathbf{e}_{V'}), \\ & \quad \textit{with } V, V' \in \{A, V_1, \ldots, V_m\}, V \neq V', p \in \mathcal{R}, \end{aligned} \tag{6.3}$$

where $\top$ and $\perp$ denote a t-norm and a t-conorm – a continuous generalisation of the logical conjunction and disjunction, respectively – and $\phi_p(\mathbf{e}_s, \mathbf{e}_o) \in [0, 1]$ denotes the neural link prediction score for the atom $p(s, o)$.

Arakelyan et al. (2021) propose two possible strategies for solving the optimisation problem in eq. (6.3) – one is based on a greedy combinatorial search of the optimal variable-to-entity mappings, while the other is based on a gradient-based continuous optimisation process of the variable embeddings. In their results, Arakelyan et al. (2021); Minervini et al. (2022) show that the former discrete search strategy systematically produces significantly better results than the latter. Hence, we use it in this work as the query-answering process of choice.

6.3.7 Complex Query Answering via Combinatorial Optimisation

Following Arakelyan et al. (2021); Minervini et al. (2022), we solve the optimisation problem in eq. (6.3) by greedily searching for a set of variable substitutions $S = \{A \leftarrow a, V_1 \leftarrow v_1, \ldots, V_m \leftarrow v_m\}$, with $a, v_1, \ldots, v_m \in \mathcal{E}$, that maximises the complex query score, in a procedure akin to *beam search*. We do so by traversing the dependency graph of a query $\mathcal{Q}$ and, whenever we find an atom in the form $p(c, V)$, where $p \in \mathcal{R}$, c is either an entity or a variable for which we already have a substitution, and V is a variable for which we do not have a substitution yet, we replace V with all entities in $\mathcal{E}$ and retain the top-k entities $t \in \mathcal{E}$ that maximise $\phi_p(\mathbf{e}_c, \mathbf{e}_t)$ – i.e. the most likely entities to appear as a substitution of V according to the neural link predictor. As we traverse the dependency graph of a query, we keep a beam with the most likely variable-to-entity substitutions identified so far.

Example 6.3.1 (Combinatorial Optimisation)**.** Consider the query "*Which musicians M received awards associated with a genre g?*, which can be rewritten as $?M : \exists A.received(M, A) \wedge assoc(A, g)$. To answer this query using combinatorial optimisation, we must find the top-k awards a that are candidates to substitute the variable A in $received(A, g)$. This will allow us to understand the awards associated with the genre g. Afterwards, for each candidate substitution for A, we search for the top-k musicians m that are most likely to substitute M in $interacts(M, A)$, ending up with at most k^2 candidate musicians. We rank the candidate musicians' M using the final query score produced by a t-norm. ■

6.4 Adapting Neural Link Prediction Scores

The main limitation in the CQD method outlined in section 6.3 is that neural link predictors ϕ are trained to answer simple, atomic queries, and the resulting answer scores are not trained to interact with one another.

Example 6.4.1. Consider the running example query "*Which people are male and produced the music for the film Constantine?*" which can be rewritten as a complex query $\mathcal{Q} \equiv ?T : gender(male, T) \wedge producerOf(Constantine, T)$. To answer this complex query, CQD answers the atomic sub-queries $\mathcal{Q}_1 = gender(male, T)$ and $\mathcal{Q}_2 = producerOf(Constantine, T)$ using a neural link predictor, and aggregates the resulting scores using a t-norm. However, the neural link predictor was only trained on answering atomic queries, and the resulting scores are not calibrated to interact with each other. For example, the scores for the atomic queries about the relations $gender$ and $producerOf$ may be on different scales, which causes problems when aggregating such scores via t-norms. Let us assume the top candidates for the variable T coming from the atomic queries $\mathcal{Q}_1, \mathcal{Q}_2$ are $\mathcal{A}_1 \leftarrow$ *Sam Shepard* and $\mathcal{A}_2 \leftarrow$ *Klaus Badelt*, with their corresponding neural link prediction scores 1.2 and 8.9, produced using $\phi_itgender$, $\phi_itproducerOf$. We must also factor in the neural link prediction score of the candidate $\mathcal{A}_1$ for query $\mathcal{Q}_2$ at 7.4 and vice versa at 0.5. When using the Gödel t-norm $\top_{min}(x, y) = \min\{x, y\}$, the scores associated with the variable assignments $\mathcal{A}_1, \mathcal{A}_2$ are computed as, $\min(8.0, 0.5) = 0.5 \min(7.4, 1.2) = 1.2$. We see that for both answers $\mathcal{A}_1$ and $\mathcal{A}_2$, the scores produced by $\phi_itgender$ for $\mathcal{Q}_1$ are always lower than the scores produced with $\phi_itproducerOf$ for $\mathcal{Q}_2$, meaning that the scores of the latter are not considered when producing the final answer. This phenomenon can be broadly observed in CQD, illustrated in fig. 6.2. We can see that smaller values dominate the aggregated results, meaning that atomic relations with higher distribution bounds tend not to be considered during the complex query-answering process. ■

To this end, we propose a method for adaptively learning to re-calibrate neural link prediction scores by back-propagating through the complex query-answering process. More formally, let ϕ_p denote a neural link predictor. We learn an additional adaptation function ρ_θ, parameterised by θ, and use the composition of ρ_θ and ϕ_p, $\rho_\theta \circ \phi_p$, such that:

$$\rho_\theta(\phi_p(\mathbf{e}_V, \mathbf{e}_{V'})) = \phi_p(\mathbf{e}_V, \mathbf{e}_{V'})(1 + \alpha) + \beta, \qquad (6.4)$$

where $\theta = \{\alpha, \beta\}$, with $\alpha, \beta \in \mathbb{R}$, are parameters of the adaptation function ρ. Here, the function ρ defines an affine transformation of the score and when the parameters $\alpha = \beta = 0$, the transformed score $\rho_\theta(\phi_p(\mathbf{e}_V, \mathbf{e}_{V'}))$ recovers the original scoring function. The parameters θ can then be conditioned on the representation of the predicate p or the entities V or V'. We experimented with several variants of this model by modelling the affine transformation parameters θ as a function of a) the predicate representation, i.e. $\theta = \mathbf{W}\mathbf{e}_p$, with $\mathbf{W} \in \mathbb{R}^{2 \times k}$; b) the source entity representation, i.e. $\theta = \mathbf{W}\mathbf{e}_c$, with $\mathbf{W} \in \mathbb{R}^{2 \times k}$; c) or both the source entity and the predicate representation, i.e. $\theta = \mathbf{W}[\mathbf{e}_p; \mathbf{e}_c]$, with $\mathbf{W} \in \mathbb{R}^{2 \times 2k}$.

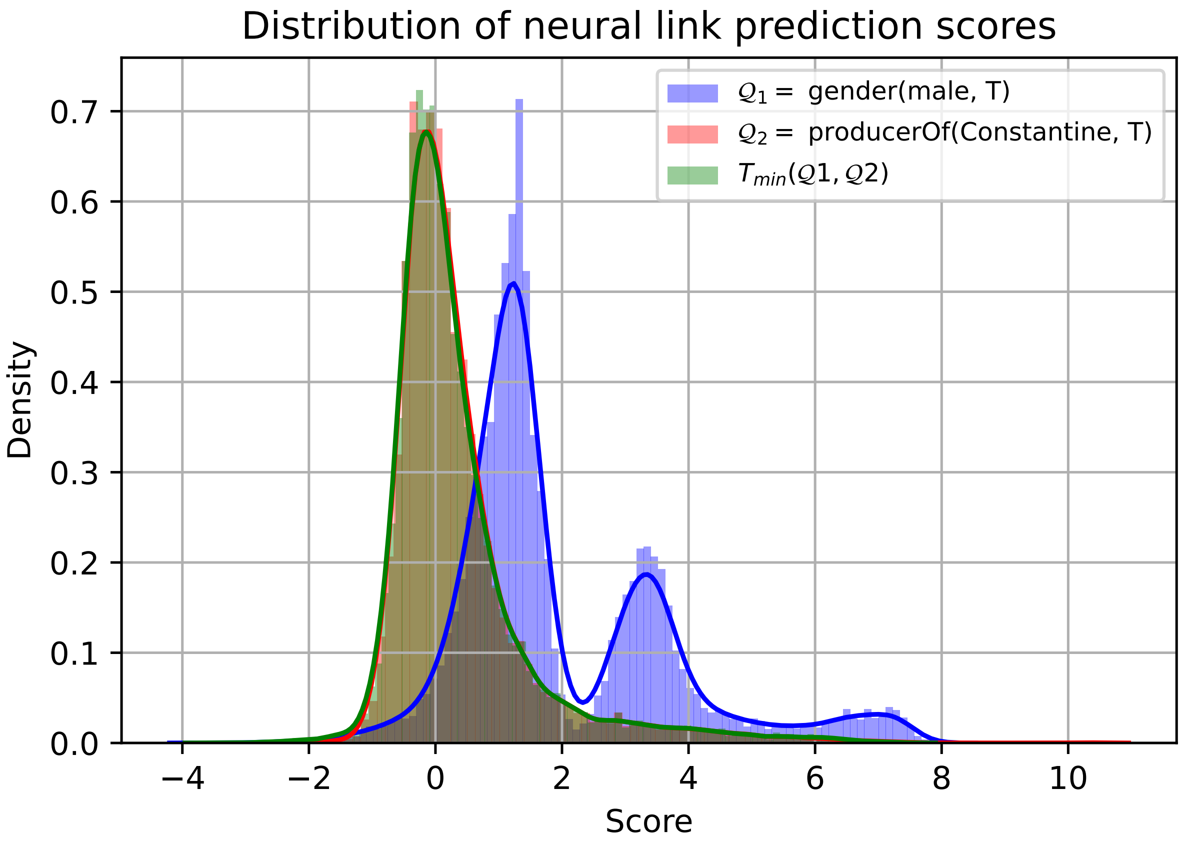


Figure 6.2: The distributions of two atomic scores and the $\top_{min}$ t-norm aggregated results – we can see that the scores from $\mathcal{Q}_2$ dominate the comparison.

6.4.1 Training

As mentioned in section 6.3, for answering a complex query $\mathcal{Q}$, CQD identifies the variable assignment $S = \{A \leftarrow a, V_1 \leftarrow v_1, \ldots, V_m \leftarrow v_m\}$, where $a, v_1, \ldots, v_m \in \mathcal{E}$ are entities and $A, V_1, \ldots, V_m$ are variables, that maximises the complex query score from eq. (6.3):

$$\begin{aligned} &score(\mathcal{Q}, S) \\ &\quad = \left(e_1^1 \top \ldots \top e_{n_1}^1\right) \bot \ldots \bot \left(e_1^d \top \ldots \top e_{n_d}^d\right), \end{aligned} \tag{6.5}$$

where $e_i^j = \phi_p(\mathbf{e}_c, \mathbf{e}_V)$ or $e_i^j = \phi_p(\mathbf{e}_V, \mathbf{e}_{V'})$ corresponds to the atomic query score produced by a neural link predictor, where and $\top$ and $\bot$ are t-norms and t-conorms, respectively. In eq. (6.5), $\mathbf{e}_c$ denotes the representation of the entity $c \in \mathcal{E}$, while $\mathbf{e}_V$ and $\mathbf{e}_{V'}$ denote the embedding of the entities associated with the variables V and V' in the substitution S. More formally, during the complex query-answering process, CQD solves the following optimisation problem:

$$\arg\max_S score(\mathcal{Q}, S), \tag{6.6}$$

Table 6.1: Statistics on the different types of query structures in FB15K, FB15K-237, and NELL995.

Split	Query Type	FB15K	FB15K-237	NELL995
Training	1p, 2p, 3p, 2i, 3i	273,710	149,689	107,982
	2in, 3in, inp, pin, pni	27,371	14,968	10,798
Validation	1p	59,078	20,094	16,910
	Others	8,000	5,000	4,000
Test	1p	66,990	22,804	17,021
	Others	8,000	5,000	4,000

where $S = \{A \leftarrow a, V_1 \leftarrow v_1, \ldots, V_m \leftarrow v_m\}$ is the resulting substitution that maximises the query score, where A is the answer variable.

For training the score calibration component in eq. (6.4), we first compute how likely each entity $a' \in \mathcal{E}$ is to be an answer to the query $\mathcal{Q}$. To this end, for each candidate answer $a' \in \mathcal{E}$, we compute the *answer score* as the complex query score assuming that $a' \in \mathcal{E}$ is the final answer as:

$$\begin{aligned} score(\mathcal{Q}, A \leftarrow a') = \max_{S} score(\mathcal{Q}, S) \\ where\ A \leftarrow a' \in S. \end{aligned} \tag{6.7}$$

eq. (6.7) identifies the variable-to-entity substitution S that 1) maximises the query score $score(\mathcal{Q}, S)$, provided in eq. (6.5), and 2) associates the answer variable A with $a' \in \mathcal{E}$, i.e. $A \leftarrow a' \in S$. For computing S with the additional constraint that $A \leftarrow a' \in S$, we use the complex query answering procedure outlined in section 6.3.

For training the additional parameters $\mathbf{W}$ introduced in section 6.4, we maximise the likelihood of the true answers on a dataset $\mathcal{D} = \{(\mathcal{Q}, a)\}$ of query-answer pairs by using a *one-versus-all* cross-entropy loss, introduced by Lacroix et al. (2018), which was also used to train the neural link prediction model:

$$\begin{aligned} \mathcal{L}(\mathcal{D}) = \sum_{(\mathcal{Q},a)\mathcal{D}} & -score(\mathcal{Q}, A \leftarrow a) \\ & + \log \left[\sum_{a' \in \mathcal{E}} \exp\left(score(\mathcal{Q}, A \leftarrow a')\right) \right]. \end{aligned} \tag{6.8}$$

In addition to the *one-versus-all* loss in eq. (6.8), we also use a binary cross-entropy loss, using the negative sampling procedure from Ren and Leskovec (2020).

Table 6.2: MRR results for FOL queries on the testing sets. **avg**$_p$ designates the averaged results for EPFO queries ($\wedge, \vee$), while **avg**$_n$ pertains to queries including negations ($\neg$). The results for CQD are taken from Minervini et al. (2022), while all the remaining come from Zhu et al. (2022b).

Model	avg$_p$	avg$_n$	1p	2p	3p	2i	3i	pi	ip	2u	up	2in	3in	inp	pin	pni
FB15K																
GQE	28.0	-	54.6	15.3	10.8	39.7	51.4	27.6	19.1	22.1	11.6	-	-	-	-	-
Q2B	38.0	-	68.0	21.0	14.2	55.1	66.5	39.4	26.1	35.1	16.7	-	-	-	-	-
BetaE	41.6	11.8	65.1	25.7	24.7	55.8	66.5	43.9	28.1	40.1	25.2	14.3	14.7	11.5	6.5	12.4
CQD-CO	46.9	-	**89.2**	25.3	13.4	74.4	78.3	44.1	33.2	41.8	21.9	-	-	-	-	-
CQD-Beam	58.2	-	**89.2**	54.3	28.6	74.4	78.3	58.2	67.7	42.4	30.9	-	-	-	-	-
ConE	49.8	14.8	73.3	33.8	29.2	64.4	73.7	50.9	35.7	55.7	31.4	17.9	18.7	12.5	9.8	15.1
GNN-QE	**72.8**	38.6	88.5	**69.3**	58.7	**79.7**	**83.5**	69.9	70.4	**74.1**	**61.0**	44.7	41.7	**42.0**	30.1	**34.3**
CQD$^{\mathcal{A}}$	70.4	**42.8**	**89.2**	64.5	57.9	76.1	79.4	**70.0**	**70.6**	68.4	57.9	**54.7**	**47.1**	37.6	**35.3**	24.6
FB15K-237																
GQE	16.3	-	35.0	7.2	5.3	23.3	34.6	16.5	10.7	8.2	5.7	-	-	-	-	-
Q2B	20.1	-	40.6	9.4	6.8	29.5	42.3	21.2	12.6	11.3	7.6	-	-	-	-	-
BetaE	20.9	5.5	39.0	10.9	10.0	28.8	42.5	22.4	12.6	12.4	9.7	5.1	7.9	7.4	3.5	3.4
CQD-CO	21.8	-	**46.7**	9.5	6.3	31.2	40.6	23.6	16.0	14.5	8.2	-	-	-	-	-
CQD-Beam	22.3	-	**46.7**	11.6	8.0	31.2	40.6	21.2	18.7	14.6	8.4	-	-	-	-	-
ConE	23.4	5.9	41.8	12.8	11.0	32.6	47.3	25.5	14.0	14.5	10.8	5.4	8.6	7.8	4.0	3.6
GNN-QE	**26.8**	10.2	42.8	**14.7**	**11.8**	**38.3**	**54.1**	**31.1**	18.9	16.2	**13.4**	10.0	16.8	9.3	7.2	**7.8**
CQD$^{\mathcal{A}}$	25.3	**10.9**	**46.7**	13.6	11.4	33.1	45.4	26.5	**20.4**	**17.5**	11.4	**13.6**	**16.8**	**9.5**	**8.9**	5.8
NELL995																
GQE	18.6	-	32.8	11.9	9.6	27.5	35.2	18.4	14.4	8.5	8.8	-	-	-	-	-
Q2B	22.9	-	42.2	14.0	11.2	33.3	44.5	22.4	16.8	11.3	10.3	-	-	-	-	-
BetaE	24.6	5.9	53.0	13.0	11.4	37.6	47.5	24.1	14.3	12.2	8.5	5.1	7.8	10.0	3.1	3.5
CQD-CO	28.8	-	**60.4**	17.8	12.7	39.3	46.6	30.1	22.0	17.3	13.2	-	-	-	-	-
CQD-Beam	28.6	-	**60.4**	20.6	11.6	39.3	46.6	25.4	23.9	17.5	12.2	-	-	-	-	-
ConE	27.2	6.4	53.1	16.1	13.9	40.0	50.8	26.3	17.5	15.3	11.3	5.7	8.1	10.8	3.5	3.9
GNN-QE	28.9	9.7	53.3	18.9	14.9	42.4	52.5	30.8	18.9	15.9	12.6	9.9	14.6	11.4	6.3	6.3
CQD$^{\mathcal{A}}$	**32.3**	**13.3**	**60.4**	**22.9**	**16.7**	**43.4**	**52.6**	**32.1**	**26.4**	**20.0**	**17.0**	**15.1**	**18.6**	**15.8**	**10.7**	**6.5**

6.5 Experiments

6.5.1 Datasets

To evaluate the complex query answering capabilities of our method, we use a benchmark comprising of 3 KGs: FB15K (Bordes et al., 2013), FB15K-237 (Toutanova and Chen, 2015) and NELL995 (Xiong et al., 2017). For a fair comparison with previous work, we use the datasets of FOL queries proposed by Ren and Leskovec (2020), which includes 9 structures of EPFO queries and 5 query types with negations, seen in fig. 6.3. The datasets provided by Ren and Leskovec (2020) introduce queries with *hard* answers, where the answers cannot be obtained by direct graph traversal and do not include queries with an excessive amount of feasible answers, increasing the difficulty of the complex query answering task. The statistics for each dataset can be seen in table 6.1.

