University of Energy And Natural Resources, Sunyani
SCHOOL OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATICS

MITIGATING HALLUCINATION IN LARGE LANGUAGE MODELS: A MODULAR FRAMEWORK FOR DETECTION AND COUNTERFACTUAL CORRECTION

BY

Bruce Asare Nyantakyi

December 3, 2025

MITIGATING HALLUCINATION IN LARGE LANGUAGE MODELS: A MODULAR FRAMEWORK FOR DETECTION AND COUNTERFACTUAL CORRECTION

BY

Bruce Asare Nyantakyi

MSC. COMPUTER SCIENCE

A THESIS SUBMITTED TO THE DEPARTMENT OF

COMPUTER SCIENCE AND INFORMATICS

SCHOOL OF SCIENCES

UNIVERSITY OF ENERGY AND NATURAL RESOURCES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF

MASTER OF SCIENCE

IN

COMPUTER SCIENCE

December 3, 2025

DECLARATION

I, Nyantakyi Asare Bruce (UEMS1200224), hereby declare that, except for the references cited which have been duly acknowledged, this submission is my own work toward a Master of Science in Computer Science degree, and that to the best of my knowledge, it contains no materials previously published by another person. I also declare that this has not been presented either in whole or in part for another degree in this University or elsewhere.

Signature: _____ Date: _____

Nyantakyi Asare Bruce

(UEMS1200224)

(Student)

Signature: _____ Date: _____

Dr. Mighty Abra Ayidzoe

(Supervisor)

Signature: _____ Date: _____

Dr. Peter Nimbe

(Co-Supervisor)

Signature: _____ Date: _____

Prof. Patrick Kwabena Mensah

(Head of Department of Computer Science and Informatics)

Abstract

Large Language Models (LLMs) demonstrate impressive fluency yet remain unreliable in safetycritical environments due to persistent hallucination; confidently generating factually incorrect or semantically not supported answers. This research proposes a modular mitigation framework integrating Hallucination Potential Minimization (HPM) with Self-Generated Counterfactual Training (SGCT) to improve factual consistency in generative outputs. A lightweight DistilBERT-based HPM classifier was trained as a binary factuality judge using benchmark datasets including FEVER and TruthfulQA, prioritising recall to ensure conservative hallucination detection. Building on this foundation, SGCT fine-tuned a GPT-2 generative model rather than more recent architectures due to its computational accessibility, reproducibility, and suitability for controlled experimentation under resource constraints. SGCT incorporates likelihood loss for factual responses, unlikelihood loss to penalize hallucinations, and a contrastive objective to separate factual versus hallucinated answers representations in an embedding space.

Experimental results demonstrated measurable improvements following SGCT, with accuracy increasing from 0.556 to 0.614, recall from 0.705 to 0.890, precision from 0.532 to 0.548, and F1-score from 0.607 to 0.692. Threshold calibration further revealed flexible trade-offs between factuality and output strictness, enabling uncertain responses to be routed into a safe “abstain” category. The findings indicate that classifier-guided generation provides a practical strategy for enhancing reliability in LLM-based systems while maintaining computational efficiency. The proposed SGCT-HPM pipeline represents a reproducible and adaptable approach for hallucination mitigation, with potential applications in domains requiring verifiable AI-generated content.

DEDICATION

This thesis is dedicated to my family, whose steadfast love, patience, and encouragement have been the foundation of my academic and personal journey. To my mother, whose example instilled in me the values of diligence and perseverance, and to my siblings, whose constant

reminders of resilience and faith have strengthened my resolve. I also dedicate this work to every student and researcher who dares to explore new horizons and pursue dreams beyond conventional limits. May this thesis serve as a testament that with determination, support, and belief, remarkable achievements are always within reach.

ACKNOWLEDGEMENT

I wish to begin by expressing my profound gratitude to God Almighty for granting me the strength, wisdom, and perseverance that sustained me throughout this research endeavor. My sincere appreciation goes to my supervisors, Dr. Mighty Abra Ayidzoe and Dr. Peter Nimbe, for their invaluable guidance, encouragement, and constructive feedback. Their scholarly insights not only shaped the direction of this thesis but also enriched my growth as a researcher. I remain deeply indebted to my family for their unwavering love, sacrifices, and belief in my potential. Their support provided the motivation I needed to persevere through moments of challenge. I am equally grateful to my friends and colleagues, whose thoughtful discussions, encouragement, and companionship made this journey both intellectually rewarding and personally fulfilling. Finally, I acknowledge the University of Energy and Natural Resources for fostering an enabling academic environment and for providing the resources necessary to complete this work. To all who contributed, in ways great or small, to the realization of this thesis, I extend my deepest appreciation.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background of Study

The emergence of large language models (LLMs), beginning with GPT-2 and continuing through more advanced versions, has transformed natural language processing (NLP) and expanded its applications across diverse fields. Due to their broad domain coverage, LLMs are now applied in areas such as academic research, programming, creative writing, technical advising, and skill development. They have therefore become integral to everyday activities, serving as tools for generating accurate and reliable information. By processing vast amounts of internet data, these models are able to produce grammatically coherent text that aligns with contemporary discussions, often achieving a level of fluency comparable to human writers in specific domains. Despite these strengths, LLMs remain prone to generating inaccurate content. This technical problem, known as hallucination, undermines trust in AI outputs and poses serious challenges when such systems are applied in sensitive fields including education, healthcare, policy-making, law, and journalism.

Nevertheless, a central limitation of large language models is their tendency to produce inaccurate or fabricated information about real-world topics. This phenomenon, commonly known as hallucination, remains a critical challenge for researchers in the field. Even advanced models such as GPT-4 may generate references that are misleading or entirely unfounded (Rawte et al., 2023). Hallucination often arises from the pattern-based generation techniques used during training, combined with the lack of access to real-time internet updates or continuously refreshed data, which leads to inconsistencies in the information produced (Ray, 2023).