Note that during training, we only use *2i*, *3i*, *2in*, and *3in* queries, $\leq 35\%$ of the training dataset for the Adaptation of the Neural Link Predictor. To assess the ability of the model to generalise, we evaluate it on all query types.

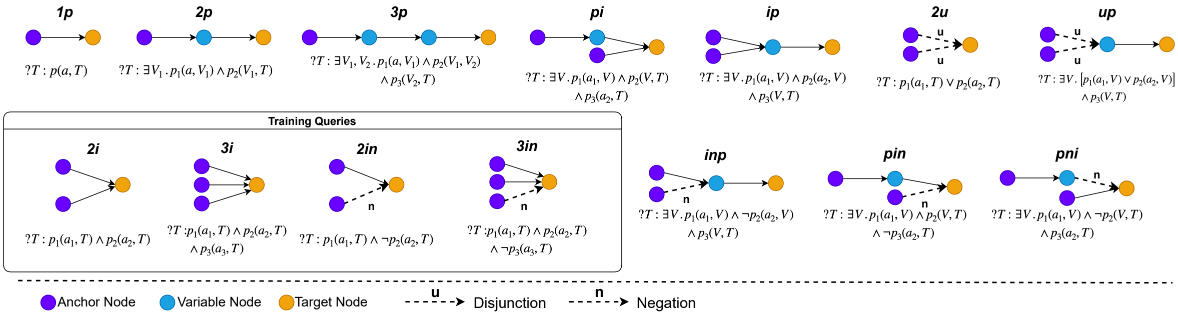

1p
$?T : p(a, T)$
2p
$?T : \exists V_1 . p_1(a, V_1) \land p_2(V_1, T)$
3p
$?T : \exists V_1, V_2 . p_1(a, V_1) \land p_2(V_1, V_2) \land p_3(V_2, T)$
pi
$?T : \exists V . p_1(a_1, V) \land p_2(V, T) \land p_3(a_2, T)$
ip
$?T : \exists V . p_1(a_1, V) \land p_2(a_2, V) \land p_3(V, T)$
2u
u
u
$?T : p_1(a_1, T) \lor p_2(a_2, T)$
up
u
u
$?T : \exists V . [p_1(a_1, V) \lor p_2(a_2, V)] \land p_3(V, T)$
Training Queries
2i
$?T : p_1(a_1, T) \land p_2(a_2, T)$
3i
$?T : p_1(a_1, T) \land p_2(a_2, T) \land p_3(a_3, T)$
2in
n
$?T : p_1(a_1, T) \land \neg p_2(a_2, T)$
3in
n
$?T : p_1(a_1, T) \land p_2(a_2, T) \land \neg p_3(a_3, T)$
inp
n
$?T : \exists V . p_1(a_1, V) \land \neg p_2(a_2, V) \land p_3(V, T)$
pin
n
$?T : \exists V . p_1(a_1, V) \land p_2(V, T) \land \neg p_3(a_2, T)$
pni
n
$?T : \exists V . p_1(a_1, V) \land \neg p_2(V, T) \land p_3(a_2, T)$
● Anchor Node
● Variable Node
● Target Node
u
Disjunction
n
Negation

Figure 6.3: Query structures considered in our experiments, as proposed by Ren and Leskovec (2020) – the naming of each query structure corresponds to *projection* (**p**), *intersection* (**i**), *union* (**u**) and *negation* (**n**), reflecting how they were generated in the BetaE paper (Ren and Leskovec, 2020). An example of a **pin** query is $?T : \exists V.p(a, V), q(V, T), \neg r(b, T)$, where a and b are anchor nodes, V is a variable node, and T is the query target node.

6.5.2 Evaluation Protocol

For a fair comparison with prior work, we follow the evaluation scheme in Ren and Leskovec (2020) by separating the answer of each query into *easy* and *hard* sets. For test and validation splits, we define *hard* queries as those that cannot be accessed via direct search along the KG and should be reached by predicting a missing link, meaning *non-trivial* reasoning should be completed. We evaluate the method on non-trivial queries by calculating the rank r for each hard answer against non-answers and computing the Mean Reciprocal Rank (MRR).

6.5.3 Baselines

We compare CQD$^{\mathcal{A}}$ with state-of-the-art methods from various solution families presented in section 6.2. In particular, we choose GQE (Hamilton et al., 2018), Query2Box (Ren et al., 2020), BetaE (Ren and Leskovec, 2020) and ConE (Zhang et al., 2021) as strong baselines for embedding methods. We also compare with methods that utilise Fuzzy Logic and GNNs for enhancing the properties of the embedding space, such as FuzzQE (Chen et al., 2022) and the current state-of-the-art GNN-QE (Zhu et al., 2022b). Finally, we evaluate the method against CQD (Arakelyan et al., 2021) that uses neural link predictors for answering EPFO queries.

6.5.4 Model Details

Our method is model-agnostic and can be used with any neural link prediction model. Following Arakelyan et al. (2021); Minervini et al. (2022), we use ComplEx-N3 (Lacroix et al., 2018). We identify the optimal hyper-parameters using the validation MRR. We train for 50000 steps using Adagrad as an optimiser and 0.1 as the learning rate. The beam-size hyper-parameter k was selected in $k \in \{512, 1024, \ldots, 8192\}$,

Table 6.3: The number of parameters of other methods is approximated based on the dimensionality of entity/relation representations chosen by the designated method and excluding any learnable operators/functions. The numbers for GNN-QE are approximated using the backbone NBFNet (Zhu et al., 2021), while the remaining use their original studies.

	Number of parameters		
	FB15K	FB15K-237	NELL
CQD$^{\mathcal{A}}$	$\underbrace{1.3 \times 10^7}_{frozen}$ $+\mathbf{4 \times 10^3}$	$\underbrace{1.3 \times 10^7}_{frozen}$ $+\mathbf{4 \times 10^3}$	$\underbrace{7.5 \times 10^7}_{frozen}$ $+\mathbf{4 \times 10^3}$
BetaE	1.3×10^7	1.3×10^7	6×10^7
Q2B	1.2×10^7	1.2×10^7	6×10^7
GNN-QE	3×10^6	3×10^6	3×10^6
ConE	1.2×10^7	1.2×10^7	6×10^7
GQE	1.5×10^7	1.5×10^7	7.5×10^7

and the loss was selected across *one-versus-all* (Lacroix et al., 2018) and binary cross-entropy with one negative sample.

6.5.5 Parameter Efficiency

We use the query types *2i*, *3i*, *2in*, *3in* for training the calibration module proposed in section 6.4. We selected these query types as they do not require variable assignments other than for the answer variable A, making the training process efficient. As the neural link prediction model is frozen, we only train the adapter layers that have a maximum of $\mathbf{W} \in \mathbb{R}^{2\times 2k}$ learnable weights. Compared to previous works, we have $\approx 10^3$ times fewer trainable parameters, as shown in table 6.3, while maintaining competitive results.

During inference, for queries requiring multiple hops of reasoning, we confine the beam size of the search, meaning that all the candidate branches can fit on a single GPU, allowing for faster computation.

Results

6.5.6 Complex Query Answering

table 6.2 shows the predictive accuracy of CQD$^{\mathcal{A}}$ for answering complex FOL queries compared to the current state-of-the-art methods. Some methods do not support queries that include negations, so we leave the corresponding entries blank. We can see that CQD$^{\mathcal{A}}$ achieves an increase from 34.4 to 35.1 averaged across all query types and datasets. In particular, CQD$^{\mathcal{A}}$ shows the most substantial increase in predictive accuracy on NELL995 by producing more accurate results than all other methods for all query types. CQD$^{\mathcal{A}}$ can achieve these results while using $\leq 35\%$ of the complex query types during training while maintaining competitive results across each of the datasets and query types. In our experiments, we found that calculating the neural adaptation parameters θ of the adaptation function ρ_θ in section 6.4 as a function of the predicate representation yields the most accurate results followed by computing θ as a function of the source entity and predicate representation, which is strictly more expressive. In section 6.7.1, we show the impact of the adaptation layers on the neural link prediction scores.

Additionally, the adaptation process does not require data-intensive training and allows the model to generalise to queries it was not explicitly trained on. This prompts us to investigate the minimal amount of data samples and query types required for adaptation. We explore the full scale of the data efficiency and generalisation robustness of the method in the following paragraphs.

6.5.7 Negations

We can observe from table 6.2 that the method can achieve a relative improvement of 6.8% to 37.1% for queries, including negations. We attribute this to the fact that the training included query types *2in* and *3in* that include negation, which allows for learning an adaptation layer that is robust for these types of queries.

6.5.8 Data Efficiency

One benefit of our method is that adapting the scores of the neural link predictor does not require a large amount of data. To this end, we systematically study the data-efficiency of CQD$^{\mathcal{A}}$ by comparing the behaviour of the models trained with 1% and 100% of the training complex query examples in FB15K-237, presented in fig. 6.4. For adapting on 1% of the training complex queries, we used the same hyper-parameters we identified when training on the full dataset. Even when using 1% of the dataset from the training queries (3290 samples), the model still achieves competitive results, with an average MRR difference of 2.4 in comparison with the model trained using the full training set.

We can also confirm that the adaptation process converges after $\leq 10\%$ of the training epochs as seen in fig. 6.5. The rate of convergence is not hindered when using only 1% of the training queries. This shows that CQD$^{\mathcal{A}}$ is a scalable method with a fast convergence rate that can be trained in a data-efficient manner.

6.5.9 Out-of-Distribution Generalisation

To study the generalisation properties of CQD$^{\mathcal{A}}$, we trained the adaptation layer on only 1% of samples for *one* training query type *2i*, which does not require a greedy

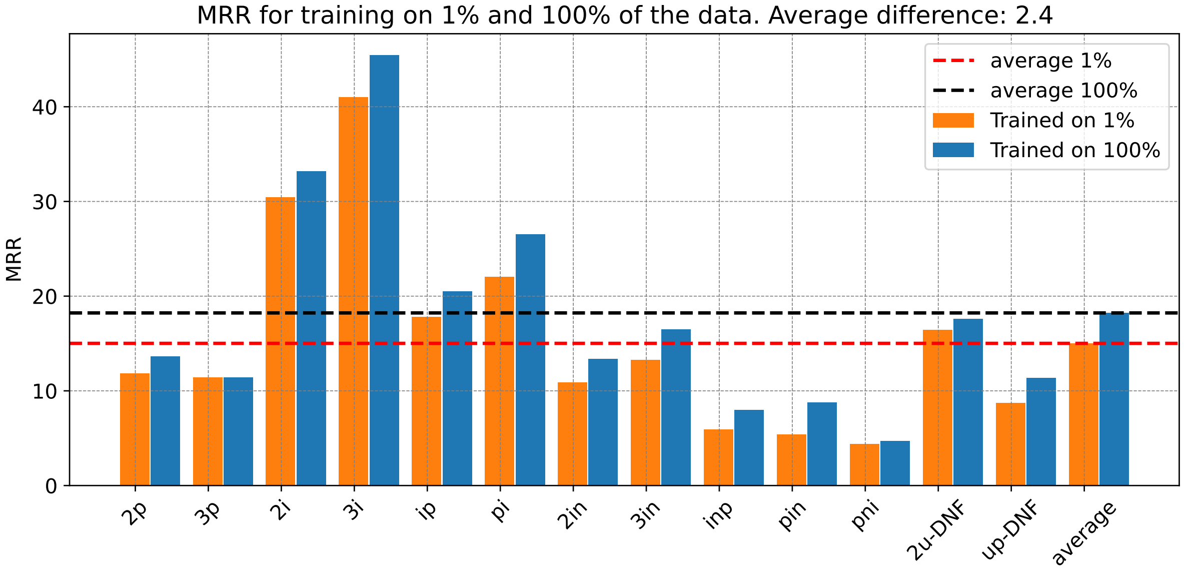


Figure 6.4: Evaluation of CQD$^{\mathcal{A}}$ using 1% and 100% of the training queries from FB15K-237. The link prediction task *1p* does not require adaptation and is thus omitted.

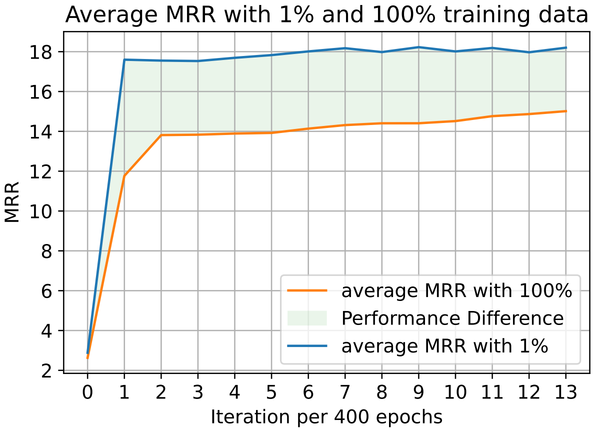


Figure 6.5: Average MRR score increase of CQD$^{\mathcal{A}}$ using 1% and 100% of the training queries from FB15K-237 throughout the epochs. The X-axis corresponds to 10% of training epochs.

beam search during the complex query answering process. We see in fig. 6.6 that CQD$^{\mathcal{A}}$ can generalise to other types of complex queries not observed during training with an average MRR difference of 2.9 compared to training on all of the training query types. The greatest degradation in predictive accuracy occurs for the queries containing negations, with an average decrease of 2.7. This prompts us to conjecture that being able to answer general EPFO queries may not be enough to generalise to the complete set of FOL queries. However, our method can competitively generalise on all FOL queries, using only 1% of *2i* queries, with 1496 overall samples for adaptation.

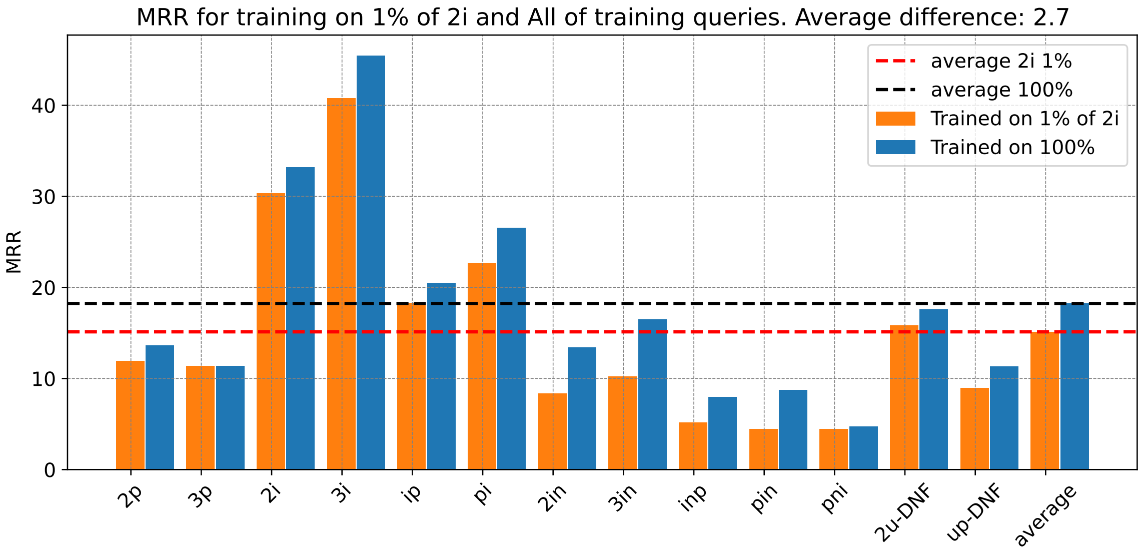


Figure 6.6: Evaluation of CQD$^{\mathcal{A}}$ using 1% of *2i* and all of the training queries from FB15K-237. The link prediction task *1p* does not require adaptation and is thus omitted.

6.6 Conclusions

In this work, we propose the novel method CQD$^{\mathcal{A}}$ for answering complex FOL queries over KGs, which achieves an increase in averaged MRR over the previous state-of-the-art from 34.4 to 35.1 while using $\leq 35\%$ of query types. Our method uses a single adaptation layer over neural link predictors, which allows for training in a data-efficient manner. We show that the method can maintain competitive predictive accuracy even when using 1% of the training data. Furthermore, our experiments on training on a subset (1%) of the training queries from a single query type (*2i*) show that it can generalise to new queries that were not used during training while being data-efficient.

6.7 Appendix

6.7.1 Impact of adaptation

We investigate the effect of the adaptation process in CQD$^{\mathcal{A}}$ by comparing the score of the neural link predictor before and after applying the adaptation layer. As we see from fig. 6.7, the scores before adaptation have a variation of 5.04 with the boundaries at $[-8, 12]$. This makes them problematic for complex query answering as discussed in section 6.4. The Adapted scores have a smaller variation at 0.03 while the maximum and minimum lie in the range $[0, 1]$.

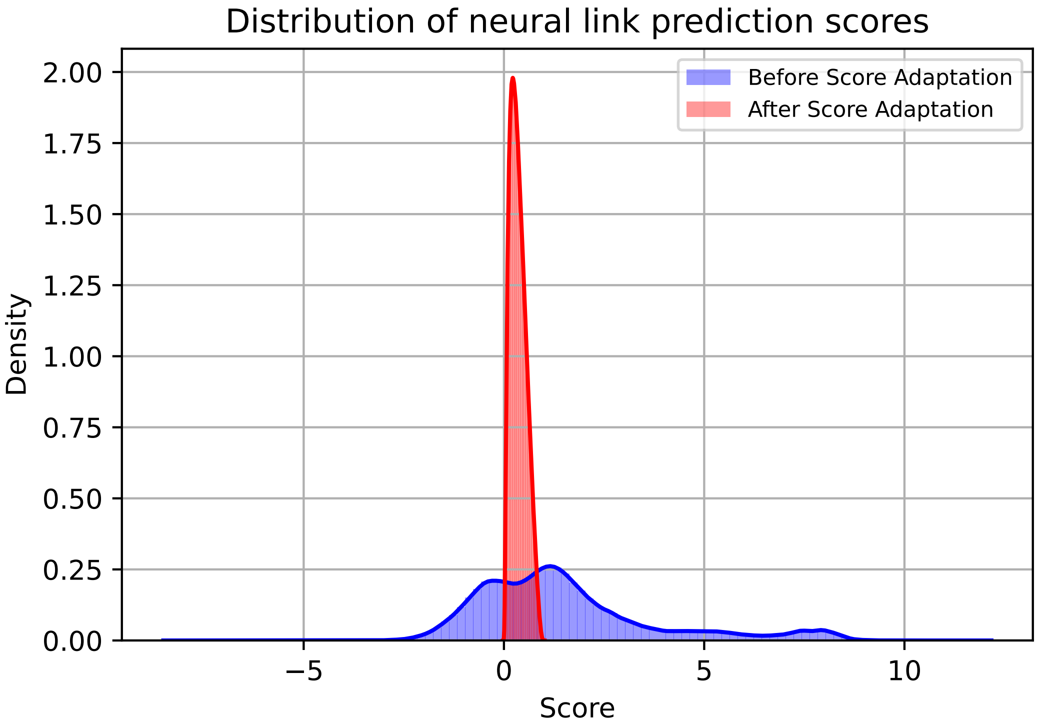


Figure 6.7: The distribution of the scores of the neural link predictor before applying the adaptation layer and after.

FLARE: Faithful Logic-Aided Reasoning and Exploration

7

7.1 Introduction

Complex Reasoning in natural Question Answering (QA) tasks assumes the capability to explore the problem space of the designated query with a formalised set of facts, relations, commonsense knowledge and logical implications. In line with this, LLMs have been enhanced with CoT (Wei et al., 2022b) prompting, which supplements the QA process by generating intermediate reasoning chains given a set of in-context examples (Brown et al., 2020a), as shown in fig. 7.1. This allowed for advancement in commonsense (Madaan et al., 2022), symbolic (Wang et al., 2022b; Sprague et al., 2024) and mathematical (Jie et al., 2023) reasoning. Although CoT allows for a problem exploration in natural language steps, such an approach has been shown to cause performance degradation for reasoning tasks involving multi-step planning (Valmeekam et al., 2022; Suzgun et al., 2023), problem exploration (Yao et al., 2022), and arithmetic tasks (Hendrycks et al., 2021c; Madaan and Yazdanbakhsh, 2022a). These discrepancies arise as CoT suffers from a limited ability to decompose, search, verify and backtrack using intermediate rationale chains (Yao et al., 2022), cascading hallucinations and errors (Ling et al., 2023) and that natural language might not be an optimal representation for describing the reasoning process (Li et al., 2024). Simultaneously, LLM output has been shown to be unfaithful and inconsistent w.r.t. the intermediate CoT rationale (Jacovi et al., 2024; Lanham et al., 2023a; Turpin et al., 2023).

To mitigate the problem of CoT faithfulness and allow for more robust reasoning during QA, Lyu et al. (2023, Faithful CoT) and Logic-LM (Pan et al., 2023) suggested generating code which is further executed using an external symbolic solver. Producing and executing code enables the generation of outputs guided by external solvers, leveraging search with backtracking to explore the problem space effectively. However, strict translations of natural language queries into code, such as *autoformalisation* (Szegedy, 2020; Wang et al., 2018b), is a non-trivial task involving direct inference of implicit commonsense and domain-specific knowledge and the ability to align abstract and informal concepts directly to constrained formal definitions for further execution (Wu et al., 2022). An example query, *"Do all parts of the aloe vera plant taste good?"*, is challenging to formalize or address with a strict algorithmic

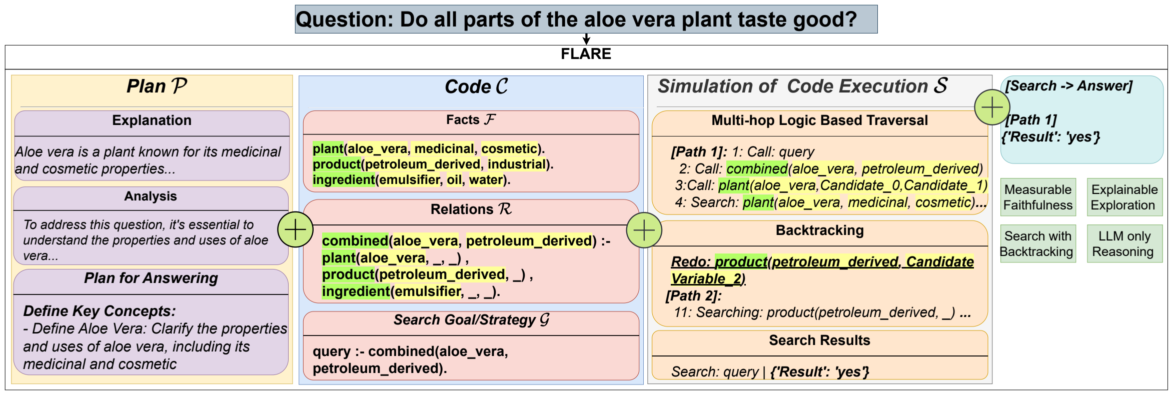

Question: Do all parts of the aloe vera plant taste good?
FLARE
Plan $\mathcal{P}$
Explanation
Aloe vera is a plant known for its medicinal and cosmetic properties...
Analysis
To address this question, it's essential to understand the properties and uses of aloe vera...
Plan for Answering
Define Key Concepts:
- Define Aloe Vera: Clarify the properties and uses of aloe vera, including its medicinal and cosmetic
Code $\mathcal{C}$
Facts $\mathcal{F}$
plant(aloe_vera, medicinal, cosmetic).
product(petroleum_derived, industrial).
ingredient(emulsifier, oil, water).
Relations $\mathcal{R}$
combined(aloe_vera, petroleum_derived) :-
plant(aloe_vera, _, _) ,
product(petroleum_derived, _) ,
ingredient(emulsifier, _, _).
Search Goal/Strategy $\mathcal{G}$
query :- combined(aloe_vera,
petroleum_derived).
Simulation of Code Execution $\mathcal{S}$
Multi-hop Logic Based Traversal
[Path 1]: 1: Call: query
2: Call: combined(aloe_vera, petroleum_derived)
3:Call: plant(aloe_vera,Candidate_0,Candidate_1)
4: Search: plant(aloe_vera, medicinal, cosmetic)...
Backtracking
Redo: product(petroleum_derived, Candidate Variable_2)
[Path 2]:
11: Searching: product(petroleum_derived, _) ...
Search Results
Search: query | {'Result': 'yes'}
[Search -> Answer]
[Path 1]
{'Result': 'yes'}
Measurable Faithfulness
Explainable Exploration
Search with Backtracking
LLM only Reasoning

Figure 7.1: A depiction of the *plan*, *code* and simulated *search* in FLARE. Each module is generated separately and iteratively, allowing us to obtain the final answer. The green and yellow highlighted text shows the overlap between the facts and the relations between the code and the simulated search.

solution, as it requires interpretative, deductive and context-dependent reasoning, referred to as soft or fuzzy reasoning. Using external solvers makes such fuzzy reasoning impossible and requires consistently generating syntactically correct executable code. While some LLMs have coding capabilities stemming from their pretraining (Jiang et al., 2024; Aryabumi et al., 2024), relative code consistency is more probable with models explicitly trained for coding (Chen et al., 2021).

To overcome these problems, we propose Faithful Logic-Aided Reasoning and Exploration (FLARE), an interpretable method that allows for planning, fuzzy reasoning, and traversing the problem space with backtracking, exact task decomposition, and measuring faithfulness. In FLARE, given a natural language query, we prompt an LLM to sequentially generate a *plan* that includes an analysis and the logical steps necessary for formalising and answering the question, a logic programming (Wielemaker et al., 2012) *code* that allows formalising the query into a set of facts, relations and their composition forming the space for exploring that query and the *search*, which is an LLM-generated code execution simulation. An illustration of FLARE can be seen in fig. 7.1. In our framework, the generated code must not be consistently executable by an external solver, allowing for the soft-formalisation of natural language. Although we see that even generalist LLMs are able to produce executable code in $\geq 50\%$ of cases. FLARE allows us to measure the faithfulness of the outcome w.r.t. the simulated code execution by directly comparing the search paths produced by the external solver to that LLM generation. This comparison also allows for pinpointing model hallucinations and inconsistencies. We systematically study the effectiveness of our method using 4 general-purpose LLMs of varying scales across 9 diverse QA and 3 logical inference benchmarks, covering Math World Problems, Multi-hop QA, Relation inference, deductive and analytical reasoning and show that our method achieves state-of-the-art results in 7 out of 9 QA datasets and 2 out of 3 logic datasets in comparison to CoT, F-CoT and Logic-LM. We also show that the method is competitive for models tuned for coding, with an average overall increase of 16% over F-Cot and 9% over CoT. Our findings show that model accuracy strongly correlates with the faithfulness of the reasoning process towards search traces from the simulated code execution. We also provide ablations showing that the model can interpretably pinpoint hallucinations, underutilized knowledge, and the limitations of the search over the problem space. Our key contributions are the following:

- We introduce FLARE a novel paradigm for logic-aided and interpretable formalisation and search over the problem space in QA and logic reasoning tasks.
- We perform a systematic evaluation across 9 QA and 3 logical inference benchmarks and 4 models of varying scales, showing the advantages of using FLARE for QA in a few-shot setup over prior approaches.
- The modularity of FLARE allows defining a simple ingrained method for measuring model faithfulness, which is further shown to be strongly correlated with performance.
- We further show that using FLARE allows us to interpretably and rigorously detect hallucinations along with sub-optimal and inconsistent reasoning patterns.

7.2 Related Work

7.2.1 Reasoning in Natural Language

Few-shot prompting (Brown et al., 2020c) has been shown to be an effective approach for increasing the reasoning capabilities of LLMs in natural language generation (Gehrmann et al., 2021; Reif et al., 2022; Sanh et al., 2022). LLM reasoning can be further enhanced with prompting techniques such as CoT (Wei et al., 2022b), which attempts to segment reasoning into explicitly written intermediate steps. Concurrent work has also proposed that models *"think step by step"* (Kojima et al., 2022b), or divide the problem into subtasks before the solution (Zhou et al., 2023a, Least-to-Most). These approaches have been shown to suffer from arithmetic inaccuracies (Lewkowycz et al., 2022; Hendrycks et al., 2021b) and reasoning inconsistencies (Madaan and Yazdanbakhsh, 2022b). Further attempts have been made to add a planning stage before reasoning by dividing the process into recursive plan formulation and execution steps (Yao et al., 2023b; Wang et al., 2023a). The *plan* generation step in FLARE is a hybrid technique inspired by these methods but focused on generating a natural language strategy for formalising the query into code.

7.2.2 Reasoning with Search

Several lines of work propose using techniques to expand the reasoning paths over the problem space. Self-consistency decoding (Wang et al., 2023b) is an approach used to sample many natural language reasoning paths and take a majority vote for an answer. Another popular approach is Tree-of-Thoughts (ToT; Yao et al., 2023a), which proposes to explore the query with reasoning similar to a tree traversal, where each state is created and evaluated using an LLM. Similar techniques try to adapt symbolic search approaches akin to DFS, BFS (Besta et al., 2024), A* (Lehnert et al., 2024) or other combinations (Gandhi et al., 2024) with direct tuning (Lehnert et al., 2024), imitation training (Yang et al., 2022) or few-shot prompting (Zhang et al., 2024). It must be noted that all of these techniques have only been tested in constrained mathematical puzzle-solving and algorithmic domains like the 24 Game (Yang et al., 2022), Countdown (Wikipedia, 2024), Sorting (Besta et al., 2024), maze solving (Yang et al., 2022), Sokoban (Lehnert et al., 2024), and others. Although the *search* component of FLARE has some similarities to these techniques, we argue that our method allows for generalistic reasoning with interpretable multi-hop search through simulated code execution.

7.2.3 Reasoning with Formalisation

Another line of research has tried formalising natural language queries into code (Gao et al., 2023; Li et al., 2024) or pseudo-code (Chae et al., 2024; Gandhi et al., 2024). This allows the translation of the query into a strict structure and offloads the reasoning and search components to deterministic solvers like Python (Chen et al., 2023), PDDL (Lyu et al., 2023; Liu et al., 2023), DataLog (Lyu et al., 2023) and others. While models are capable of synthesising programs (Austin et al., 2021; Nijkamp et al., 2023) and benefit from the use of code in numerical and algorithmic reasoning settings (Chen et al., 2023; Gao et al., 2023), the usage of code for general QA has not been rigorously explored. The reasons are that formalisation from natural language into a strict and executable code is challenging (Wu et al., 2022), following the exact syntactic constraints of the programming language not abundantly used during pre-training is onerous (Liu et al., 2024) and can require models explicitly tuned for coding (Chen et al., 2021). Using an external solver for reasoning also limits the capability for soft reasoning in commonsense knowledge and implications. Although we formalise the natural language query into a logic programming Prolog program during the *code* generation part of FLARE, we do not explicitly require the code to be executable and do not use external solvers during inference. This allows for the further use of the LLM for soft-reasoning to simulate code execution in a logic-based

	Math Word Problems					Multi-hop QA			Relation
Method	GSM8K	SVAMP	MultiArith	ASDiv	AQuA	StrategyQA	Date	Sport	CLUTRR
$Llama-3.1-8B_{FLARE}$	**72.7**	**86.0**	**96.3**	**83.1**	**62.9**	**70.2**	**59.3**	76.6	36.8
$Llama-3.1-8B_{F-CoT}$	it0	it0	it0	it0	it12.2	53.2	it0	it0	it32
$Llama-3.1-8B_{CoT}$	59.2	58.6	60.1	61.9	35	it2.9	20.9	**95.8**	**42.2**
$CmDR_{FLARE}$	**52.4**	**74.0**	**84.5**	**72.2**	**43.7**	**67.0**	**52.3**	**78.9**	29.1
$CmDR_{F-CoT}$	it0	it0	it0	it0	it0	59.7	it0	it0	it8.6
$CmDR_{CoT}$	46.5	57.3	83.1	37.2	28.3	it21.3	47.4	55.2	**29.5**
$CmDR+_{FLARE}$	**71.4**	**83.5**	**90.4**	**81.3**	**55.9**	**70.8**	61.8	**77.7**	**41.0**
$CmDR+_{F-CoT}$	it0	it0	it0	it0	it15.4	57.6	it0	it0	it35.3
$CmDR+_{CoT}$	48.7	81.1	86.6	44.6	44.1	it48.4	**79.1**	62.6	42.5
$GPT-3.5_{FLARE}$	it68.1	82.7	**98.3**	**85.4**	55.1	**65.5**	**82.4**	85.6	**49.8**
$GPT-3.5_{F-CoT}$	75.8	**83.0**	it95.3	81.7	it53.5	it51.5	73.5	it52.3	12.1
$GPT-3.5_{CoT}$	**79.8**	it82.4	98.2	it75.8	**59.4**	51.7	it69.9	**95.8**	4.3

Table 7.1: The following table shows the performance of each of the tested models given a technique for reasoning. Each **bold**, underlined, and ititalicised element highlights the best, second best and worst technique per specific model. The overall best method per dataset is highlighted in green.

problem space traversal similar to Prolog while circumventing the need for code tuning a generalist model.

7.2.4 Reasoning Faithfulness

An explanation is considered *faithful* if it explicitly and accurately describes the reasoning process of the model during inference (Gilpin et al., 2018; Jacovi and Goldberg, 2020). In the context of prompting techniques such as CoT, we are interested in the faithfulness of the intermediate reasoning chains towards the final output. Faithful intermediate reasoning chains should not just look *plausible* (Herman, 2017) but have exact reflections of the problem exploration and reasoning used to arrive at the final answer. Natural language reasoning chains prevalent in CoT and similar methods are shown to be unfaithful, either masking the reasoning biases (Turpin et al., 2023) of the model or outright ignoring the intermediate reasoning (Lanham et al., 2023b). In FLARE, we introduce a method to seamlessly measure the faithfulness of the final outcome w.r.t. completed search.

7.3 Methodology

7.3.1 LLM Simulated Search

FLARE comprises three modules for generating a *plan*, *code* and simulated *search* for answering a natural language query $\mathcal{Q} = \{T_1^{\mathcal{Q}} \dots T_{|\mathcal{Q}|}^{\mathcal{Q}}\}$, where each $T_i^{\mathcal{Q}}$ is a token in the query $\mathcal{Q}$.

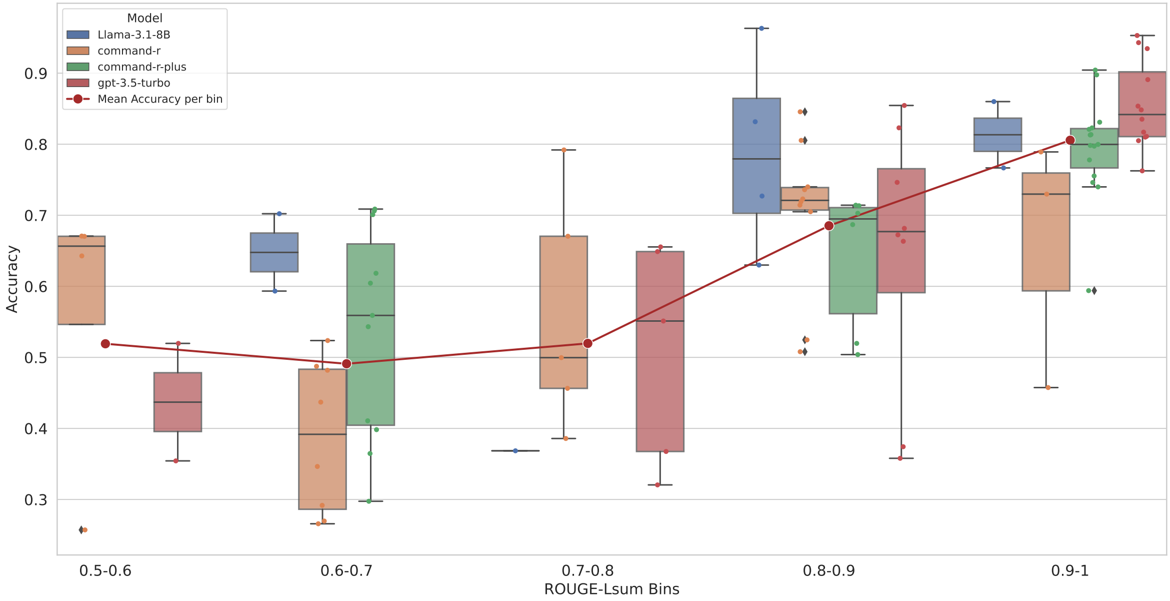


Figure 7.2: The trend of mean model accuracy w.r.t mean faithfulness for all the models.