1.2 Problem Statement

While there have been significant improvements, the processes for currently confirming factual accuracy in large language models (LLMs) continue to have significant limitations with respect to their scope and scalability. Most primaries rely on post-hoc verification filters or expensive humanin-the-loop tasks. These methods may provide some effectiveness, but their lack of common protocols proves to be relatively impossible to generalize to other use cases. Most importantly, these techniques usually do not identify the initial generative processes causing factual errors, otherwise known as ”hallucinations.” As a result, although the LLM output may be highly lexically-coherent and persuasive, it often still contains semantically false information. The lack of a common architecture that not only predicts, but also reduces these hallucinations in the generation process, remains a major barrier to ultimately developing truly reliable and trustworthy AI systems.

1.3 Objectives of the Study

The main objective of this study is to design, implement, and empirically evaluate a modular two-part framework for hallucination mitigation in large language models. The proposed framework comprises a Hallucination Detection module based on Hallucination Potential Minimization (HPM), and a Counterfactual Correction module implemented through Self-Generated Counterfactual Training (SGCT).

1.3.1 Specific Objectives

The study is guided by the following specific objectives:

1. To develop a hallucination detection model using Hallucination Potential Minimization (HPM) capable of distinguishing factual from hallucinated responses.

2. To implement a Self-Generated Counterfactual Training (SGCT) mechanism that improves the factual consistency of generated outputs.

3. To empirically evaluate the effectiveness of the integrated SGCT–HPM framework using benchmark datasets and standard evaluation metrics.

4. To examine the effect of threshold calibration on factuality–coverage trade-offs within the proposed framework.

1.4 Research Questions

To achieve the specific objectives outlined above, the study addresses the following research questions:

1. To what extent can hallucinated outputs be automatically identified by the HPM classifier?

2. Does the application of SGCT significantly improve the factual consistency of generated responses?

3. How does the integrated SGCT–HPM model perform relative to the baseline generator on metrics such as Accuracy, Precision, Recall, and F1-score?

4. What performance trade-offs arise when varying factuality thresholds in the generated outputs?

1.5 Significance of the Study

Although the hallucination problem has been acknowledged in various studies, few approaches have attempted to resolve it through a fully automated, generative-corrective architecture. By focusing on the synergy between classification-based detection and counterfactual training, this study makes both theoretical and methodological contributions to emerging efforts aimed at enhancing the factual reliability of large language models (LLMs). It also holds practical significance for developers who want to deploy language models in domains where factually

accurate statements are domain requirements. The modularity of the paper allows for modifications to other languages and task-specific contexts. This enhances the value across many subfields of computational linguistics.

1.6 Scope of Study

The research focuses on English-language text and uses two well-established datasets: FEVER, which is built around fact verification tasks, and TruthfulQA, which evaluates model responses to adversarially crafted queries. The classification model for HPM is DistilBERT, while GPT-2 is employed as the basis for counterfactual generation. The evaluation metrics are limited to accuracy, F1-score, and a hybrid metric called the H-Score, designed to measure the combined performance in both factuality and fluency.

1.7 Limitations of Study

Several constraints shape the contours of this research. First, although the datasets used are robust, they do not fully capture the diversity of hallucination types that can occur in open-ended generation. Second, the counterfactual generation process depends on synthetic transformations that, despite being automated, may not always maintain the nuances of the original prompt. Third, due to computational limitations, model training is often confined to a subset of the available data, which may impact the model's generalizability.

1.8 Outline of Thesis

The remainder of the thesis is organized as follows. Chapter Two provides a review of the literature on hallucination, factuality detection, and model alignment methods. Chapter Three provides an overview of the methods used in the thesis. This describes the process of data preparation, the development of models, and evaluations. In Chapter Four, the findings and experimental results followed by discussion and reflection on results and implications will be

presented. In Chapter Five, the paper will provide a summary of findings, reflection on the limitations of the research, and suggestions for future research.

Chapter 2

Literature Review

2.1 Introduction

Large Language Models (LLMs) are now central in modern AI applications ranging from conversational agents to biomedical question answering. However, a fundamental limitation remains: their propensity to hallucinate by producing fluent but inaccurate, misleading, or unverifiable statements. This chapter critically examines strategies for hallucination mitigation in LLMs and synthesizes existing approaches into five major categories: decoding-level strategies, knowledge-grounding using structured resources such as knowledge graphs, faithfulness-based loss optimization, supervised fine-tuning mechanisms, and Retrieval-Augmented Generation (RAG). Rather than merely summarizing existing work, this review highlights the contributions of previous studies, identifies unresolved research gaps, and positions the proposed SGCT–HPM framework within current research directions.

2.2 Conceptualising Hallucination in Language Models

Hallucination in language models may broadly be divided into two types: intrinsic hallucinations, which contradict or misinterpret the given input (e.g., inaccuracies relative to an evidence paragraph), and extrinsic hallucinations, which involve unsupported or unverifiable claims lacking grounding in any source.

From a theoretical perspective, hallucination originates from the probabilistic generative mechanisms underpinning LLMs, which optimize for the most probable next token rather than the most truthful one. Since these models are not explicitly trained to prioritize factuality, they

frequently conflate statistical plausibility with factual correctness. Epistemologically, hallucination reflects a tension between parametric memory (knowledge encoded during pretraining) and real-time external truth that may evolve independently of what the model has seen.

Operationalizing hallucination remains an ongoing challenge. Existing evaluation frameworks employ either binary assessments such as AIS (Attributable to Identified Sources) or scalar scoring techniques such as FactScore and FaithScore to capture graded levels of alignment between model outputs and ground truth evidence. Beyond scoring systems, recent studies have explored alternative strategies to mitigate hallucination, including prompt engineering, retrieval-based augmentation, self-refinement through feedback, and prompt tuning.

2.3 Developing Novel Models

Some researchers focus on designing entirely new models or frameworks to mitigate hallucination. These approaches introduce architectural modifications or inference-time interventions rather than relying solely on fine-tuning. By altering the internal mechanisms of language generation, these methods aim to directly influence the production of factual and contextually aligned outputs.

2.3.1 Introducing New Decoding Strategies

Decoding-level strategies guide the text generation process to reduce hallucinations by influencing token selection. These methods encourage outputs that are more faithful, contextually relevant, and less likely to introduce unsupported information (Lango & Dusek, 2023). Three notable contribu-ˇ tions in this direction include Context-Aware Decoding (CAD), Decoding by Contrasting Layers (DoLa), and Inference-Time Intervention (ITI).

Context-Aware Decoding (CAD)

Shi et al. (2024) propose Context-Aware Decoding (CAD), a method that employs a contrastive output distribution to amplify differences between probabilities when the model is used with context versus without context. The approach overrides a model's parametric memory when it conflicts with external context, which is crucial for resolving knowledge contradictions. A key contribution of CAD is that it can be applied to pretrained language models without additional fine-tuning, making it computationally efficient and adaptable across tasks. Experimental results show that CAD significantly reduces hallucination rates by ensuring the model prioritizes reliable contextual information.

Limitation and Research Gap

While CAD is effective during inference, it does not modify the underlying model parameters, meaning hallucination reduction depends solely on decoding adjustments rather than learning more truthful generative behavior over time.

Decoding by Contrasting Layers (DoLa)

Chuang et al. (2023) introduce Decoding by Contrasting Layers (DoLa), a method that retrieves the next-token distribution by contrasting logit differences between early and late transformer layers. Their work demonstrates that factual knowledge is encoded at distinct layer depths, and by leveraging this representation more effectively, DoLa improves truthfulness without requiring retraining. The method consistently enhances performance on multiple-choice and open-ended benchmarks such as TruthfulQA, particularly across the LLaMA family of models.

Limitation and Research Gap

Although DoLa improves factual accuracy, it focuses only on inference-time correction and does not incorporate structured hallucination detection or counterfactual learning, limiting its ability to address hallucination at the training stage.

Inference-Time Intervention (ITI)

Li, Patel, Viegas, Pfister, and Wattenberg (2023) present Inference-Time Intervention (ITI), which´ adjusts model activations by identifying attention heads correlated with truthful reasoning and shifting activations along these directions during generation. This autoregressive intervention significantly boosts truthfulness on benchmarks like TruthfulQA, particularly in reasoning-intensive tasks.

Limitation and Research Gap

Despite strong results, ITI requires access to model internals such as attention head activations, making it impractical for closed-source or commercial models and limiting reproducibility in constrained environments.

Summary

Collectively, decoding-based interventions demonstrate that hallucination can be reduced by manipulating token selection and internal representations during inference. However, these methods operate post-hoc and do not modify the generative model's learning process or integrate explicit hallucination detection. This gap creates the need for an approach that combines classifier-based detection with counterfactually grounded correction; an issue addressed by the SGCT–HPM modular framework proposed in this study.

2.4 Utilization of Knowledge Graphs (KG)

Knowledge Graphs (KGs) are structured repositories that capture entities such as people, places, and objects along with their attributes and semantic relationships (Sun, Xu, Zha, Liu, & Dong, 2023). By encoding interconnected factual information, KGs enable machines to perform complex reasoning and evidence retrieval, supporting more reliable and interpretable decision-making. Due to their grounding capabilities, KGs have been widely adopted as an external mechanism for

mitigating hallucinations in language models, particularly where verifiability and factual grounding are essential (Bayat et al., 2023). The following studies demonstrate how KGs are applied within hallucination reduction strategies.

2.4.1 Reducing Hallucination in Open-Domain Dialogues (RHO)

Ji et al. (2022) propose RHO, a framework designed to address hallucination in open-domain dialogue generation by integrating structured entity and relation information from external knowledge graphs. The system employs both local and global knowledge-grounding mechanisms to align model outputs with verified information. Additionally, a conversational reasoning component reranks generated responses to enhance factual coherence, ensuring that responses remain grounded in relevant contextual knowledge. The results demonstrate that combining graph-based grounding with discourse-aware re-ranking significantly reduces hallucinations in dialogue generation.

2.4.2 Limitation and Research Gap

Although effective for correcting factual inconsistencies, RHO is heavily dependent on the completeness and availability of structured knowledge sources. When relevant entities or relations are absent from the graph, the model lacks fallback mechanisms, limiting scalability to open-world or rapidly changing domains.

2.4.3 Factual Error Detection and Correction with Evidence Retrieval from External Knowledge (FLEEK)

Bayat et al. (2023) introduce FLEEK, a model-agnostic framework for fact verification and correction that automatically extracts verifiable claims and retrieves supporting evidence from structured knowledge graphs and open web sources. The tool provides interpretable verification through generated questions and visual evidence indicators, enabling users to inspect and revise

potentially incorrect statements. The authors highlight strong potential for integrating automated evidence retrieval with human oversight, improving transparency and trust in generated content.

2.4.4 Limitation and Research Gap

While FLEEK demonstrates strong post-generation verification capabilities, it functions primarily as a correction interface rather than influencing generative behaviour directly. As such, hallucinations are managed reactively after text has been generated, rather than prevented during the generation process itself.

Summary

The reviewed studies illustrate that knowledge-grounding approaches improve factual alignment through structured evidence retrieval and contextual grounding. However, their success relies on external knowledge availability and they predominantly operate as post-hoc verification frameworks rather than modifying model training dynamics. These limitations highlight the need for approaches that internalize factual reasoning and reduce hallucination proactively. This motivates the exploration of a modular detection-and-correction framework such as SGCT–HPM, which integrates hallucination identification with counterfactual training to influence generative behaviour at the model level.