7.3.2 Generating A Plan

For each query $\mathcal{Q}$, given an LLM $\mathcal{M}$, we initially use instructions $\mathcal{I}^{\mathcal{P}}$ to prompt it to generate a *plan* $\mathcal{P}$, which should be comprised of task explanation, analysis and a plan for further formalising the query. An example of this can be seen in the *plan* section in fig. 7.1. We use in-context few shot examples $\mathcal{E}_{\mathcal{P}}$ of such *plan* generations along with bfgreedy decoding for obtaining the final plan.

$$\mathcal{P}_i \sim argmaxp_{\mathcal{M}}(T_i^{\mathcal{P}} \mid T_{:i-1}^{\mathcal{P}}, \mathcal{E}_{\mathcal{P}}, \mathcal{Q}, \mathcal{I}^{\mathcal{P}}) \tag{7.1}$$

Where $\mathcal{P}_i$ and $T_i^{\mathcal{P}}$ is the i-th token in the generated *plan* $\mathcal{P}$ and $p_{\mathcal{M}}$ is the probability of the next token over the vocabulary obtained from model $\mathcal{M}$.

7.3.3 Generating Code

After generating the *plan*, we use instructions $\mathcal{I}^{\mathcal{C}}$ to prompt the LLM $\mathcal{M}$ to generate a Prolog code $\mathcal{C}$, an example of which can be seen in fig. 7.1. We append executable code generation samples $\mathcal{C}_{sample}$ to the previous in-context examples $\mathcal{E}_{\mathcal{P}}$ and obtain few-shot code generation demonstrations $\mathcal{E}_{\mathcal{C}} = [\mathcal{E}_{\mathcal{P}}; \mathcal{C}_{sample}]$

$$\mathcal{C}_i \sim \arg\max p_{\mathcal{M}}(T_i^{\mathcal{C}} \mid T_{:i-1}^{\mathcal{C}} \mathcal{E}_{\mathcal{C}}, \mathcal{Q}, \mathcal{I}^{\mathcal{P}}, \mathcal{P}, \mathcal{I}^{\mathcal{C}}) \tag{7.2}$$

$$\mathcal{F}_{code}, \mathcal{R}_{code}, \mathcal{G}_{code} = EXTRACT(\mathcal{C}_i)$$

Where $\mathcal{C}_i and T_i^{\mathcal{C}}$ is the i-th token in the generated *code* $\mathcal{C}$.

7.3.4 Benefits of Prolog

Prolog is a symbolic logic-programming engine (Bowen, 1979) designed for heuristic search over Horn Clauses (Chandra and Harel, 1985). As a declarative programming paradigm (Lloyd, 1994), the code is expressed as the logic of computation, expressed as a set of facts $\mathcal{F}$ and relations $\mathcal{R}$ defining the problem space, with the goal $\mathcal{G}$ being a first-order logic combination of them. Prolog employs depth-first search (DFS) (Bowen, 1979) for sub-goal decomposition and traversal of the problem space, satisfying $\mathcal{G}$ through a sequence of steps known as the *trace*. Each step either confirms/invalidates a sub-goal, expands the search tree, or retries failed sub-goals with new combinations. An example of such a search is shown in fig. 7.1. Prolog supports exhaustive search by exploring all paths that satisfy or fail the goal. This explicit segmentation of facts, relations, and search strategies simplifies query formalization. As a declarative language, Prolog enables segmentation using a simple regexp heuristic, referred to as EXTRACT in eq. (7.2) and eq. (7.3). Including exhaustive traces in-context allows an LLM to simulate sub-goal decomposition, backtracking, and intermediate goal invalidation, discussed further in the next paragraph.

7.3.5 Simulating Search

After generating the logic-programming *code*, we want to simulate program execution by generating a problem space traversal trace with our LLM $\mathcal{M}$. We use instructions $\mathcal{I}^{\mathcal{S}}$ and update our in-context samples by appending search traces $\mathcal{S}_{sample}$ constructed from Prolog execution of sample codes $\mathcal{C}_{sample}$, i.e. $\mathcal{E}_{\mathcal{S}} = [\mathcal{E}_{\mathcal{C}}; \mathcal{S}_{sample}]$.

$$\begin{aligned} \mathcal{S}_i &\sim \arg\max p_{\mathcal{M}}(T_i^{\mathcal{S}} \mid T_{:i-1}^{\mathcal{S}} \mathcal{E}_{\mathcal{C}}, \mathcal{Q}, \mathcal{I}^{\mathcal{P}}, \mathcal{P}, \mathcal{I}^{\mathcal{C}}, \mathcal{C}, \mathcal{I}^{\mathcal{S}}) \\ \mathcal{A}_{search}, \mathcal{F}_{search}, \mathcal{R}_{search} &= EXTRACT(\mathcal{S}_i) \end{aligned} \tag{7.3}$$

Where $T_i^{\mathcal{S}}$ is the i-th token in the generated *search* trace $\mathcal{S}$. During iterative problem space traversal, we can segment the facts $\mathcal{F}_{search}$, relations $\mathcal{R}_{search}$, completed and backtracked paths with their answers $\mathcal{A}_{search}$ used during the search simulation. To get the final answer we update in-context samples with their correct final answers $\mathcal{A}_{sample}$ from the executed search $\mathcal{S}_{sample}$, $\mathcal{E}_{\mathcal{A}} = [\mathcal{E}_{\mathcal{S}}; \mathcal{A}_{sample}]$ and use instructions $\mathcal{I}^{\mathcal{A}}$ to obtain the final answer from the model.

$$\mathcal{A}_{Final} \sim \arg\max p_{\mathcal{M}}(T_i^{\mathcal{A}} \mid T_{:i-1}^{\mathcal{A}} \mathcal{E}_{\mathcal{C}}, \mathcal{Q}, \mathcal{I}^{\mathcal{P}}, \mathcal{P}, \mathcal{I}^{\mathcal{C}}, \mathcal{C}, \mathcal{I}^{\mathcal{S}}, \mathcal{S}, \mathcal{I}^{\mathcal{A}}) \tag{7.4}$$

The prompts used for generating each part in FLARE can be seen in section 7.7 along with a complete example in table 7.8.

7.3.6 Detecting Reasoning Inconsistencies

For each query $\mathcal{Q}$ given the *code* $\mathcal{C}$ and the simulated *search* $\mathcal{S}$ along with the extracted facts $\mathcal{F}_{code}, \mathcal{F}_{search}$ and relations $\mathcal{R}_{code}, \mathcal{R}_{search}$ from each designated module, we aim to detect the inconsistencies during the reasoning process of the LLM. We use exact string matching between all these facts and relations in *code* and simulated *search*.

$$\forall i, \exists j \quad suchthat \quad \mathcal{F}^i_{code} = \mathcal{F}^j_{search} \quad and \quad \forall v, \exists q \quad \mathcal{R}^v_{code} = \mathcal{R}^q_{search} \tag{7.5}$$

$$\forall j, \exists i \quad suchthat \quad \mathcal{F}^i_{code} = \mathcal{F}^j_{search} \quad and \quad \forall q, \exists v \quad \mathcal{R}^v_{code} = \mathcal{R}^q_{search} \tag{7.6}$$

With this framework in mind, we define two reasoning failure modes. In the *first* failure mode, given that some fact or relation was used in the simulated *search* but did not exist in the generated *code*, i.e. $\exists j suchthat \mathcal{F}^j_{search} \notin \mathcal{F}_{code}$, we claim that the LLM has *hallucinated*. We postulate that the model either produced incomplete knowledge during formalisation to *code* or created a piece of non-existing information during the *search*. We do not consider facts that emerged during a direct inference step within the simulated search during our calculation. For example, if we are dealing with a mathematical query $4 \cdot (5 + 6) = ?$, the search would involve separately evaluating the expression $5 + 6 = 11$. In this case, 11 will not be treated as a hallucinated fact within the search but rather as an emergent fact obtained from direct inference. The *second* failure mode is the reciprocal case, where a fact or relation present in the *code* is not used during the *search*. We refer to this phenomenon as *sub-optimal reasoning* as it shows that the LLM could not explore the problem space completely or injected unsuitable knowledge during formalisation into *code*.

Dataset	ChatGPT (gpt-3.5-turbo)			
	Standard	CoT	Logic-LM	FLARE
PrOntoQA	47.40	67.80	61.00	**73.40**
LogicalDeduction	40.00	42.33	**65.67**	58.60
AR-LSAT	20.34	17.31	26.41	**27.39**

Table 7.2: Comparison of Direct Prompting, CoT, Logic-LM and FLARE.

7.3.7 Measuring Faithfulness

We propose a method to measure the faithfulness of the LLM reasoning process when using FLARE. As mentioned in section 7.3.1, for each query in a dataset $\mathcal{D} = [\mathcal{Q}_1, \ldots, \mathcal{Q}_{|\mathcal{D}|}]$, we generate a set of codes $\Phi = [\mathcal{C}_1, \ldots, \mathcal{C}_{|\Phi|}]$ and simulated problem space searches $\Psi = [\mathcal{S}_1, \ldots, \mathcal{S}_{|\Psi|}]$. We use the Prolog engine to execute all of the codes Φ and obtain a set of correctly written programs Φ' and exact search paths Ψ'. As we do not require explicit programmatic correctness during inference in FLARE for any code $\mathcal{C}_i$, some Prolog executions resulting in an error are filtered out in Ψ'. To assess model reasoning faithfulness towards code formalisations, we compare the search paths Φ' obtained from Prolog execution with their designated counterparts Φ'_{gen} generated by the LLM from the same code. We use ROUGE (Lin, 2004) to compute the matching score for each executed and simulated search path. In particular, we use ROUGE-Lsum, which uses the longest common subsequence (LCS) over each line to obtain the final score. This method fits our cause as a line in a Prolog search execution represents a single logic step within the traversal. This allows us to measure the similarity of the reasoning contents and structure in exact and simulated searches.

7.4 Experimental Setup

7.4.1 Datasets

To evaluate FLARE, we use a benchmark of 9 tasks covering Math Word Problems (MWP), multi-hop QA and relation inference, and 3 common logical reasoning datasets. For testing numeric and mathematical reasoning, we follow CoT (Wei et al., 2022b) by including GSM8K (Cobbe et al., 2021), SVAMP (Patel et al., 2021b), MultiArith (Roy and Roth, 2015), ASDiv (Miao et al., 2020) and AQuA (Ling et al., 2017). Among these, GSM8K, SVAMP, MultiArith and ASDiv cover elementary and middle school arithmetic word problems with a set of integers or decimals as the answer. AQuA is a multiple-choice numerical, symbolic reasoning dataset where each answer is a mathematical expression containing notations, values and expressions not defined in the query. We also test FLARE using three multi-hop QA datasets. We use StrategyQA (Geva et al., 2021), which is a boolean QA task that requires sub-goal decomposition and a multi-hop reasoning strategy to answer. The example *"Do all parts of the aloe vera plant taste good?"* used in fig. 7.1, is taken from StrategyQA. The multi-hop QA testing also includes Date and Sports Understanding, subsets of BIG-Bench (bench authors, 2023). The tasks involve inferring an exact date given some calculations in the relative time period and understanding if an artificially created sports statement is feasible. Furthermore, we assess FLARE on Relational Inference using CLUTRR (Sinha et al., 2019), which involves inferring the familial relation between two entities mentioned in a natural language description of the partial family graph. We evaluate FLARE on challenging logic inference datasets: ProntoQA (Saparov and He, 2023), AR-LSAT (Zhong et al., 2021), and LogicalDeductions from BigBench (et al., 2023), focusing on harder subsets proposed by (Pan et al., 2023). These datasets, covering deductive, analytical, and logical reasoning, allow us to assess FLARE's performance. Details, including descriptions and examples, are in table 7.9 of section 7.7.

Method	$CmDR_{plan-only}$	$CmDR_{FLARE}$	$CmDR+_{plan-only}$	$CmDR+_{FLARE}$	$GPT-3.5_{plan-only}$	$GPT-3.5_{FLARE}$
GSM8K	24.7	**52.4**	40.7	**71.4**	36.1	**68.1**
AQuA	35.0	**43.7**	55.1	**55.9**	54.3	**55.1**
StrategyQA	65.5	**67.0**	**75.7**	70.8	62.3	**65.5**

Table 7.3: The table shows the accuracy of an LLM with FLARE compared to prompting for a final answer directly after generating (plan-only) a plan $\mathcal{P}$.

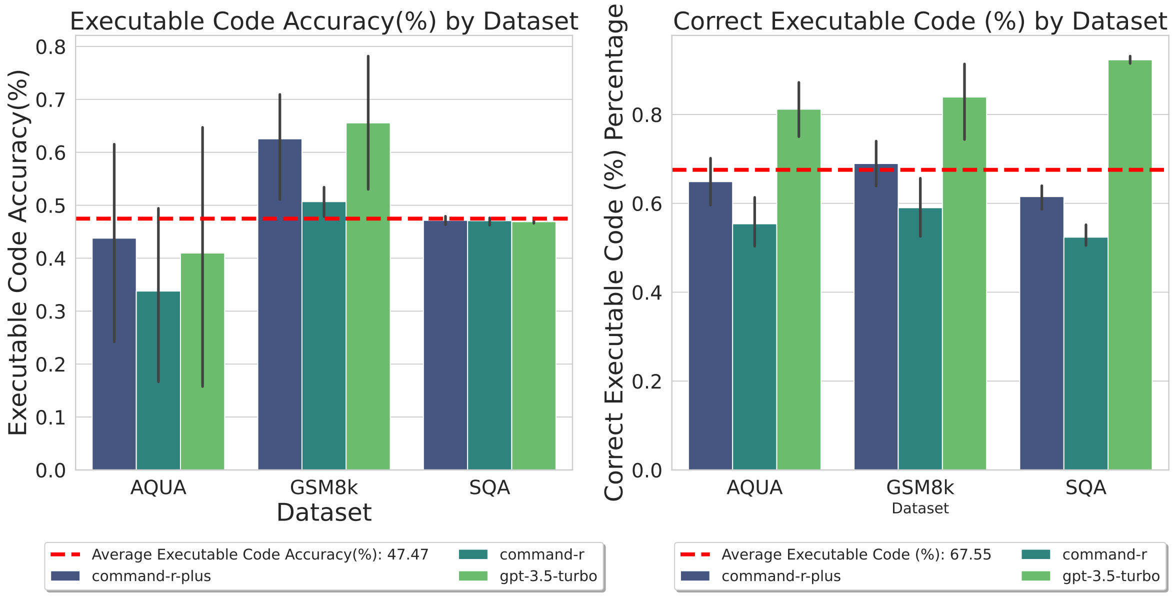


Figure 7.3: The figure shows the percentage of executable code per model (right) and the accuracy of the executable code when answering the queries (left).

7.4.2 Benchmarks

We compare FLARE with CoT (Wei et al., 2022b) as a prompting method that reasons using natural language chains and with F-CoT (Lyu et al., 2023) and Logic-LM (Pan et al., 2023) that formalise the query into a code and offload the reasoning to an external symbolic solver. We use Llama3.1 (8B) (Dubey et al., 2024), CmDR (30B) (Cohere, 2024), CmDR+ (100B) (Cohere, 2024) and GPT3.5 (Brown et al., 2020c) ($\geq 100B$ (Ye et al., 2023)). As the coding model OpenAI Codex (code-DaVinci-002) (Chen et al., 2021) used in F-CoT has been deprecated, we replace it with the new GPT3.5 as suggested by OpenAI and recalculate the results accordingly.

7.5 Results

7.5.1 Few-shot prompting

To evaluate FLARE, we use a set of models of varying sizes on diverse benchmarks, as defined in section 7.4. We compare the performance of each model while using FLARE, CoT and F-CoT prompting. The results for F-CoT and CoT on all the models are computed using the codebase of the original study (Lyu et al., 2023). We additionally compare Logic-LM and FLARE using the logic reasoning benchmarks proposed in (Pan et al., 2023).

7.5.2 LLMs for general reasoning

Our results, presented in table 7.1, show that using FLARE allows the LLMs to achieve state-of-the-art results on 7 out of 9 datasets, with an average 28% increase over CoT. We can see a clear trend that FLARE increases the performance compared to CoT and F-CoT for all the models of varying scales. We also see that LLMs that are not explicitly tuned for coding suffer massive degeneracies when using F-CoT. We postulate that they are unable to consistently produce executable programs that satisfy a predefined scheme in F-CoT, thus resulting in an error during execution. This further highlights the value of simulating program execution using an LLM instead of using external solvers. The results show that using FLARE yields more benefit on datasets that require longer chains of multi-hop and symbolic reasoning, like AQuA and StrategyQA. Our findings in table 7.2 show that FLAREachieves state-of-the-art results on 2 out of 3 logic inference benchmarks with 10% increase over CoT and 7% increase over Logic-LM.

7.5.3 LLMs for code generation

To understand the effect of FLARE on models tuned for coding, we use GPT3.5 (Brown et al., 2020a) as it was the OpenAI suggested succession model for Codex (Chen et al., 2021) which is used in F-CoT and possesses strong coding capabilities (Ye et al., 2023). The results in table 7.1 show that using FLARE is beneficial for models that are tuned for coding and boost the accuracy with a 16% increase over F-CoT and 9% over CoT. The reason is that many natural language queries with non-trivial formalisations are more suited to be tackled with more commonsense soft reasoning than direct code execution. This is evident in table 7.1 where FLARE and CoT are consistently better than F-CoT in StrategyQA, Sports and CLUTRR. The opposite case of numeric and algorithmic heavy reasoning tasks is also covered by FLARE as it maintains strong performance similar to F-CoT on MWP problems table 7.1. Consequently, FLARE allows combining algorithmic formalisation with simulated soft-reasoning, circumventing the pitfalls of using a deterministic external solver while still producing a query formalisation and problem space traversal.

7.5.4 Is simulating search useful?

To understand if simulating a search over the problem space is useful, we compare the performance of FLARE where we only generate the *plan* without the subsequent

Model	Avg. Number of Paths	Avg. #Hops per path	Avg. #Fails per path	Avg. Total Hops	Avg. Total Fails
			Incorrect Answers		
$Llama-3.1-8B_{FLARE}$	1.55	11.12	1.52	15.09	2.26
$CmDR_{FLARE}$	1.51	6.55	0.68	10.56	1.39
$CmDR+_{FLARE}$	0.92	7.52	1.13	8.57	1.32
GPT-3.5	0.68	5.22	0.71	5.32	0.74
			Correct Answers		
$Llama-3.1-8B_{FLARE}$	1.43	9.12	0.62	12.36	0.96
$CmDR_{FLARE}$	1.19	7.10	0.42	11.29	0.66
$CmDR+_{FLARE}$	0.97	7.19	0.42	8.22	0.61
$GPT-3.5_{FLARE}$	0.82	5.65	0.26	5.69	0.27

Table 7.4: The table depicts the difference in the average explored paths, hops, and fails during the reasoning process, which leads to incorrect or correct answers. The purple colour illustrates that incorrect reasoning paths have fewer explorations that led to Failed search paths.

code or *search* components. We refer to this framework setup as *plan-only*, which can be seen in fig. 7.1 if we were to use only the *plan* for answer generation. We completed this ablation using CmDR, CmDR+, and GPT-3.5, and we used GSM8K, AQuA, and StrategyQA as our baselines. The results in table 7.3 confirm that all of the models suffer massive performance degradation from $61.1 \rightarrow 49.9$ when omitting the *code* and the *search* components of FLARE. We hypothesise that this is caused by insufficient problem space exploration when using the *plan-only* setting. Furthermore, we have already seen in table 7.1 that in methods, like F-CoT, that do not use simulated problem space exploration for soft-reasoning and only rely on *plan* and *code*, the performance also deteriorates even resulting in a complete breakdown of reasoning over the designated datasets. This can be viewed as a constrained version of FLARE with *code-only* execution. Consequently, our results show that simulating problem space traversal is highly beneficial as it avoids the pitfalls posed by *plan-only* and *code-only* modes by exploring the problem space more rigorously and soft-reasoning during that traversal instead of using external solvers.