2.5 Introducing Faithfulness-Based Loss Functions

Faithfulness-based strategies focus on ensuring that generated outputs align with source information or verified ground-truth evidence. In this context, faithfulness refers to the extent to which a model can accurately represent provided information without introducing distortions, omissions, or unsupported claims (Chrysostomou & Aletras, 2021). The following studies illustrate loss-level interventions designed to reduce hallucinations during model training.

2.5.1 Text Hallucination Mitigating (THAM) Framework

Yoon, Yoon, Yoon, Kim, and Yoo (2022) propose the Text Hallucination Mitigating (THAM) framework for video-grounded dialogue generation, addressing the problem of surface-level copying where models reproduce text from input content without genuine semantic reasoning. THAM introduces an information-theoretic regularization mechanism that incorporates Text Hallucination Regularization (THR) loss, which leverages mutual information between a hallucination-aware language model and the primary response model. By minimizing THR loss, the framework suppresses hallucinated continuations at the feature-representation level, resulting in responses that exhibit stronger contextual alignment and higher semantic fidelity. Experimental results demonstrate improvements in dialogue relevance and grounded reasoning.

Limitation and Research Gap

Although THAM effectively reduces hallucinations in video-grounded dialogue tasks, its reliance on dual-model training and additional mutual-information computation introduces substantial computational overhead. The framework is also highly domain-specific and has not been extensively validated in open-ended text generation settings typical of large-scale LLMs.

2.5.2 Loss Weighting Method

Qiu, Embar, Cohen, and Han (2023) introduce mFACT, a metric designed to evaluate faithfulness in low-resource and multilingual summarization tasks. The method extends cross-lingual transfer learning by incorporating weighted training samples based on their faithfulness scores, thereby emphasizing reliable examples while down-weighting noisy or unfaithful samples. The authors report that although cross-lingual transfer improves summary fluency and coverage, it simultaneously increases hallucination frequency relative to monolingual systems. Their proposed loss-weighting technique mitigates this issue by guiding optimization toward more grounded outputs, yielding measurable improvements in factual accuracy.

Limitation and Research Gap

Despite its advantages, the loss-weighting method functions primarily within summarization tasks and lacks evaluation in broader generative contexts such as question answering or reasoning-based text generation. Additionally, the approach assumes the availability of faithfulness metrics, which limits its practicality for real-world deployment where reference datasets may be unavailable.

Summary

Faithfulness-based loss functions demonstrate that hallucinations can be reduced not only via output-level verification but also through direct optimization of model training objectives. However, the reviewed approaches remain constrained by task-specific domains, computational complexity, and limited generalization to open-ended generative tasks. These gaps highlight the need for scalable solutions that integrate hallucination detection with generative correction across diverse contexts. This motivates the development of the SGCT–HPM pipeline, which applies contrastive and unlikelihood-based objectives to proactively enforce factual generation while maintaining general applicability.

2.6 Supervised Fine-Tuning (SFT)

Supervised Fine-Tuning (SFT) plays a central role in aligning large language models (LLMs) to downstream tasks using labelled datasets. By exposing LLMs to curated examples, SFT adjusts model weights via gradient-based optimization to reduce the discrepancy between predicted and target outputs, thereby improving controllability and factual reliability (Sun et al., 2023). The effectiveness of SFT depends heavily on the quality and structure of training data (Touvron et al., 2023; Xu et al., 2023), and it has proven valuable for enhancing the adaptability of LLMs in novel task domains. Recent research explores how SFT can be used specifically to mitigate hallucinations.

2.6.1 Hallucination Augmented Recitations (HAR)

Koksal, Aksitov, and Chang (2023) introduce Hallucination-Augmented Recitations (HAR), which¨ reframes hallucinations as training signals rather than errors to avoid. HAR constructs counterfactual datasets generated from model hallucinations to strengthen attribution and improve grounding in open-book question answering tasks. Experiments on CF-TriviaQA show that HAR-trained models consistently outperform those trained on traditional factual datasets, even with smaller model sizes and reduced training resources.

Limitation and Research Gap

While HAR improves attribution accuracy, its dependence on open-book QA limits applicability to broader generative scenarios such as free-form dialogue or long-form reasoning. Additionally, HAR requires reliable scoring to identify meaningful hallucinations, which may be challenging without a robust classifier.

2.6.2 Refusal-Aware Instruction Tuning (R-Tuning)

Zhang et al. (2024) propose R-Tuning to instill refusal behaviours that prevent models from providing answers when uncertain or when prompts exceed their knowledge boundaries. By quantifying uncertainty and augmenting instructional datasets with refusal expressions, R-Tuning teaches models when to abstain. Evidence demonstrates improved safety alignment and reduced unsupported claims.

Limitation and Research Gap

Although effective for safety and abstention behaviours, R-Tuning focuses on refusal rather than proactive hallucination correction. It does not improve generative accuracy when the model chooses to respond.

2.6.3 Think While Effectively Articulating Knowledge (TWEAK)

Qiu et al. (2023) introduce TWEAK, a decoding method that evaluates candidate sequences using a Hypothesis Verification Model (HVM), ranking outputs based on factual support rather than likelihood alone. Unlike traditional decoding (e.g., beam search), TWEAK prioritizes factual consistency and integrates easily into existing models without requiring retraining. The authors additionally release FATE, a dataset that aligns facts with factual and counterfactual statements at a fine-grained level.

Limitation and Research Gap

TWEAK operates strictly at inference time and provides no mechanism for updating model parameters to reduce hallucination tendencies long-term.

2.6.4 Fine-Tuning Language Models for Factuality

Tian, Mitchell, Yao, Manning, and Finn (2023) fine-tune LLaMA-2 using Direct Preference Optimization (DPO) guided by automated fact-checking signals, reducing factual error rates in longform medical and biographical text without human annotation. The study demonstrates that factual preference learning is both scalable and cost-efficient.

Limitation and Research Gap

Despite strong improvements, the approach relies on reference-based evaluation pipelines external to the model, which may not generalize to open-ended generative tasks or unsupported knowledge domains.

2.6.5 Knowledge Injection and Teacher-Student Approaches

Elaraby et al. (2023) explore hallucination mitigation in smaller open-source models such as BLOOM 7B by combining knowledge injection and teacher–student learning. Their HALOCHECK framework evaluates hallucination severity using entailment-based scoring and fine-tunes weaker

models using curated knowledge and teacher-generated examples. This approach reduces hallucinations while preserving computational efficiency.

Limitation and Research Gap

Knowledge injection requires domain-specific curated datasets, and teacher–student reliance assumes access to significantly stronger proprietary models such as GPT-4, limiting reproducibility.

2.6.6 BEINFO

Razumovskaia et al. (2024) propose BEINFO, a behavioral fine-tuning method for aligning dialogue models by training them on curated conversations augmented with random distractor information. The goal is to enforce strict reliance on grounded sources rather than parametric memory.

Experimental results show improvements in factual consistency and reliability.

Limitation and Research Gap

BEINFO requires extensive annotated dialogue datasets and remains limited to conversational environments, lacking validation in broader text generation contexts.

Summary

SFT-based approaches demonstrate substantial potential for hallucination mitigation by modifying model behaviour through labelled examples or preference-guided optimization. However, these methods are constrained by high data demands, domain specificity, and limited generalization beyond narrow tasks. Moreover, most solutions address hallucination through reactive filtering or refusal rather than proactive generative correction. These limitations highlight the need for a modular approach that unifies hallucination detection and counterfactual correction within the training process which is an objective pursued through the SGCT–HPM framework introduced in this study.

2.7 Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) enhances large language models by integrating external, authoritative knowledge sources into the generation process. Instead of relying solely on internal parametric memory, RAG retrieves relevant and up-to-date evidence to support output generation, thereby improving factual reliability and reducing hallucination risk (Kang, Ni, & Yao, 2023). Grounding responses in retrieved information strengthens verifiability and reduces dependency on static pretraining corpora, addressing issues of temporal drift and limited world knowledge.

2.7.1 Before Generation

These approaches retrieve knowledge prior to model inference, using the retrieved information to condition the generation process.

LLM-Augmenter

Peng et al. (2023) present LLM-Augmenter, a plug-and-play framework that enhances black-box models such as GPT-3.5 by grounding generation in externally retrieved knowledge. The system retrieves evidence, constructs reasoning chains, and consolidates them into prompts that guide generation. Output candidates are then verified, and hallucinations trigger iterative prompt refinement until a validated response is produced. This framework demonstrates that retrieval and automated verification can be combined to significantly improve factual reliability without modifying model parameters.

Limitation and Research Gap:

LLM-Augmenter relies on recursive querying and verification cycles, which may introduce latency and computational overhead. Additionally, its effectiveness depends on access to high-quality retrieval sources and verification signals.

FreshPrompt

Vu et al. (2023) introduce FreshPrompt, a retrieval-based few-shot prompting method designed for dynamic question answering. Using FreshQA, a benchmark evaluating temporal reasoning and false-premise questions, the study shows that many retrieval-based systems struggle with temporal drift and hallucination. FreshPrompt integrates search-engine retrieval into prompts, enabling access to current information and outperforming baseline approaches across multiple tasks and commercial systems.

Limitation and Research Gap

FreshPrompt improves factual correctness but does not integrate structured hallucination detection or correction mechanisms; it also remains highly dependent on retrieval quality and evidence ordering.

2.7.2 During Generation

These approaches retrieve knowledge dynamically during sentence-by-sentence generation.

Knowledge Retrieval During Generation

Varshney, Yao, Zhang, Chen, and Yu (2023) propose a retrieval-based strategy that detects hallucinations in real time using logit-based signals to identify high-risk tokens. When hallucinations are detected, retrieved evidence is used to validate or correct statements. The study highlights the compounding nature of hallucinations, demonstrating that early correction reduces cascading

factual drift.

Limitation and Research Gap

The method assumes access to model logits, limiting applicability to black-box commercial APIs. Additionally, the system focuses on correction rather than altering model behavior during training.

Real-time Verification and Rectification (EVER)

(Kang et al., 2023) introduce EVER, a real-time verification framework consisting of generation, validation, and rectification stages that address hallucinations stepwise during inference. EVER improves reliability across tasks such as biography generation and multi-hop reasoning while mitigating the snowballing effect where early hallucinations propagate through the generated response.

Limitation and Research Gap

EVER operates at inference time and does not update the underlying model to prevent hallucinations in future generations.

2.7.3 After Generation

These methods retrieve knowledge after the text is generated to validate and correct outputs.

Retrofit Attribution using Research and Revision (RARR)

Gao et al. (2022) propose RARR, a research-and-revision framework that improves attribution by aligning generated text with retrieved supporting evidence. RARR formalizes editing for attribution and introduces evaluation metrics for revision quality. Results show improved factual consistency and preservation of stylistic fluency.

Limitation and Research Gap

The system functions as a post-hoc editor and assumes access to revision resources, which may limit real-time deployment.

High Entropy Word Spotting and Replacement

(Rawte et al., 2023) propose a system that detects high-entropy words—tokens correlated with hallucination vulnerability and substitutes them using alternative models. Experiments show that distilroberta-base effectively replaces high-risk tokens and reduces hallucinations, particularly in acronym ambiguity and entity conflicts.

Limitation and Research Gap

The approach is constrained by model compatibility and primarily addresses token-level hallucination rather than larger structural inaccuracies.

2.7.4 End-to-End Retrieval-Augmented Generation (RAG)

(Lewis et al., 2020) introduce the end-to-end RAG framework combining Dense Passage Retrieval (DPR) with a BART generator. Unlike modular systems, RAG jointly trains retrieval and generation such that retrieved documents influence token-level production. This architecture demonstrates significant improvements on knowledge-intensive tasks and shows the effectiveness of combining parametric and non-parametric memory.