7.5.5 Faithful Reasoning Improves Performance

As described in section 7.3, using FLARE allows us to measure the faithfulness of the LLM reasoning process by comparing the simulated problem space traversals Φ'_{gen} with actual traces Φ' produced from a symbolic Prolog solver. To do this, we initially compute the percentage of syntactically correct executable code each LLM produces. We can see from the right part of fig. 7.3 that all of the models are capable of producing correct executable Prolog code in 67% of cases on average and $\geq 50\%$ of cases at the very least. This shows that the simulated searches Φ'_{gen} can be considered a representative sample that will be further used to accurately measure the faithfulness of the simulated search w.r.t. the generated code. After measuring the reasoning faithfulness for each model, we want to understand what impact it

Model	Unique Explorations (%) in Search	Relation overlap (%)	Unused Code relations (%)
	Correct Answers		
$Llama-3.1-8B_{FLARE}$	74.14	43.65	5.73
$CmDR_{FLARE}$	59.06	35.96	4.02
$CmDR+_{FLARE}$	64.30	34.47	4.54
$GPT-3.5_{FLARE}$	64.46	37.55	1.90
	Incorrect Answers		
$Llama-3.1-8B_{FLARE}$	54.69	35.04	9.28
$CmDR_{FLARE}$	54.50	32.76	6.23
$CmDR+_{FLARE}$	44.12	24.98	8.22
$GPT-3.5_{FLARE}$	36.02	24.44	6.94

Table 7.5: The table shows how the percentage of unique emergent inferences in search, overlapping relations between code and search, and unused relations in code impact answer correctness.

Model	Avg. hops per Paths	Hallucination (%)	Unutilised knowledge (%)
Llama-3.1-8B	9.4	63.3	62.9
CmDR	6.7	54.7	56.9
CmDR+	7.2	54.3	56.3
GPT-3.5	5.5	49.3	52.1

Table 7.6: The table shows the changes in simulated search statistics when using FLARE w.r.t model scale from 8B to 100B+. Hallucinations refer to facts and predicates only used in trace, while unutilised knowledge relates to the facts and relations only seen in the code.

has on the performance of the LLM. In fig. 7.2, we segment the models w.r.t. their ROUGE-Lsum scores. The results show that model performance is strongly positively correlated with reasoning faithfulness. However, we also observe in the left part of fig. 7.3 that executing semantically precise code results in an accurate answer only in 47% of cases on average. Indeed, having a simulated search trace with a ROUGE-Lsum faithfulness score of 1, would be equivalent to simply executing the program as proposed in F-CoT. Yet we have priorly shown that F-CoT struggles with reasoning tasks that are hard to formalise and require multi-hop commonsense and soft reasoning. These two discoveries show that optimal LLM reasoning, conditioned on a search in the problem space, should be increasingly faithful toward the facts, relations and the search strategy defined within the code while simultaneously maintaining the capability for soft-reasoning along more abstractly defined concepts. Our results show that FLARE allows LLMs to maintain a similar reasoning capacity.

7.5.6 What is important during the search?

We expand the analysis of the simulated search traces to detect the reasons which can lead to optimal reasoning within an LLM. For this purpose, we calculate several statistics, like the average number of explored paths, average and total hops and failures per path, for each model during the simulated traversal. The failure in a path is an invalidation of a solution for a sub-goal explored during the search, which is used for backtracking, as explained in section 7.3. Calculating these statistics is simple as the *search* component of FLARE, seen in fig. 7.1, is a structured simulation of a Prolog trace, where each line contains a hop of reasoning inference. We split these statistics for the reasoning paths that lead to correct or incorrect outcomes. Our results in table 7.4 show that LLM performance and reasoning optimally are not directly connected to the amount of explored paths or multi-hop inferences per path. We also see that traces that lead to incorrect answers have a higher number of failures per path and in total. We explain this phenomenon with the hypothesis that LLMs with traces that were optimal for reasoning and led to correct answers could skip exploring degenerate solutions due to strong commonsense reasoning capabilities. Further analyses focus on identifying inconsistencies and failure modes (section 7.3.6). By comparing relations in code with those in search traces, we measure emergent hallucinations and unused relations, highlighting areas of sub-optimal reasoning. Additionally, we assess the uniqueness of emergent facts per inference hop, which indicates the extent of problem-space exploration (table 7.5). The results in table 7.5 show consistently over each model that, on average, traces that lead to correct answers had a higher percentage of unique emergent facts (UEF) and overlap in the relations (OR) used between the code and search, while the portion of underutilized relations (UR) was lower. This means that optimal reasoning with an LLM requires a great degree of problem-space exploration with fewer relation hallucinations during the search and more relation utilization from the defined code. This aligns with our prior discoveries, which show a strong correlation between simulated search faithfulness towards the formalised code and model performance. Our framework FLARE has these reasoning patterns ingrained within its inference pipeline.

7.5.7 The effect of scale

We want to assess the impact of the number of parameters in the model on the overall performance and faithfulness. The results in fig. 7.4 show no precise relation between model scale, performance and faithfulness. However, scaled models from the same family, i.e. CmDR (30B) and CmDR+ (100B), show improvements in reasoning faithfulness and model performance. We can also see in table 7.6 that as the model size increases, the average number of hops and the portion of hallucinations and unutilised knowledge decreases. This further confirms our prior assumptions that models with strong commonsense soft-reasoning capabilities can skip steps during

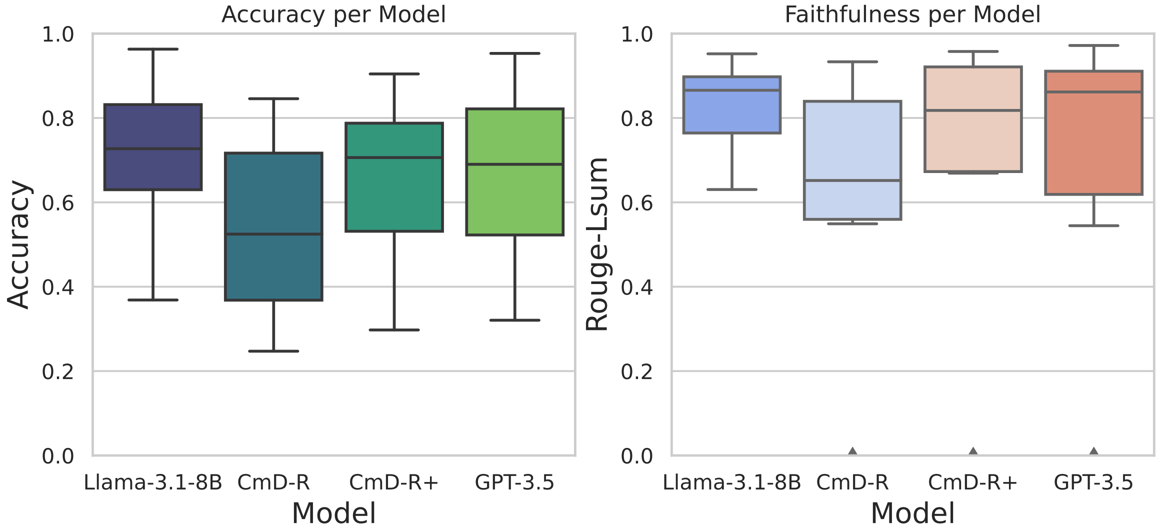


Figure 7.4: The effect of the model parameter scale from 8B to 100B+ on model accuracy (left) and faithfulness (right).

the search while maintaining the knowledge and structure of the traversal strategy outlined in the code.

7.6 Conclusion

This work introduces FLARE, a novel approach for logic-aided interpretable formalisation and reasoning with simulated search over the problem space. We show that models of varying scales obtain state-of-the-art results compared to prompting paradigms like CoT and F-CoT. We further pinpoint that using FLARE allows us to perform soft-reasoning with simulated search, making it flexible for diverse reasoning benchmarks. We introduce a method to measure model reasoning faithfulness w.r.t. the problem formalization ingrained within FLARE. Our results show that model performance is positively correlated with the faithfulness of the reasoning process. The systematic studies of the method show the benefits of using simulated search compared to natural language reasoning and external symbolic solvers. We further show that using FLARE allows us to interpretably and rigorously detect hallucinations and sub-optimal and inconsistent reasoning patterns.

Reproducibility Report

To reproduce the results of our study, we provide the complete codebase, processing pipelines and prompts for each dataset. The only model hyper-parameter we explicitly fix is the temperature for greedy decoding. We also make the inference of all of the models using FLARE, F-CoT and CoT across all of the datasets publicly available for further experimentation and exploration.

7.7 Appendix

7.7.1 LLM Prompts

We define straight-forward prompts for generating *plan*, *code* and *search* simulation in FLARE, which can be observed in section 7.7.2.

7.7.2 Dataset Statistics

The datasets used in this study encompass a variety of domains, specifically targeting the performance of the models in interpreting Math Word Problems, multi-hop question answering, and relational inference. Table 7.9 provides a detailed breakdown of each dataset, including the number of few-shot in-context samples (shots), the number of test samples, and representative examples from each dataset. The datasets provide a comprehensive basis for evaluating the models' abilities to handle complex tasks across different domains, facilitating an in-depth analysis of model performance under few-shot conditions.

7.7.3 FLARE Pseudo-code

Below, we present the pseudo-code for the execution of the *plan*, *code*, and *search* procedures in FLARE. The pseudo-code describes the modular pipeline in FLARE for tackling natural language queries with faithful simulated search.

- bfPlan Generation: This stage creates a structured natural language outline of the reasoning process, breaking down the query into logical steps and analysis. The plan serves as the foundation for formalization into a logic-based representation.

- bfCode Generation: Based on the generated plan, a logic programming code (e.g., in Prolog) is synthesized. This code formalizes the query into a set of facts, relations, and goals, which collectively define the problem space for reasoning.

- bfSearch Simulation: The generated code is utilized to simulate a search trace over the problem space. This includes iterative reasoning, backtracking when goals are unmet, and extracting emergent facts or relations during the process.

Each of these stages is implemented as a modular component. The generation from each of the stages feeds into the next, allowing seamless integration and incremental improvement in reasoning accuracy.

Task	Prompt	Description
Plan Generation	Generate an explanation and analysis, and plan to generate a prompt for writing a swi-prolog code for the last task. The 3 sections should be exactly outlined. Your plan should show enough intermediate reasoning steps towards the answer. Construct the plan as much as you can and describe the logic specifically. When constructing the plan for the code prompt, actively use swi prolog search capabilities.	Detailed instructions for generating an outline and plan, with an emphasis on reasoning steps and using Prolog's search capabilities.
Code Generation	Write a Prolog code to solve using the plan. If there are unknown or stochastic atoms or predicates, fill in the values for them as a logical assumption and add a comment in the same line Assumed atom/predicate". Do not use write and read commands within the code. The code should be very detailed and utilize swi prolog capabilities to the fullest. To run the program, at the end create a predicate named "query" that returns the correct numerical answer. The last line of the program should be the commented-out driver predicate "query". Write only the code.	Instructions for generating a Prolog code based on the plan with assumptions for unknown atoms. Emphasizes code details and a final "query" predicate.
Simulated Search	Ignoring the read commands, explicitly write out the search paths that are explored by the code: #### Here are the paths [Starting Search Simulation]: #### [Path 1]:	A task to simulate and display the search paths that the Prolog code would follow during execution.
Final Answer	Given the plan, the code and the explored search paths answer the question above. Answer with the correct numerical answer. ##### Here is the answer:	Final prompt asking for the correct numerical answer based on the previous steps.

Table 7.7: Table of Prompts for Plan, Code, Simulated Search, and Final Answer generation for GSM8K (Cobbe et al., 2021).

Section	Content
Question	"Can Reiki be stored in a bottle?"
Explanation	Reiki is a form of alternative therapy that involves the channeling of energy to promote healing and relaxation. It is often administered ...
Analysis	To address the question of whether Reiki can be stored in a bottle, it is important to consider...
Plan	• Understand Reiki Energy: ... • Explore Energy Storage: ... • Consider Symbolism and Intention: ... • Evaluate Practicality: ... • Reflect on Tradition: ...
Code	reiki_energy(spiritual_energy). reiki_energy(channeling_through_touch). store_energy_in_bottle(feasible) :- reiki_energy(spiritual_energy), reiki_energy(channeling_through_touch). not_common_practice(store_reiki_in_bottle). query :-store_energy_in_bottle(feasible), not_common_practice(store_reiki_in_bottle).
Search Paths	Path 1: – 0: Start of execution: Beginning Search – 1: Call: query – 2: Call: store_energy_in_bottle(feasible) – 3: Call: reiki_energy(spiritual_energy) – 4: Call: reiki_energy(channeling_through_touch) – 5: Call: not_common_practice(store_reiki_in_bottle) – 6: Call: feasible=no – 7: Fail: feasible=no | ’Result’: ’Search Failed’ – 8: Fail: query | ’Result’: ’Search Failed’
Answer	"No, Reiki energy cannot be stored in a bottle based on the logical evaluation of its abstract, non-physical nature and traditional practices of Reiki."

Domain	Dataset	Shots	Test Samples	Example
Math Word Problems	GSM8K	8	1,319	Q: A robe takes 2 bolts of blue fiber and half that much white fiber. How many bolts in total does it take? A: 3
	SVAMP	8	1,000	Q: Dan had $3 left with him after he bought a candy bar. If he had $4 at the start, how much did the candy bar cost?A: 1
	MultiArith	8	600	Q: A pet store had 13 siamese cats and 5 house cats. During a sale they sold 10 cats. How many cats do they have left? A: 8
	ASDiv	8	2,096	Q: Adam has five more apples than Jackie. Jackie has nine apples. How many apples does Adam have? A: 14
	AQuA	8	254	Q: A man walks at 5 kmph for 6 hrs and at 4 kmph for 12 hrs. His average speed is Answer option: A)4 1/3 km/h, B)7 2/3 km/h, C)9 ½ km/h, D)8 km/h, E)81 km/h A: A
Multi-hop QA	StrategyQA	6	2,290	Q: Did Aristotle use a laptop? A: False
	Date Understanding	10	359	Q: Yesterday was April 30, 2021. What is the date tomorrow in MM/DD/YYYY? A: "05/02/2021"
	Sports Understanding	10	977	Q: Is the following sentence plausible? Lionel Messi was called for icing? A: False
Relational Inference	CLUTRR	8	1,042	Q: [Carlos] is [Clarence]'s brother. [Carlos] and his sister, [Annie], went shopping. asked her mom [Valerie] if she wanted anything, but [Valerie] said no. How is [Valerie] related to [Clarence]? A: "mother"

Table 7.9: The statistics and examples of the datasets used in benchmarking. Shots refers to the number of few-shot in-context samples used during benchmarking.

Table 7.8: Complete example of FLARE

References

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

Priyanka Agrawal, Chris Alberti, Fantine Huot, Joshua Maynez, Ji Ma, Sebastian Ruder, Kuzman Ganchev, Dipanjan Das, and Mirella Lapata. 2023. Qameleon: Multilingual qa with only 5 examples. *Transactions of the Association for Computational Linguistics*, 11:1754–1771.

Ehud Aharoni, Anatoly Polnarov, Tamar Lavee, Daniel Hershcovich, Ran Levy, Ruty Rinott, Dan Gutfreund, and Noam Slonim. 2014. A Benchmark Dataset for Automatic Detection of Claims and Evidence in the Context of Controversial Topics. In *Proceedings of the First Workshop on Argumentation Mining*, pages 64–68, Baltimore, Maryland. Association for Computational Linguistics.

Kabir Ahuja, Harshita Diddee, Rishav Hada, Millicent Ochieng, Krithika Ramesh, Prachi Jain, Akshay Nambi, Tanuja Ganu, Sameer Segal, Maxamed Axmed, et al. 2023. Mega: Multilingual evaluation of generative ai. *arXiv preprint arXiv:2303.12528*.

Naveed Akhtar, Ajmal Mian, Navid Kardan, and Mubarak Shah. 2021. Advances in adversarial attacks and defenses in computer vision: A survey. *IEEE Access*, 9:155161–155196.

Abeer Aldayel and Walid Magdy. 2019. Your Stance is Exposed! Analysing Possible Factors for Stance Detection on Social Media. *Proc. ACM Hum.-Comput. Interact.*, 3(CSCW).

Abeer AlDayel and Walid Magdy. 2021. Stance detection on social media: State of the art and trends. *Inf. Process. Manag.*, 58(4):102597.

Alaaeldin Ali, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. 2021. Xcit: Cross-covariance image transformers. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 20014–20027.

Emily Allaway and Kathleen McKeown. 2020. Zero-Shot Stance Detection: A Dataset and Model using Generalized Topic Representations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8913–8931, Online. Association for Computational Linguistics.

James Allen. 1995. *Natural language understanding*. Benjamin-Cummings Publishing Co., Inc.

Saadullah Amin, Stalin Varanasi, Katherine Ann Dunfield, and Günter Neumann. 2020. Lowfer: Low-rank bilinear pooling for link prediction. In *International Conference on Machine Learning*, pages 257–268. PMLR.

Aixiu An, Peng Qian, Ethan Wilcox, and Roger Levy. 2019. Representation of constituents in neural language models: Coordination phrase as a case study. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2888–2899, Hong Kong, China. Association for Computational Linguistics.

John Robert Anderson. 1983. *Machine Learning: An Artificial Intelligence Approach, volume II*, volume 2. Morgan Kaufmann.

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. 2020. Square attack: A query-efficient black-box adversarial attack via random search. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXIII*, volume 12368 of *Lecture Notes in Computer Science*, pages 484–501. Springer.

Peter Adam Angeles. 1981. Dictionary of philosophy.

Richard B Angell. 1989. Deducibility, entailment and analytic containment. *Directions in relevant logic*, pages 119–143.

Dimo Angelov. 2020. Top2vec: Distributed representations of topics. *ArXiv preprint*, abs/2008.09470.