Limitation and Research Gap

Although powerful, end-to-end RAG is computationally expensive to train and depends on large, well-indexed retrieval corpora. It does not integrate hallucination detection or counterfactual learning and rarely generalizes well to domains with sparse evidence.

Summary

RAG-based approaches highlight the importance of grounding model responses in external evidence and demonstrate strong potential for improving factual reliability across multiple domains. However, the reviewed techniques primarily operate as retrieval-and-correction mechanisms and do not address hallucination proactively within model training. These limitations

reveal a gap for methods that integrate structured hallucination detection with generative correction to influence underlying model behavior rather than relying solely on evidence retrieval. This motivates the development of the SGCT–HPM framework proposed in this study.

2.8 Summary and Research Gap

The review of existing literature reveals significant progress in mitigating hallucinations in large language models through a range of strategies including decoding-based interventions, knowledgegrounded retrieval mechanisms, faithfulness-oriented loss functions, and supervised fine-tuning approaches. These methods collectively demonstrate that hallucinations can be reduced by improving factual grounding, enhancing reasoning alignment, and incorporating external knowledge or structured preference feedback. However, the evidence also shows that most existing solutions operate either at inference time or through post-hoc verification, relying heavily on retrieval or external sources to filter or correct hallucinated outputs. As a result, they do not fundamentally modify the internal generative behaviour of LLMs or proactively prevent hallucinations at the model-training level. Additionally, current approaches rarely integrate detection and correction within a unified framework, and many remain domain-specific, computationally expensive, or dependent on resources unavailable in real-world settings.

These limitations reveal a clear research gap: the need for a scalable and model-agnostic approach that can both identify hallucinations and correct them during generation, rather than merely filtering them after the fact. In response to this gap, the present study proposes a modular hallucination mitigation framework that combining Hallucination Potential Minimization (HPM) as a detection mechanism with Self-Generated Counterfactual Training (SGCT) as a correction strategy to proactively enforce factual consistency. By integrating classifier-guided detection with counterfactual learning within a unified pipeline, this study contributes a practical, reproducible, and empirically validated solution to improve the reliability of LLM outputs. The following chapter details the methodology used to design, implement, and evaluate this framework.

Chapter 3

Methodology

3.1 Introduction

This chapter outlines the methodological framework adopted for investigating hallucination mitigation in large language models (LLMs) through a combined approach involving Hallucination Potential Minimization (HPM) and Self-Generated Counterfactual Training (SGCT). The design integrates dataset preparation, model architecture, fine-tuning strategies, and evaluation metrics, all within a supervised learning context. The choice of methodology is informed by empirical precedents in the literature and tailored to address the dual objectives of hallucination detection and correction.

3.2 Research Design

The study adopts an experimental research design rooted in supervised fine-tuning of transformerbased language models. The research is structured in two phases: the first involves training a hallucination detector (HPM), and the second focuses on the correction of hallucinated responses using SGCT. The dual-phase structure enables modular development and evaluation of the detection and correction components.

3.3 Dataset Preparation

The training data comprises a merged and manually labeled combination of the FEVER and TruthfulQA datasets. Each sample is structured in JSON format with four fields: question, answer, evidence, and label. Labels were reversed for alignment with the research framework, with 0 denoting hallucinated outputs and 1 representing factual responses. A balanced stratified

sampling strategy was applied to the FEVER dataset, selecting an equal number of 'SUPPORTS' and 'REFUTES' claims (1,500 each), while discarding the 'NOT ENOUGH INFO' class. This ensured a balanced binary dataset of 3,000 claims. To address the limited size of the original TruthfulQA dataset (817 questions), an augmentation procedure was employed. Each question was paraphrased using a transformer-based text-to-text generation model (T5 paraphraser). For every item, up to three semantically equivalent paraphrases were generated, paired with the original ground-truth answer and label. This process effectively expanded the dataset to 1,634 entries. During augmentation, only items that produced valid paraphrases were retained, and duplicates or malformed outputs were discarded. This approach can be viewed as a balanced data augmentation strategy that preserves the factual integrity of the original annotations while increasing sample diversity. Stratified sampling was applined on the combined dataset to split the dataset into training (70 %), validation (15%), and testing (15 %) subsets to ensure class balance and avoid dataset shift, hence ensuring robust training and generalisation analysis.

Figure 3.1: Snapshot of the dataset structure showing JSON fields (question, answer, evidence, label, and source) used for training and evaluation of the HPM classifier and SGCT fine-tuning.

3.4 Baseline Model Configuration

To provide a fair comparison for evaluating the performance of the proposed SGCT-HPM framework, the GPT-Neo model was selected as a supervised fine-tuning baseline. GPT-Neo was finetuned on the combined FEVER and TruthfulQA training split using standard cross-entropy loss

without SGCT, counterfactual sampling, or classifier-judge loss functions. The motivation for including this baseline model is to isolate the contribution of the SGCT-HPM process by comparing the proposed model against a strong fine-tuning baseline trained under identical conditions. Both the SGCT final output and the GPT-Neo baseline used comparable hyperparameters, including a maximum of five epochs with early stopping, AdamW optimisation, and identical decoding and scoring pipelines. This ensures that performance differences are attributable to the SGCT-HPM methodology.

3.5 Hallucination Minimization Potential

The Hallucination Potential Minimization (HPM) phase constituted the first stage of the experimental pipeline, with the primary objective of building a robust binary classification system capable of distinguishing between factual and hallucinatory content in language model outputs. This system served as the factuality evaluation backbone in subsequent stages of the study, ensuring that generative models could be reliably assessed and guided towards more accurate responses.

HPM was conceptualised as a supervised fine-tuning task in which a transformer-based encoder was trained to predict the factuality of a generated answer when presented alongside its originating question and supporting evidence. The choice of a binary classification framework — labelling responses as either factual (label = 1) or hallucinated (label = 0) — was motivated by the need for clarity and interpretability in hallucination detection, as well as compatibility with a wide range of downstream fact-checking applications.

The central design principle guiding HPM development was the prioritisation of recall over precision, reflecting the view that missing an actual hallucination presents a greater risk than occasionally misclassifying a factual statement as incorrect. This high-recall orientation ensures that the detector functions conservatively, flagging potentially inaccurate outputs for further review rather than allowing them to pass unchallenged.

3.5.1 Dataset and Preprocessing

The training corpus was derived from the FEVER dataset (Thorne, Vlachos, Christodoulopoulos, & Mittal, 2018) and TruthfulQA datasets (Lin, Hilton, & Evans, 2021), reformatted into structured JSON instances with four fields: question, answer, evidence, and label. The label was defined as 1 for factual responses and 0 for hallucinations.

During preprocessing, all fields were normalised as strings, with empty strings inserted where evidence was not available. Instances with missing or malformed inputs were removed to maintain data integrity. Class balance was monitored to ensure that both factual and hallucinatory examples were adequately represented. To guarantee reproducibility and prevent data leakage, the dataset was split by stratified sampling into training (70%), validation (15%), and test (15%) sets using a fixed random seed (randomstate=42).

3.5.2 Model Architecture

The classification model selected for HPM was DistilBERT (distilbert-base-uncased) (Sanh, Debut, Chaumond, & Wolf, 2019), a transformer-based encoder designed for efficiency while retaining strong representational capacity. The model was initialised with weights pretrained on large-scale language understanding tasks and then fine-tuned for binary sequence classification.

Input formatting followed a paired sequence approach: the question and evidence were concatenated as the left-hand sequence (‘‘Q: {question} EVIDENCE: {evidence}’’), while the candidate answer was treated as the right-hand sequence. The DistilBERT tokenizer was used to encode the pair jointly with a maximum sequence length of 512 tokens, truncating longer inputs. Dynamic padding and attention masks ensured that padded tokens were excluded from contributing to the model loss.

3.5.3 Training Configuration

Training was implemented using the Hugging Face Transformers library. The model was trained with a batch size of 8, and the AdamW optimiser (Loshchilov & Hutter, 2017) was employed with a learning rate (Vaswani et al., 2017) of 2×10^{-5} and weight decay of 0.01. A linear learning rate scheduler with 10% warm-up steps was used to stabilise training during the early epochs. Gradient clipping with a maximum norm of 1.0 was applied to prevent exploding gradients. The model was trained for a maximum of three epochs, with early stopping triggered if validation F1 did not improve for three consecutive checks.

The binary cross-entropy loss was used as the objective function:

$$L_{HPM} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i)],$$

where $y_i \in \{0,1\}$ is the true label and $\hat{y}_i$ is the predicted probability of being factual.

3.5.4 Early Stopping and Checkpointing

To mitigate overfitting, validation F1 score was monitored after each epoch. Whenever performance improved, model checkpoints were saved in Hugging Face's savepretrained format. If validation F1 failed to improve for three consecutive epochs, training was halted early. Both the tokenizer and model weights were stored to ensure reproducibility.

3.5.5 Monitoring and Visualization

The training process included real-time logging of loss and metrics for both training and validation. At the conclusion of training, consolidated plots were generated showing the trajectories of accuracy, precision, recall, F1 score, and AUC across epochs. These visualisations were used to confirm training stability, detect overfitting, and ensure that recall was consistently prioritised.

3.5.6 Evaluation Protocol

Evaluation of the final classifier was conducted on the held-out test set. Both threshold-dependent and threshold-independent metrics were employed. Accuracy measured the overall correctness of predictions; Precision quantified the proportion of predicted factual statements that were actually factual; Recall measured the proportion of factual statements correctly identified as such; and the F1 score provided the harmonic mean of precision and recall (Hao & Ho, 2019). The Area Under the Receiver Operating Characteristic Curve (AUC) was also computed to capture the model's discriminative capacity across all thresholds.

Given the safety-oriented design philosophy of this work, Recall was emphasised as the primary performance metric. A conservative factuality detector that occasionally over-flags factual outputs is preferable to one that allows hallucinations to go undetected.

3.5.7 HPM Hyperparameters

Table 3.1: Summary of HPM (Hallucination Potential Minimization) Training Hyperparameters and Settings

Category	Setting
Model Backbone	DistilBERT (distilbert-base-uncased), pre-trained on general language understanding tasks.
Tokenizer	DistilBERT tokenizer; paired encoding of ‘‘Q: {question} EVIDENCE: {evidence}’’ and answer.
Sequence Length	Maximum of 512 tokens (truncated if exceeded).
Batch Size	8
Learning Rate	2×10^{-5}
Optimizer	AdamW with weight decay 0.01
Learning Schedule	Linear scheduler with 10% warm-up steps
Gradient Clipping	Maximum norm = 1.0
Epochs	Up to 3 (with early stopping)
Early Stopping	Patience of 3 validation checks, monitored on F1 score
Loss Function	Binary cross-entropy loss

Metrics	Accuracy, Precision, Recall, F1 score, ROC-AUC
Splitting Strategy	Stratified sampling: 70% train, 15% validation, 15% test
Random Seed	randomstate=42 for reproducibility
Checkpointing	Best model (highest validation F1) saved in Face Hugging savepretrained format

3.6 Self Generated Counterfactual Training

This study addresses the task of factuality-aware text generation, in which a generative language model (LM) is conditioned on a question q to produce a factual answer $\hat{a}$. To improve factual consistency, we propose a pipeline combining two core components:

1. A causal language model trained with Self-Generated Counterfactual Training (SGCT), which encourages the model to generate factual responses and penalize hallucinated outputs.
2. A factuality classifier, referred to as the Hallucination Potential Minimization (HPM), which acts both as a weak oracle during training and an evaluation proxy at test time.