Erik Arakelyan, Arnav Arora, and Isabelle Augenstein. 2023a. Topic-guided sampling for data-efficient multi-domain stance detection. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 13448–13464. Association for Computational Linguistics.

Erik Arakelyan, Daniel Daza, Pasquale Minervini, and Michael Cochez. 2021. Complex query answering with neural link predictors. In *ICLR*. OpenReview.net.

Erik Arakelyan, Karen Hambardzumyan, Davit Papikyan, Pasquale Minervini, Aram H. Markosyan, Albert Gordo, and Isabelle Augenstein. 2024a. With great backbones comes great adversarial transferability. *CoRR*. To appear on arXiv.

Erik Arakelyan, Zhaoqi Liu, and Isabelle Augenstein. 2024b. Semantic sensitivities and inconsistent predictions: Measuring the fragility of NLI models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2024 - Volume 1: Long Papers, St. Julian's, Malta, March 17-22, 2024*, pages 432–444. Association for Computational Linguistics.

Erik Arakelyan, Pasquale Minervini, Daniel Daza, Michael Cochez, and Isabelle Augenstein. 2023b. Adapting neural link predictors for data-efficient complex query answering. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Erik Arakelyan, Pasquale Minervini, Pat Verga, Patrick S. H. Lewis, and Isabelle Augenstein. 2024c. FLARE: faithful logic-aided reasoning and exploration. *CoRR*, abs/2410.11900.

Alejandro Barredo Arrieta, Natalia Díaz Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, et al. 2020. Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion*, 58:82–115.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2020. Translation artifacts in cross-lingual transfer learning. *arXiv preprint arXiv:2004.04721*.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the association for computational linguistics*, 7:597–610.

Viraat Aryabumi, Yixuan Su, Raymond Ma, Adrien Morisot, Ivan Zhang, Acyr Locatelli, Marzieh Fadaee, Ahmet Üstün, and Sara Hooker. 2024. To code, or not to code? exploring impact of code in pre-training. *CoRR*, abs/2408.10914.

Akari Asai, Akiko Eriguchi, Kazuma Hashimoto, and Yoshimasa Tsuruoka. 2018. Multilingual extractive reading comprehension by runtime machine translation. *arXiv preprint arXiv:1809.03275*.

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020. A diagnostic study of explainability techniques for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 3256–3274. Association for Computational Linguistics.

Sara Atito, Muhammad Awais, and Josef Kittler. 2021. Sit: Self-supervised vision transformer. *arXiv preprint arXiv:2104.03602*.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. 2007. DBpedia: A nucleus for a web of open data. In *ISWC/ASWC*, volume 4825 of *Lecture Notes in Computer Science*, pages 722–735. Springer.

Isabelle Augenstein, Tim Rocktäschel, Andreas Vlachos, and Kalina Bontcheva. 2016. Stance detection with bidirectional conditional encoding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 876–885, Austin, Texas. Association for Computational Linguistics.

Jacob Austin, Augustus Odena, Maxwell I. Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie J. Cai, Michael Terry, Quoc V. Le, et al. 2021. Program synthesis with large language models. *CoRR*, abs/2108.07732.

Hayastan Avetisyan and David Broneske. 2023. Large language models and low-resource languages: An examination of armenian nlp. *Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023 (Findings)*, pages 199–210.

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*.

Vineeth Balasubramanian, Shen-Shyang Ho, and Vladimir Vovk. 2014. *Conformal prediction for reliable machine learning: theory, adaptations and applications*. Newnes.

Ivana Balažević, Carl Allen, and Timothy M Hospedales. 2019. Tucker: Tensor factorization for knowledge graph completion. *arXiv preprint arXiv:1901.09590*.

Roy Bar-Haim, Indrajit Bhattacharya, Francesco Dinuzzo, Amrita Saha, and Noam Slonim. 2017. Stance classification of context-dependent claims. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 251–261, Valencia, Spain. Association for Computational Linguistics.

Bjorn Barz and Joachim Denzler. 2020. Deep learning on small datasets without pre-training using cosine loss. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1371–1380.

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

Talia Ben-Zeev. 1998. Rational errors and the mathematical mind. *Review of General Psychology*, 2(4):366–383.

Talia Ben-Zeev. 2012. When erroneous mathematical thinking is just as “correct”: The oxymoron of rational errors. In *The nature of mathematical thinking*, pages 55–79. Routledge.

BIG bench authors. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *FAccT ’21: 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event / Toronto, Canada, March 3-10, 2021*, pages 610–623. ACM.

Leonard Bereska and Efstratios Gavves. 2024. Mechanistic interpretability for AI safety - A review. *CoRR*, abs/2404.14082.

Vance W Berger and YanYan Zhou. 2014. Kolmogorov–smirnov test: Overview. *Wiley statsref: Statistics reference online*.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 17682–17690. AAAI Press.

Lucas Beyer, Olivier J Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. 2020. Are we done with imagenet? *arXiv preprint arXiv:2006.07159*.

Alina Beygelzimer, Sanjoy Dasgupta, and John Langford. 2009. Importance weighted active learning. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, pages 49–56. ACM.

Arjun Nitin Bhagoji, Warren He, Bo Li, and Dawn Song. 2018. Practical black-box attacks on deep neural networks using efficient query mechanisms. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XII*, volume 11216 of *Lecture Notes in Computer Science*, pages 158–174. Springer.

Siddhant Bhambri, Sumanyu Muku, Avinash Tulasi, and Arun Balaji Buduru. 2019. A survey of black-box adversarial attacks on computer vision models. *arXiv preprint arXiv:1912.01667*.

Prajjwal Bhargava, Aleksandr Drozd, and Anna Rogers. 2021. Generalization in NLI: ways (not) to go beyond simple heuristics. *CoRR*, abs/2110.01518.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2001. Latent Dirichlet Allocation. In *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]*, pages 601–608. MIT Press.

Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. 1989. Learnability and the vapnik-chervonenkis dimension. *J. ACM*, 36(4):929–965.

Oliver Bodenreider, Ronald Cornet, and Daniel J Vreeman. 2018. Recent developments in clinical terminologies - snomed ct, loinc, and rxnorm. *Yearbook of medical informatics*, 27:129–139.

Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD Conference*, pages 1247–1250. ACM.

Filip Boltužić and Jan Šnajder. 2014. Back up your Stance: Recognizing Arguments in Online Discussions. In *Proceedings of the First Workshop on Argumentation Mining*, pages 49–58, Baltimore, Maryland. Association for Computational Linguistics.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *NIPS*, pages 2787–2795.

Kenneth A. Bowen. 1979. Prolog. In *Proceedings of the 1979 Annual Conference, Detroit, Michigan, USA, October 29-31, 1979*, pages 14–23. ACM.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Ivan Bratko. 1997. Machine learning: Between accuracy and interpretability. In *Learning, networks and statistics*, pages 163–177. Springer.

Gianni Brauwers and Flavius Frasincar. 2023. A general survey on attention mechanisms in deep learning. *IEEE Trans. Knowl. Data Eng.*, 35(4):3279–3298.

Wieland Brendel, Jonas Rauber, and Matthias Bethge. 2018. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020a. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020b. Language models are few-shot learners.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020c. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Antonio Bruno, Davide Moroni, and Massimo Martinelli. 2022. Efficient adaptive ensembling for image classification. *arXiv preprint arXiv:2206.07394*.

Felix Buchert, Nassir Navab, and Seong Tae Kim. 2022. Exploiting Diversity of Unlabeled Data for Label-Efficient Semi-Supervised Active Learning. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 2063–2069. IEEE.

Jaime G Carbonell, Ryszard S Michalski, and Tom M Mitchell. 1983. An overview of machine learning. *Machine learning*, pages 3–23.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. 2022. Membership inference attacks from first principles. In *43rd IEEE Symposium on Security and Privacy, SP 2022, San Francisco, CA, USA, May 22-26, 2022*, pages 1897–1914. IEEE.

Nicholas Carlini and David A. Wagner. 2017. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, AISec@CCS 2017, Dallas, TX, USA, November 3, 2017*, pages 3–14. ACM.

Rudolf Carnap. 1959. *Introduction to semantics and formalization of logic*. Harvard University Press.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. 2018. Deep clustering for unsupervised learning of visual features. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIV*, volume 11218 of *Lecture Notes in Computer Science*, pages 139–156. Springer.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. 2020. Unsupervised learning of visual features by contrasting cluster assignments. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. 2021. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 9630–9640. IEEE.

Casimiro Pio Carrino, Marta R Costa-Jussà, and José AR Fonollosa. 2019. Automatic spanish translation of the squad dataset for multilingual question answering. *arXiv preprint arXiv:1912.05200*.

Hyungjoo Chae, Yeonghyeon Kim, Seungone Kim, Kai Tzu-iunn Ong, Beong-woo Kwak, Moohyeon Kim, Seonghwan Kim, Taeyoon Kwon, Jiwan Chung, Youngjae Yu, et al. 2024. Language models as compilers: Simulating pseudocode execution improves algorithmic reasoning in language models. *CoRR*, abs/2404.02575.

Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. 2018. Adversarial attacks and defences: A survey. *arXiv preprint arXiv:1810.00069*.

Ashok K. Chandra and David Harel. 1985. Horn clauses queries and generalizations. *J. Log. Program.*, 2(1):1–15.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017a. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *CoRR*, abs/2107.03374.

Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. 2017b. ZOO: zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, AISec@CCS 2017, Dallas, TX, USA, November 3, 2017*, pages 15–26. ACM.

Sihao Chen, Daniel Khashabi, Wenpeng Yin, Chris Callison-Burch, and Dan Roth. 2019. Seeing things from a different angle: Discovering diverse perspectives about claims. 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 542–557, Minneapolis, Minnesota. Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020a. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. 2020b. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255.

Weilun Chen, Zhaoxiang Zhang, Xiaolin Hu, and Baoyuan Wu. 2020c. Boosting decision-based black-box adversarial attacks with random sign flip. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XV*, volume 12360 of *Lecture Notes in Computer Science*, pages 276–293. Springer.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2023. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Trans. Mach. Learn. Res.*, 2023.

Xiaojun Chen, Shengbin Jia, and Yang Xiang. 2020d. A review: Knowledge reasoning over knowledge graph. *Expert Syst. Appl.*, 141.

Xuelu Chen, Ziniu Hu, and Yizhou Sun. 2022. Fuzzy logic based logical query answering on knowledge graphs. In *AAAI*, pages 3939–3948. AAAI Press.

Nurendra Choudhary, Nikhil Rao, Sumeet Katariya, Karthik Subbian, and Chandan K Reddy. 2021. Self-supervised hyperboloid representations from logical queries over knowledge graphs. In *Proceedings of the Web Conference 2021*, pages 1373–1384.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models.

Aaron Cicourel. 1991. Semantics, pragmatics, and situated meaning. *Pragmatics at Issue*, 1:37–66.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Michael Clark. 1967. The general notion of entailment. *The Philosophical Quarterly (1950-)*, 17(68):231–245.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.

Michael Cochez, Dimitrios Alivanistos, Erik Arakelyan, Max Berrendorf, Daniel Daza, Mikhail Galkin, Pasquale Minervini, Mathias Niepert, and Hongyu Ren. 2023. Approximate answering of graph queries. In Pascal Hitzler, Md. Kamruzzaman Sarker, and Aaron Eberhart, editors, *Compendium of Neurosymbolic Artificial Intelligence*, volume 369 of *Frontiers in Artificial Intelligence and Applications*, pages 373–386. IOS Press.

Cohere. 2024. Command r: Retrieval-augmented generation at production scale. `https://txt.cohere.com/command-r`.

Cleo Condoravdi, Dick Crouch, Valeria De Paiva, Reinhard Stolle, and Daniel Bobrow. 2003. Entailment, intensionality and text understanding. In *Proceedings of the HLT-NAACL 2003 workshop on Text meaning*, pages 38–45.

Costanza Conforti, Jakob Berndt, Mohammad Taher Pilehvar, Chryssi Giannitsarou, Flavio Toxvaerd, and Nigel Collier. 2020. Will-they-won't-they: A very large dataset for stance detection on Twitter. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1715–1724, Online. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*, pages 177–190. Springer.

Bo Dai and Dahua Lin. 2017. Contrastive Learning for Image Captioning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 898–907.

Nilesh N. Dalvi and Dan Suciu. 2004. Efficient query evaluation on probabilistic databases. In *VLDB*, pages 864–875. Morgan Kaufmann.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. 2017. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. *arXiv preprint arXiv:1711.05851*.

Rajshekhar Das, Yu-Xiong Wang, and José MF Moura. 2021. On the importance of distractors for few-shot classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9030–9040.

Ishita Dasgupta, Demi Guo, Andreas Stuhlmüller, Samuel J. Gershman, and Noah D. Goodman. 2018. Evaluating compositionality in sentence embeddings.

Ishita Dasgupta, Andrew K Lampinen, Stephanie CY Chan, Antonia Creswell, Dharshan Kumaran, James L McClelland, and Felix Hill. 2022. Language models show human-like content effects on reasoning. *arXiv preprint arXiv:2207.07051*.

Brian A. Davey and Hilary A. Priestley. 2002. *Introduction to Lattices and Order, Second Edition*. Cambridge University Press.

Daniel Daza and Michael Cochez. 2020. Message passing query embedding. In *ICML Workshop - Graph Representation Learning and Beyond*.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Leon Derczynski, Kalina Bontcheva, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Arkaitz Zubiaga. 2017. SemEval-2017 task 8: RumourEval: Determining rumour veracity and support for rumours. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 69–76, Vancouver, Canada. Association for Computational Linguistics.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *AAAI*, pages 1811–1818. AAAI Press.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. 2018. Boosting adversarial attacks with momentum. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 9185–9193. Computer Vision Foundation / IEEE Computer Society.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Omar Elharrouss, Younes Akbari, Noor Almaadeed, and Somaya Al-Maadeed. 2022. Backbones-review: Feature extraction networks for deep learning and deep reinforcement learning approaches. *arXiv preprint arXiv:2206.08016*.

Aarohi Srivastava et al. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Trans. Mach. Learn. Res.*, 2023.

William Ferreira and Andreas Vlachos. 2016. Emergent: a novel data-set for stance classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1163–1168, San Diego, California. Association for Computational Linguistics.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. 2020. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*.

Bent Fuglede and Flemming Topsoe. 2004. Jensen-shannon divergence and hilbert space embedding. In *International symposium onInformation theory, 2004. ISIT 2004. Proceedings.*, page 31. IEEE.

Mikhail Galkin, Xinyu Yuan, Hesham Mostafa, Jian Tang, and Zhaocheng Zhu. 2024. Towards foundation models for knowledge graph reasoning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Kanishk Gandhi, Denise Lee, Gabriel Grand, Muxin Liu, Winson Cheng, Archit Sharma, and Noah D. Goodman. 2024. Stream of search (sos): Learning to search in language. *CoRR*, abs/2404.03683.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. PAL: program-aided language models. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 10764–10799. PMLR.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ismael Garrido-Muñoz, Arturo Montejo-Ráez, Fernando Martínez-Santiago, and L Alfonso Ureña-López. 2021. A survey on bias in deep nlp. *Applied Sciences*, 11(7):3184.

Sebastian Gehrmann, Tosin P. Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Aremu Anuoluwapo, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu, Dipanjan Das, Kaustubh D. Dhole, et al. 2021. The GEM benchmark: Natural language generation, its evaluation and metrics. *CoRR*, abs/2102.01672.

Atticus Geiger, Kyle Richardson, and Christopher Potts. 2020. Neural natural language inference models partially embed theories of lexical entailment and negation. *arXiv preprint arXiv:2004.14623*.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? A question answering benchmark with implicit reasoning strategies. *Trans. Assoc. Comput. Linguistics*, 9:346–361.

Gayane Ghazaryan, Erik Arakelyan, Pasquale Minervini, and Isabelle Augenstein. 2024. Syndarin: Synthesising datasets for automated reasoning in low-resource languages. *The 31st International Conference on Computational Linguistics*, 31.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. 2018. Unsupervised representation learning by predicting image rotations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael A. Specter, and Lalana Kagal. 2018. Explaining explanations: An overview of interpretability of machine learning. In *5th IEEE International Conference on Data Science and Advanced Analytics, DSAA 2018, Turin, Italy, October 1-3, 2018*, pages 80–89. IEEE.

John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2021. DeCLUTR: Deep contrastive learning for unsupervised textual representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 879–895, Online. Association for Computational Linguistics.

Kyle Glandt, Sarthak Khanal, Yingjie Li, Doina Caragea, and Cornelia Caragea. 2021. Stance detection in COVID-19 tweets. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1596–1611, Online. Association for Computational Linguistics.

Elliot Glazer, Ege Erdil, Tamay Besiroglu, Diego Chicharro, Evan Chen, Alex Gunning, Caroline Falkman Olsson, Jean-Stanislas Denain, Anson Ho, Emily de Oliveira Santos, et al. 2024. Frontiermath: A benchmark for evaluating advanced mathematical reasoning in ai. *arXiv preprint arXiv:2411.04872*.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia. Association for Computational Linguistics.

Micah Goldblum, Hossein Souri, Renkun Ni, Manli Shu, Viraj Prabhu, Gowthami Somepalli, Prithvijit Chattopadhyay, Mark Ibrahim, Adrien Bardes, Judy Hoffman, et al. 2023. Battle of the backbones: A large-scale comparison of pretrained models across computer vision tasks. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Micah Goldblum, Hossein Souri, Renkun Ni, Manli Shu, Viraj Prabhu, Gowthami Somepalli, Prithvijit Chattopadhyay, Mark Ibrahim, Adrien Bardes, Judy Hoffman, et al. 2024. Battle of the backbones: A large-scale comparison of pretrained models across computer vision tasks. *Advances in Neural Information Processing Systems*, 36.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative adversarial networks. *CoRR*, abs/1406.2661.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Priya Goyal, Mathilde Caron, Benjamin Lefaudeux, Min Xu, Pengchao Wang, Vivek Pai, Mannat Singh, Vitaliy Liptchinsky, Ishan Misra, Armand Joulin, et al. 2021a. Self-supervised pretraining of visual features in the wild. *arXiv preprint arXiv:2103.01988*.

Priya Goyal, Quentin Duval, Jeremy Reizenstein, Matthew Leavitt, Min Xu, Benjamin Lefaudeux, Mannat Singh, Vinicius Reis, Mathilde Caron, Piotr Bojanowski, et al. 2021b. Vissl. `https://github.com/facebookresearch/vissl`.

Herbert P Grice. 1975. Logic and conversation. In *Speech acts*, pages 41–58. Brill.

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.

Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, et al. 2018. Recent advances in convolutional neural networks. *Pattern recognition*, 77:354–377.

Nuno M Guerreiro, Duarte M Alves, Jonas Waldendorf, Barry Haddow, Alexandra Birch, Pierre Colombo, and André FT Martins. 2023. Hallucinations in large multilingual translation models. *Transactions of the Association for Computational Linguistics*, 11:1500–1517.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018a. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. 2018b. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 107–112. Association for Computational Linguistics.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2018. The Argument Reasoning Comprehension Task: Identification and Reconstruction of Implicit Warrants. 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 1930–1940, New Orleans, Louisiana. Association for Computational Linguistics.

Perttu Hämäläinen, Mikke Tavast, and Anton Kunnari. 2023. Evaluating large language models in generating synthetic hci research data: a case study. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–19.

Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. 2021. WARP: Word-level Adversarial ReProgramming. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4921–4933, Online. Association for Computational Linguistics.

William L. Hamilton, Payal Bajaj, Marinka Zitnik, Dan Jurafsky, and Jure Leskovec. 2018. Embedding logical queries on knowledge graphs. In *NeurIPS*, pages 2030–2041.

Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. 2022. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):87–110.

Andreas Hanselowski, Avinesh PVS, Benjamin Schiller, Felix Caspelherr, Debanjan Chaudhuri, Christian M. Meyer, and Iryna Gurevych. 2018. A Retrospective Analysis of the Fake News Challenge Stance-Detection Task. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1859–1874, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Andreas Hanselowski, Christian Stab, Claudia Schulz, Zile Li, and Iryna Gurevych. 2019. A richly annotated corpus for different tasks in automated fact-checking. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 493–503, Hong Kong, China. Association for Computational Linguistics.

Momchil Hardalov, Arnav Arora, Preslav Nakov, and Isabelle Augenstein. 2021a. Cross-domain label-adaptive stance detection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9011–9028, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Momchil Hardalov, Arnav Arora, Preslav Nakov, and Isabelle Augenstein. 2021b. Cross-domain label-adaptive stance detection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 9011–9028. Association for Computational Linguistics.

Momchil Hardalov, Arnav Arora, Preslav Nakov, and Isabelle Augenstein. 2022a. A Survey on Stance Detection for Mis- and Disinformation Identification. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1259–1277, Seattle, United States. Association for Computational Linguistics.

Momchil Hardalov, Arnav Arora, Preslav Nakov, and Isabelle Augenstein. 2022b. Few-Shot Cross-Lingual Stance Detection with Sentiment-Based Pre-training. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):10729–10737.

Kazi Saidul Hasan and Vincent Ng. 2013. Stance classification of ideological debates: Data, models, features, and constraints. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1348–1356, Nagoya, Japan. Asian Federation of Natural Language Processing.

Kazi Saidul Hasan and Vincent Ng. 2014. Why are You Taking this Stance? Identifying and Classifying Reasons in Ideological Debates. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 751–762, Doha, Qatar. Association for Computational Linguistics.

Tawfiq Hasanin, Taghi M. Khoshgoftaar, Joffrey L. Leevy, and Richard A. Bauder. 2019. Severely imbalanced big data challenges: investigating data sampling approaches. *J. Big Data*, 6:107.

David Haussler. 1990. Probably approximately correct learning. In *Proceedings of the 8th National Conference on Artificial Intelligence. Boston, Massachusetts, USA, July 29 - August 3, 1990, 2 Volumes*, pages 1101–1108. AAAI Press / The MIT Press.

Haibo He and Edwardo A Garcia. 2009. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021c. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Bernease Herman. 2017. The promise and peril of human evaluation for model interpretability. *arXiv preprint arXiv:1711.07414*.

John Hewitt and Christopher D Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138.

Marcel Hildebrandt, Jorge Andres Quintero Serna, Yunpu Ma, Martin Ringsquandl, Mitchell Joblin, and Volker Tresp. 2020. Reasoning on knowledge graphs with debate dynamics. In *AAAI*, pages 4123–4131. AAAI Press.

Daniel S. Himmelstein, Antoine Lizee, Christine Hessler, Leo Brueggeman, Sabrina L. Chen, Dexter Hadley, Ari Green, Pouya Khankhanian, and Sergio E. Baranzini. 2017. Systematic integration of biomedical knowledge prioritizes drugs for repurposing. *bioRxiv*.

Hideitsu Hino. 2020. Active learning: Problem settings and recent developments. *ArXiv preprint*, abs/2012.04225.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spacy: Industrial-strength natural language processing in python. If you use spaCy, please cite it as below.

Tamanna Hossain, Robert L. Logan IV, Arjuna Ugarte, Yoshitomo Matsubara, Sean Young, and Sameer Singh. 2020. COVIDLies: Detecting COVID-19 misinformation on social media. In *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*, Online. Association for Computational Linguistics.

Xia Hu, Lingyang Chu, Jian Pei, Weiqing Liu, and Jiang Bian. 2021. Model complexity of deep learning: a survey. *Knowl. Inf. Syst.*, 63(10):2585–2619.

Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. 2018. Black-box adversarial attacks with limited queries and information. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2142–2151. PMLR.

V Ivan Sanchez Carmona, Jeff Mitchell, and Sebastian Riedel. 2018. Behavior analysis of nli models: Uncovering the influence of three factors on robustness. In *NAACL*

HLT 2018-2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies-Proceedings of the Conference, volume 2018, pages 1975–1985. Association for Computational Linguistics (ACL).

Pavel Izmailov, Polina Kirichenko, Nate Gruver, and Andrew Gordon Wilson. 2022. On feature learning in the presence of spurious correlations. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Pauline I Jacobson. 2014. *Compositional semantics: An introduction to the syntax/semantics interface*. Oxford Textbooks in Linguistic.

Alon Jacovi, Yonatan Bitton, Bernd Bohnet, Jonathan Herzig, Or Honovich, Michael Tseng, Michael Collins, Roee Aharoni, and Mor Geva. 2024. A chain-of-thought is as strong as its weakest link: A benchmark for verifiers of reasoning chains. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 4615–4634. Association for Computational Linguistics.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4198–4205. Association for Computational Linguistics.

Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. 2020. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does bert learn about the structure of language? In *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics*.

Paloma Jeretic, Alex Warstadt, Suvrat Bhooshan, and Adina Williams. 2020a. Are natural language inference models imppressive? learning implicature and presupposition. *arXiv preprint arXiv:2004.03066*.

Paloma Jeretic, Alex Warstadt, Suvrat Bhooshan, and Adina Williams. 2020b. Are natural language inference models IMPPRESsive? Learning IMPlicature and PRESupposition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8690–8705, Online. Association for Computational Linguistics.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Juyong Jiang, Fan Wang, Jiasi Shen, Sungju Kim, and Sunghun Kim. 2024. A survey on large language models for code generation. *CoRR*, abs/2406.00515.

Zhanming Jie, Trung Quoc Luong, Xinbo Zhang, Xiaoran Jin, and Hang Li. 2023. Design of chain-of-thought in math problem solving. *arXiv preprint arXiv:2309.11054*.

Longlong Jing and Yingli Tian. 2020. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):4037–4058.

Longlong Jing and Yingli Tian. 2021. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(11):4037–4058.

Justin M. Johnson and Taghi M. Khoshgoftaar. 2019a. Survey on deep learning with class imbalance. *J. Big Data*, 6:27.

Justin M Johnson and Taghi M Khoshgoftaar. 2019b. Survey on deep learning with class imbalance. *Journal of big data*, 6(1):1–54.

James M Joyce. 2011. Kullback-leibler divergence. In *International encyclopedia of statistical science*, pages 720–722. Springer.

Ziv Katzir and Yuval Elovici. 2021. Who's afraid of adversarial transferability? *CoRR*, abs/2105.00433.

Harsurinder Kaur, Husanbir Singh Pannu, and Avleen Kaur Malhi. 2019. A systematic review on imbalanced data challenges in machine learning: Applications and solutions. *ACM Comput. Surv.*, 52(4):79:1–79:36.

Koray Kavukcuoglu, Pierre Sermanet, Y-Lan Boureau, Karol Gregor, Michaël Mathieu, and Yann LeCun. 2010. Learning convolutional feature hierarchies for visual recognition. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada*, pages 1090–1098. Curran Associates, Inc.

Michael J Kearns and Umesh Vazirani. 1994. *An introduction to computational learning theory*. MIT press.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, et al. 2020. Measuring compositional generalization: A comprehensive method on realistic data. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Erich-Peter Klement, Radko Mesiar, and Endre Pap. 2000. *Triangular Norms*, volume 8 of *Trends in Logic*. Springer.

Erich-Peter Klement, Radko Mesiar, and Endre Pap. 2004. Triangular norms. position paper I: basic analytical and algebraic properties. *Fuzzy Sets Syst.*, 143(1):5–26.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022a. Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022b. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. 2020. Big transfer (bit): General visual representation learning. In *European conference on computer vision*, pages 491–507. Springer.

Ksenia Konyushkova, Raphael Sznitman, and Pascal Fua. 2017. Learning Active Learning from Data. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4225–4235.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of bert. *arXiv preprint arXiv:1908.08593*.

Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 1106–1114.

Dilek Küçük and Fazli Can. 2020. Stance detection: A survey. *ACM Computing Surveys (CSUR)*, 53(1):1–37.

Dilek Küçük and Fazli Can. 2021. Stance detection: A survey. *ACM Comput. Surv.*, 53(1):12:1–12:37.

Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. 2017. Adversarial examples in the physical world. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Trans. Assoc. Comput. Linguistics*, 7:452–466.

Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. 2018. Canonical tensor decomposition for knowledge base completion. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 2869–2878. PMLR.

Mirko Lai, Alessandra Teresa Cignarella, Delia Irazú Hernández Farías, Cristina Bosco, Viviana Patti, and Paolo Rosso. 2020. Multilingual stance detection in social media political debates. *Comput. Speech Lang.*, 63:101075.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, et al. 2023a. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, et al. 2023b. Measuring faithfulness in chain-of-thought reasoning. *CoRR*, abs/2307.13702.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot cross-lingual transfer with multilingual transformers. *arXiv preprint arXiv:2005.00633*.

Yann LeCun. 2019. 1.1 deep learning hardware: Past, present, and future. In *2019 IEEE International Solid-State Circuits Conference-(ISSCC)*, pages 12–19. IEEE.

Yann LeCun, Yoshua Bengio, and Geoffrey E. Hinton. 2015. Deep learning. *Nat.*, 521(7553):436–444.

Lucas Lehnert, Sainbayar Sukhbaatar, Paul McVay, Michael Rabbat, and Yuandong Tian. 2024. Beyond a*: Better planning with transformers via search dynamics bootstrapping. *CoRR*, abs/2402.14083.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019a. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

Patrick Lewis, Barlas Oğuz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2019b. Mlqa: Evaluating cross-lingual extractive question answering. *arXiv preprint arXiv:1910.07475*.

Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. Paq: 65 million probably-asked questions and what you can do with them. *Transactions of the Association for Computational Linguistics*, 9:1098–1115.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay V. Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. 2022. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Chengshu Li, Jacky Liang, Andy Zeng, Xinyun Chen, Karol Hausman, Dorsa Sadigh, Sergey Levine, Li Fei-Fei, Fei Xia, and Brian Ichter. 2024. Chain of code: Reasoning with a language model-augmented code emulator. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. A simple, fast diverse decoding algorithm for neural generation. *arXiv preprint arXiv:1611.08562*.

Yandong Li, Lijun Li, Liqiang Wang, Tong Zhang, and Boqing Gong. 2019. NATTACK: learning the distributions of adversarial examples for an improved black-box attack on deep neural networks. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 3866–3876. PMLR.

Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. 2023. Synthetic data generation with large language models for text classification: Potential and limitations. *arXiv preprint arXiv:2310.07849*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: common objects in context. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, volume 8693 of *Lecture Notes in Computer Science*, pages 740–755. Springer.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 158–167. Association for Computational Linguistics.

Zhan Ling, Yunhao Fang, Xuanlin Li, Zhiao Huang, Mingu Lee, Roland Memisevic, and Hao Su. 2023. Deductive verification of chain-of-thought reasoning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Carolyn E. Lipscomb. 2000. Medical subject headings (mesh). *Bull Med Libr Assoc.* 88(3): 265–266.

Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi Zhang, Joydeep Biswas, and Peter Stone. 2023. LLM+P: empowering large language models with optimal planning proficiency. *CoRR*, abs/2304.11477.

Fang Liu, Yang Liu, Lin Shi, Houkun Huang, Ruifeng Wang, Zhen Yang, and Li Zhang. 2024. Exploring and evaluating hallucinations in llm-powered code generation. *CoRR*, abs/2404.00971.

Jiahua Liu, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2019a. Xqa: A cross-lingual open-domain question answering dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2358–2368.

Nelson F Liu, Matt Gardner, Yonatan Belinkov, Matthew E Peters, and Noah A Smith. 2019b. Linguistic knowledge and transferability of contextual representations. *arXiv preprint arXiv:1903.08855*.

Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. 2021. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019c. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

John W. Lloyd. 1994. Practical advtanages of declarative programming. In *1994 Joint Conference on Declarative Programming, GULP-PRODE'94 Peñiscola, Spain, September 19-22, 1994, Volume 1*, pages 18–30.

Nicholas A. Lord, Romain Müller, and Luca Bertinetto. 2022. Attacking deep networks with surrogate-based adversarial black-box methods is easy. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, IJCNLP 2023 -Volume 1: Long Papers, Nusa Dua, Bali, November 1 - 4, 2023*, pages 305–329. Association for Computational Linguistics.

Aman Madaan and Amir Yazdanbakhsh. 2022a. Text and patterns: For effective chain of thought, it takes two to tango. *arXiv preprint arXiv:2209.07686*.

Aman Madaan and Amir Yazdanbakhsh. 2022b. Text and patterns: For effective chain of thought, it takes two to tango. *CoRR*, abs/2209.07686.

Aman Madaan, Shuyan Zhou, Uri Alon, Yiming Yang, and Graham Neubig. 2022. Language models of code are few-shot commonsense learners. *arXiv preprint arXiv:2210.07128*.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Andreas Madsen, Siva Reddy, and Sarath Chandar. 2022. Post-hoc interpretability for neural nlp: A survey. *ACM Computing Surveys*, 55(8):1–42.

Frank J. Massey. 1951. The Kolmogorov-Smirnov Test for Goodness of Fit. *Journal of the American Statistical Association*, 46(253):68–78.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019a. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3428–3448. Association for Computational Linguistics.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019b. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy. Association for Computational Linguistics.

María Luisa Menéndez, JA Pardo, L Pardo, and MC Pardo. 1997. The jensen-shannon divergence. *Journal of the Franklin Institute*, 334(2):307–318.

Shen-Yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2020. A diverse corpus for evaluating and developing english math word problem solvers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 975–984. Association for Computational Linguistics.

George A. Miller. 1992. WORDNET: a lexical database for english. In *HLT*. Morgan Kaufmann.

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. 2023. Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Computing Surveys*, 56(2):1–40.

Pasquale Minervini, Erik Arakelyan, Daniel Daza, and Michael Cochez. 2022. Complex query answering with neural link predictors (extended abstract). In *IJCAI*, pages 5309–5313. ijcai.org.

Pasquale Minervini, Sebastian Riedel, Pontus Stenetorp, Edward Grefenstette, and Tim Rocktäschel. 2020. Learning reasoning strategies in end-to-end differentiable proving. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 6938–6949. PMLR.

Ishan Misra and Laurens van der Maaten. 2020. Self-supervised learning of pretext-invariant representations. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 6706–6716. Computer Vision Foundation / IEEE.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. SemEval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 31–41, San Diego, California. Association for Computational Linguistics.

Saif M Mohammad, Parinaz Sobhani, and Svetlana Kiritchenko. 2017. Stance and sentiment in tweets. *ACM Transactions on Internet Technology (TOIT)*, 17(3):1–23.

Seungyong Moon, Gaon An, and Hyun Oh Song. 2019. Parsimonious black-box adversarial attacks via efficient combinatorial optimization. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 4636–4645. PMLR.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. 2017. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. Deepfool: A simple and accurate method to fool deep neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2574–2582. IEEE Computer Society.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. On the stability of fine-tuning BERT: misconceptions, explanations, and strong baselines. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018a. Stress test evaluation for natural language inference. *arXiv preprint arXiv:1806.00692*.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018b. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Nina Narodytska and Shiva Prasad Kasiviswanathan. 2016. Simple black-box adversarial perturbations for deep networks. *CoRR*, abs/1612.06299.

Muzammal Naseer, Kanchana Ranasinghe, Salman Khan, Fahad Shahbaz Khan, and Fatih Porikli. 2022. On improving adversarial transferability of vision transformers. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Alejandro Newell and Jia Deng. 2020. How useful is self-supervised pretraining for visual tasks? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7345–7354.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2016. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2019. Adversarial nli: A new benchmark for natural language understanding. *arXiv preprint arXiv:1910.14599*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4885–4901. Association for Computational Linguistics.

Erik Nijkamp, Bo Pang, Hiroaki Hayashi, Lifu Tu, Huan Wang, Yingbo Zhou, Silvio Savarese, and Caiming Xiong. 2023. Codegen: An open large language model for code with multi-turn program synthesis. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Maria-Elena Nilsback and Andrew Zisserman. 2008. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE.

Mehdi Noroozi and Paolo Favaro. 2016. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI*, volume 9910 of *Lecture Notes in Computer Science*, pages 69–84. Springer.

Natalya Fridman Noy, Yuqing Gao, Anshu Jain, Anant Narayanan, Alan Patterson, and Jamie Taylor. 2019. Industry-scale knowledge graphs: lessons and challenges. *Commun. ACM*, 62(8):36–43.

Moritz Osnabrügge, Elliott Ash, and Massimo Morelli. 2023. Cross-domain topic classification for political texts. *Political Analysis*, 31(1):59–80.

Malte Ostendorff, Nils Rethmeier, Isabelle Augenstein, Bela Gipp, and Georg Rehm. 2022. Neighborhood Contrastive Learning for Scientific Document Representations with Citation Embeddings. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11670–11688, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Yang Wang. 2023. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 3806–3824. Association for Computational Linguistics.

Nicolas Papernot, Patrick D. McDaniel, Ian J. Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. 2017. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, AsiaCCS 2017, Abu Dhabi, United Arab Emirates, April 2-6, 2017*, pages 506–519. ACM.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. 2012. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021a. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online. Association for Computational Linguistics.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021b. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 2080–2094. Association for Computational Linguistics.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models. *Advances in Neural Information Processing Systems*, 34:11054–11070.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. *arXiv preprint arXiv:1805.01042*.

Dean Pomerleau and Delip Rao. 2017. Fake news challenge stage 1 (FNC-I): Stance detection. *URL www. fakenewschallenge. org*.

V Porkodi, M Sivaram, Amin Salih Mohammed, and V Manikandan. 2018. Survey on white-box attacks and solutions. *Asian Journal of Computer Science and Technology*, 7(3):28–32.

Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge J. Belongie. 2018. Generative adversarial perturbations. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 4422–4431. Computer Vision Foundation / IEEE Computer Society.

Vahed Qazvinian, Emily Rosengren, Dragomir R. Radev, and Qiaozhu Mei. 2011. Rumor has it: Identifying Misinformation in Microblogs. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1589–1599, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Qi Qian, Lei Shang, Baigui Sun, Juhua Hu, Tacoma Tacoma, Hao Li, and Rong Jin. 2019. Softtriple loss: Deep metric learning without triplet sampling. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 6449–6457. IEEE.

Yunxiao Qin, Yuanhao Xiong, Jinfeng Yi, and Cho-Jui Hsieh. 2023. Training meta-surrogate model for transferable adversarial attack. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 9516–9524. AAAI Press.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer.

Sara Rajaee, Yadollah Yaghoobzadeh, and Mohammad Taher Pilehvar. 2022. Looking at the overlooked: An analysis on the word-overlap bias in natural language inference. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 10605–10616. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Leonardo Ranaldi and Giulia Pucci. 2023. Does the English matter? elicit cross-lingual abilities of large language models. In *Proceedings of the 3rd Workshop on Multilingual Representation Learning (MRL)*, pages 173–183, Singapore. Association for Computational Linguistics.

Hiranmayi Ranganathan, Hemanth Venkateswara, Shayok Chakraborty, and Sethuraman Panchanathan. 2017. Deep active learning for image classification. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3934–3938. IEEE.

Abhilasha Ravichander, Aakanksha Naik, Carolyn Rose, and Eduard Hovy. 2019. EQUATE: A benchmark evaluation framework for quantitative reasoning in natural language inference. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 349–361, Hong Kong, China. Association for Computational Linguistics.

Emily Reif, Daphne Ippolito, Ann Yuan, Andy Coenen, Chris Callison-Burch, and Jason Wei. 2022. A recipe for arbitrary text style transfer with large language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 837–848. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Hongyu Ren, Weihua Hu, and Jure Leskovec. 2020. Query2box: Reasoning over knowledge graphs in vector space using box embeddings. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Hongyu Ren and Jure Leskovec. 2020. Beta embeddings for multi-hop logical reasoning in knowledge graphs. *Advances in Neural Information Processing Systems*, 33:19716–19726.

Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. 2021. A survey of deep active learning. *ACM computing surveys (CSUR)*, 54(9):1–40.

Erendira Rendon, Roberto Alejo, Carlos Castorena, Frank J Isidro-Ortega, and Everardo E Granda-Gutierrez. 2020. Data sampling methods to deal with the big data multi-class imbalance problem. *Applied Sciences*, 10(4):1276.

Nils Rethmeier and Isabelle Augenstein. 2023. A Primer on Contrastive Pretraining in Language Processing: Methods, Lessons Learned, and Perspectives. *ACM Comput. Surv.*, 55(10).

Myrthe Reuver, Antske Fokkens, and Suzan Verberne. 2021. No NLP task should be an island: Multi-disciplinarity for diversity in news recommender systems. In *Proceedings of the EACL Hackashop on News Media Content Analysis and Automated Report Generation*, pages 45–55, Online. Association for Computational Linguistics.

Arij Riabi, Thomas Scialom, Rachel Keraron, Benoît Sagot, Djamé Seddah, and Jacopo Staiano. 2020. Synthetic data augmentation for zero-shot cross-lingual question answering. *arXiv preprint arXiv:2010.12643*.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. *arXiv preprint arXiv:2005.04118*.

Kyle Richardson, Hai Hu, Lawrence Moss, and Ashish Sabharwal. 2020. Probing natural language inference models through semantic fragments. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8713–8721.

Anna Rogers, Matt Gardner, and Isabelle Augenstein. 2023. Qa dataset explosion: A taxonomy of nlp resources for question answering and reading comprehension. *ACM Computing Surveys*, 55(10):1–45.

Joost Rommers, Ton Dijkstra, and Marcel Bastiaansen. 2013. Context-dependent semantic processing in the human brain: Evidence from idiom comprehension. *Journal of Cognitive Neuroscience*, 25(5):762–776.

Corby Rosset, Chenyan Xiong, Minh Phan, Xia Song, Paul Bennett, and Saurabh Tiwary. 2020. Knowledge-aware language model pretraining. *arXiv preprint arXiv:2007.00655*.

Subhro Roy and Dan Roth. 2015. Solving general arithmetic word problems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1743–1752. The Association for Computational Linguistics.

Daniel Ruffinelli, Samuel Broscheit, and Rainer Gemulla. 2020. You CAN teach an old dog new tricks! on training knowledge graph embeddings. In *ICLR*. OpenReview.net.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(3):211–252.

Ali Sadeghian, Mohammadreza Armandpour, Patrick Ding, and Daisy Zhe Wang. 2019. Drum: End-to-end differentiable rule mining on knowledge graphs. *Advances in Neural Information Processing Systems*, 32.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, et al. 2022. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Abulhair Saparov and He He. 2023. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. 2019. Analysing mathematical reasoning abilities of neural models. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Stephen Schiffer. 1986. Compositional semantics and language understanding. *Philosophical Grounds of Rationality: Intentions, Categories, Ends, Oxford*, pages 174–207.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer.

Tal Schuster, Darsh J. Shah, Yun Jie Serene Yeo, Daniel Filizzola, Enrico Santus, and Regina Barzilay. 2019. Towards debiasing fact verification models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3417–3423. Association for Computational Linguistics.

Michael Scriven. 1976. Reasoning.

Siamak Shakeri, Noah Constant, Mihir Sanjay Kale, and Linting Xue. 2020. Towards zero-shot multilingual synthetic question and answer generation for cross-lingual reading comprehension. *arXiv preprint arXiv:2010.12008*.

Jingyu Shao, Qing Wang, and Fangbing Liu. 2019. Learning to sample: an active learning framework. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 538–547. IEEE.

Koustuv Sinha, Prasanna Parthasarathi, Joelle Pineau, and Adina Williams. 2020. Unnatural language inference. *arXiv preprint arXiv:2101.00010*.

Koustuv Sinha, Prasanna Parthasarathi, Joelle Pineau, and Adina Williams. 2021. Unnatural language inference.

Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. 2019. CLUTRR: A diagnostic benchmark for inductive reasoning from text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4505–4514. Association for Computational Linguistics.

Parinaz Sobhani, Diana Inkpen, and Stan Matwin. 2015. From Argumentation Mining to Stance Classification. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 67–77, Denver, CO. Association for Computational Linguistics.

Parinaz Sobhani, Diana Inkpen, and Xiaodan Zhu. 2017. A dataset for multi-target stance detection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 551–557, Valencia, Spain. Association for Computational Linguistics.

Swapna Somasundaran and Janyce Wiebe. 2010. Recognizing Stances in Ideological On-Line Debates. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 116–124, Los Angeles, CA. Association for Computational Linguistics.

Zayne Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. 2024. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. *arXiv preprint arXiv:2409.12183*.

Lars St, Svante Wold, et al. 1989. Analysis of variance (anova). *Chemometrics and intelligent laboratory systems*, 6(4):259–272.

Christian Stab, Tristan Miller, Benjamin Schiller, Pranav Rai, and Iryna Gurevych. 2018. Cross-topic argument mining from heterogeneous sources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3664–3674, Brussels, Belgium. Association for Computational Linguistics.

Robert Stalnaker, Milton K Munitz, and Peter Unger. 1977. Pragmatic presuppositions. In *Proceedings of the Texas conference on per˜ formatives, presuppositions, and implicatures. Arlington, VA: Center for Applied Linguistics*, pages 135–148. ERIC.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *WWW*, pages 697–706. ACM.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, et al. 2023. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051, Toronto, Canada. Association for Computational Linguistics.

Ali Raza Syed, Andrew Rosenberg, and Ellen Kislal. 2016. Supervised and unsupervised active learning for automatic speech recognition of low-resource languages. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*, pages 5320–5324. IEEE.

Vivienne Sze, Yu-Hsin Chen, Joel Emer, Amr Suleiman, and Zhengdong Zhang. 2017. Hardware for machine learning: Challenges and opportunities. In *2017 IEEE custom integrated circuits conference (CICC)*, pages 1–8. IEEE.

Christian Szegedy. 2020. A promising path towards autoformalization and general artificial intelligence. In *Intelligent Computer Mathematics - 13th International Conference, CICM 2020, Bertinoro, Italy, July 26-31, 2020, Proceedings*, volume 12236 of *Lecture Notes in Computer Science*, pages 3–20. Springer.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 641–651. Association for Computational Linguistics.

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2020. olmpics - on what language model pre-training captures. *Trans. Assoc. Comput. Linguistics*, 8:743–758.

Chameleon Team. 2024. Chameleon: Mixed-modal early-fusion foundation models. *CoRR*, abs/2405.09818.

Komal Teru, Etienne Denis, and Will Hamilton. 2020. Inductive relation prediction by subgraph reasoning. In *International Conference on Machine Learning*, pages 9448–9457. PMLR.

Matt Thomas, Bo Pang, and Lillian Lee. 2006. Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 327–335, Sydney, Australia. Association for Computational Linguistics.

Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. 2016. YFCC100M: the new data in multimedia research. *Commun. ACM*, 59(2):64–73.

Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 57–66, Beijing, China. Association for Computational Linguistics.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. 2021. Training data-efficient image transformers & distillation through attention. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 10347–10357. PMLR.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023a. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *ICML*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2071–2080. JMLR.org.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Karthik Valmeekam, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. 2022. Large language models still can't plan (a benchmark for llms on planning and reasoning about change). In *NeurIPS 2022 Foundation Models for Decision Making Workshop*.

Vladimir N Vapnik. 1999. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. 2019a. Composition-based multi-relational graph convolutional networks. *arXiv preprint arXiv:1911.03082*.

Shikhar Vashishth, Shyam Upadhyay, Gaurav Singh Tomar, and Manaal Faruqui. 2019b. Attention interpretability across nlp tasks. *arXiv preprint arXiv:1909.11218*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017a. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017b. Attention is all you need. *Advances in neural information processing systems*, 30.

Palak Verma, Neha Shukla, and AP Shukla. 2021. Techniques of sarcasm detection: A review. In *2021 international conference on advance computing and innovative technologies in engineering (ICACITE)*, pages 968–972. IEEE.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledge base. *Communications of the ACM*, 57:78–85.

Tu Vu, Tong Wang, Tsendsuren Munkhdalai, Alessandro Sordoni, Adam Trischler, Andrew Mattarella-Micke, Subhransu Maji, and Mohit Iyyer. 2020. Exploring and predicting transferability across NLP tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7882–7926, Online. Association for Computational Linguistics.

Marilyn Walker, Jean Fox Tree, Pranav Anand, Rob Abbott, and Joseph King. 2012. A Corpus for Research on Deliberation and Debate. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 812–817, Istanbul, Turkey. European Language Resources Association (ELRA).

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do nlp models know numbers? probing numeracy in embeddings. *arXiv preprint arXiv:1909.07940*.

Douglas N Walton. 1990. What is reasoning? what is an argument? *The journal of Philosophy*, 87(8):399–419.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019b. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3261–3275.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018a. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023a. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 2609–2634. Association for Computational Linguistics.

Qingxiang Wang, Cezary Kaliszyk, and Josef Urban. 2018b. First experiments with neural translation of informal to formal mathematics. In *Intelligent Computer Mathematics - 11th International Conference, CICM 2018, Hagenberg, Austria, August 13-17, 2018, Proceedings*, volume 11006 of *Lecture Notes in Computer Science*, pages 255–270. Springer.

Rui Wang, Deyu Zhou, Mingmin Jiang, Jiasheng Si, and Yang Yang. 2019c. A survey on opinion mining: From stance to product aspect. *IEEE Access*, 7:41101–41124.

Shuohang Wang and Jing Jiang. 2015. Learning natural language inference with lstm. *arXiv preprint arXiv:1512.08849*.

Tianlu Wang, Rohit Sridhar, Diyi Yang, and Xuezhi Wang. 2022a. Identifying and mitigating spurious correlations for improving robustness in NLP models. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1719–1729, Seattle, United States. Association for Computational Linguistics.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022b. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Zhao Wang and Aron Culotta. 2020. Identifying spurious correlations for robust text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3431–3440, Online. Association for Computational Linguistics.

Alex Warstadt and Samuel R Bowman. 2020. Can neural networks acquire a structural bias from raw linguistic data? *arXiv preprint arXiv:2007.06761*.

Futa Waseda, Sosuke Nishikawa, Trung-Nghia Le, Huy H. Nguyen, and Isao Echizen. 2023. Closer look at the transferability of adversarial examples: How they fool different models differently. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2023, Waikoloa, HI, USA, January 2-7, 2023*, pages 1360–1368. IEEE.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022a. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022b. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Penghui Wei, Junjie Lin, and Wenji Mao. 2018. Multi-Target Stance Detection via a Dynamic Memory-Augmented Network. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pages 1229–1232. ACM.

Penghui Wei and Wenji Mao. 2019. Modeling Transferable Topics for Cross-Target Stance Detection. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, pages 1173–1176. ACM.

Jan Wielemaker, Tom Schrijvers, Markus Triska, and Torbjörn Lager. 2012. Swi-prolog. *Theory and Practice of Logic Programming*, 12(1-2):67–96.

Wikipedia. 2024. Countdown (game show) — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Countdown%20(game%20show)&oldid=1248084922. [Online; accessed 09-September-2024].

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yuhuai Wu, Albert Qiaochu Jiang, Wenda Li, Markus N. Rabe, Charles Staats, Mateja Jamnik, and Christian Szegedy. 2022. Autoformalization with large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 3733–3742. Computer Vision Foundation / IEEE Computer Society.

Wenhan Xiong, Thien Hoang, and William Yang Wang. 2017. Deeppath: A reinforcement learning method for knowledge graph reasoning. In *EMNLP*, pages 564–573. Association for Computational Linguistics.

Xuyang Yan, Shabnam Nazmi, Biniam Gebru, Mohd Anwar, Abdollah Homaifar, Mrinmoy Sarkar, and Kishor Datta Gupta. 2022. Mitigating shortage of labeled data using clustering-based active learning with diversity exploration. *ArXiv preprint*, abs/2207.02964.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. ConSERT: A contrastive framework for self-supervised sentence representation transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5065–5075, Online. Association for Computational Linguistics.

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos. 2019. Can neural networks understand monotonicity reasoning? *arXiv preprint arXiv:1906.06448*.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014a. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

Fan Yang, Zhilin Yang, and William W Cohen. 2017. Differentiable learning of logical rules for knowledge base reasoning. *Advances in neural information processing systems*, 30.

Mengjiao Yang, Dale Schuurmans, Pieter Abbeel, and Ofir Nachum. 2022. Chain of thought imitation with procedure cloning. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Yi Yang, Zhigang Ma, Feiping Nie, Xiaojun Chang, and Alexander G Hauptmann. 2015a. Multi-class active learning by uncertainty sampling with diversity maximization. *International Journal of Computer Vision*, 113(2):113–127.

Yi Yang, Shimei Pan, Doug Downey, and Kunpeng Zhang. 2014b. Active learning with constrained topic model. In *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, pages 30–33, Baltimore, Maryland, USA. Association for Computational Linguistics.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015b. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2013–2018.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380. Association for Computational Linguistics.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023b. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Junjie Ye, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen, et al. 2023. A comprehensive capability analysis of GPT-3 and GPT-3.5 series models. *CoRR*, abs/2303.10420.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. *arXiv preprint arXiv:1909.00161*.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. 2022. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*.

Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. 2019. A review of recurrent neural networks: LSTM cells and network architectures. *Neural Comput.*, 31(7):1235–1270.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. Swag: A large-scale adversarial dataset for grounded commonsense inference. *arXiv preprint arXiv:1808.05326*.

Richard Zhang, Phillip Isola, and Alexei A. Efros. 2016. Colorful image colorization. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III*, volume 9907 of *Lecture Notes in Computer Science*, pages 649–666. Springer.

Yadong Zhang, Shaoguang Mao, Tao Ge, Xun Wang, Adrian de Wynter, Yan Xia, Wenshan Wu, Ting Song, Man Lan, and Furu Wei. 2024. LLM as a mastermind: A survey of strategic reasoning with large language models. *CoRR*, abs/2404.01230.

Zhanqiu Zhang, Jie Wang, Jiajun Chen, Shuiwang Ji, and Feng Wu. 2021. Cone: Cone embeddings for multi-hop reasoning over knowledge graphs. *Advances in Neural Information Processing Systems*, 34:19172–19183.

Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2020. Semantics-aware bert for language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9628–9635.

Wanjun Zhong, Siyuan Wang, Duyu Tang, Zenan Xu, Daya Guo, Jiahai Wang, Jian Yin, Ming Zhou, and Nan Duan. 2021. AR-LSAT: investigating analytical reasoning of text. *CoRR*, abs/2104.06598.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V. Le, et al. 2023a. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Giulio Zhou and Gerasimos Lampouras. 2021. Informed sampling for diversity in concept-to-text NLG. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2494–2509, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jiawei Zhou, Yixuan Zhang, Qianni Luo, Andrea G Parker, and Munmun De Choudhury. 2023b. Synthetic lies: Understanding ai-generated misinformation and evaluating algorithmic and human solutions. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–20.

Jingbo Zhu, Huizhen Wang, Tianshun Yao, and Benjamin K Tsou. 2008. Active learning with sampling by uncertainty and density for word sense disambiguation and text classification. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 1137–1144, Manchester, UK. Coling 2008 Organizing Committee.

Yao Zhu, Yuefeng Chen, Xiaodan Li, Kejiang Chen, Yuan He, Xiang Tian, Bolun Zheng, Yaowu Chen, and Qingming Huang. 2022a. Toward understanding and boosting adversarial transferability from a distribution perspective. *IEEE Trans. Image Process.*, 31:6487–6501.

Zhaocheng Zhu, Mikhail Galkin, Zuobai Zhang, and Jian Tang. 2022b. Neural-symbolic models for logical queries on knowledge graphs. *arXiv preprint arXiv:2205.10128*.

Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, and Jian Tang. 2021. Neural bellman-ford networks: A general graph neural network framework for link prediction. *Advances in Neural Information Processing Systems*, 34:29476–29490.

Adrian Ziegler and Yuki M Asano. 2022. Self-supervised learning of object parts for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14502–14511.

Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter. 2018a. Detection and resolution of rumours in social media: A survey. *ACM Computing Surveys (CSUR)*, 51(2):1–36.

Arkaitz Zubiaga, Elena Kochkina, Maria Liakata, Rob Procter, Michal Lukasik, Kalina Bontcheva, Trevor Cohn, and Isabelle Augenstein. 2018b. Discourse-aware rumour stance classification in social media using sequential classifiers. *Information Processing and Management*, 54(2):273–290.