The SGCT training paradigm shapes the generator's output distribution by integrating standard likelihood on factual samples, unlikelihood loss (Welleck et al., 2019) on hallucinated continuations, and a representation-level contrastive loss to separate factual and hallucinated embeddings.

3.6.1 Dataset and Preprocessing

Data Format and Fields

We employ a JSON-formatted dataset comprising records with the fields: {question, answer, evidence, label, source}. Only the question and answer fields are required during model training, although auxiliary fields are retained for analysis and optional calibration.

Input Formatting

Two distinct input formats are used:

- Generator (SGCT): "Question: {q}\nAnswer: {a}"
- Factuality Classifier (HPM): tokenized as a pair ("Q: {q}", a)

Table 3.2: Comparison of the base dataset and the contrastive pairs used in SGCT training.

Aspect	Base Dataset (FEVER + TruthfulQA)	Contrastive Pairs (for SGCT)
Source	External benchmark datasets (curated and annotated by prior work).	Generated internally using GPT2 outputs, filtered by HPM.
Structure	{question, answer, evidence, label}	{question, factual answer, hallucinated answer}
Labels	Provided directly (1 = factual, 0 = hallucination).	Assigned via HPM scoring (≥ τ_f factual, ≥ τ_h hallucinated).
Purpose	Provides ground-truth factual anchors for training and evaluation.	Provides positive vs. negative training pairs for contrastive fine-tuning.
Role in Pipeline	Used to fine-tune HPM and serve as factual references.	Used to train GPT-2 under SGCT losses (likelihood, unlikelihood, contrastive).

All inputs are truncated to a maximum of 256 tokens, balancing memory constraints and input preservation. During decoding, only the first line following "Answer:" is extracted, and completions with fewer than five words are filtered to remove trivial or degenerate outputs.

Train/Validation/Test Split

Data are split in a 70/15/15 ratio by *question*, using traintestsplit(randomstate=42)
to ensure reproducibility. Stratification is avoided due to inconsistencies in label distribution across data sources. By splitting on the question field, we minimize semantic leakage between training and evaluation sets.

3.6.2 Model Architectures

Generator: GPT-2 (SGCT Backbone)

We adopt the GPT-2 (small) (Radford et al., 2019) architecture as the causal LM due to its computational efficiency and compatibility with token-level likelihood and hidden state extraction.

Tokenization is performed using the GPT-2 tokenizer, with the eostoken used for padding.

Training is conducted on CUDA-enabled devices where available.

Factuality Judge: DistilBERT (HPM)

The HPM is instantiated as a DistilBERTForSequenceClassification model fine-tuned for binary factuality classification. Input pairs are of the form ("Q: {q}", a), and the model outputs a probability score $p_{\text{fact}} = P(y = 1 \mid q,a)$, which is interpreted as the likelihood of factuality. This score is used for both filtering training candidates and evaluating generated outputs.

3.6.3 Candidate Construction and Filtering

Candidate Generation

For each question, we generate up to eight candidate answers using a mixture of decoding strategies:

Nucleus sampling with temperatures $\{0.7, 0.9, 1.1\}$ and $top\ p = 0.9$, Top-k sampling with $k \in \{40, 60\}$ and Beam search with num beams = 3. All decoding uses repetition penalties (norepeatngramsize = 3, repetitionpenalty = 1.1) and a token cap (maxnew_tokens = 96) to ensure fluency and diversity.

Perturbation-Based Hard Negatives

One candidate per question is augmented with lightweight counterfactual edits:

- Negation flips (e.g., "is" → "is not")
- Numerical alterations (+1 offset to the first detected number)
- Named entity swaps (e.g., *Paris* ↔ *London*)

These serve as hard negatives grammatical but factually incorrect variants to enhance the counterfactual signal without relying on external retrieval.

HPM-Based Labeling and Selection

Each candidate answer is scored by the HPM. Thresholds are applied to assign labels:

$p_{\text{fact}} \geq \tau_f = 0.7$: labeled factual $p_{\text{fact}} \leq \tau_h = 0.3$: labeled hallucinated Otherwise: discarded as uncertain.

From the filtered set, we retain:

- Up to 2 factual responses
- Up to 3 hallucinated responses
- The dataset ground-truth answer (always included as factual)

A boolean mask in the training batch disables the contrastive and unlikelihood losses if no hallucinated candidates are present, thereby preventing instability from empty strings.

3.6.4 Training Objectives

The overall loss function is a weighted sum of three components:

$L = \alpha L_{LH} + \beta L_{UL} + \gamma L_{CT}$

Initial weights: $\alpha = 1.0$, $\beta = 0.1$, $\gamma = 0.2$

Likelihood Loss (Factual Cross-Entropy)

$$\mathcal{L}_{\text{LH}} = -\frac{1}{T}\sum_{t=1}^{T} \log p_\theta(x_t \mid x_{<t})$$

This standard language modeling loss is applied to factual sequences to guide fluent and accurate text generation.

Token Unlikelihood Loss (Hallucinations)

$$L_{UL} = -\frac{1}{|S|} \sum_{t \in S} \log(1 - p_\theta(\tilde{x}_t \mid \tilde{x}_{<t}))$$

where S is the set of tokens with $p > 0.05$. This loss selectively penalizes high-confidence tokens in hallucinated sequences, discouraging the model from confidently generating incorrect content.

Cosine-Margin Contrastive Loss

$$L_{CT} = \max(0, \langle f, h \rangle - m)$$

where f, h are ℓ_2-normalized final hidden states of factual and hallucinated outputs, and $m = 0.3$ is the cosine margin. This loss enforces embedding separation between correct and incorrect outputs.

Optimization and Regularization

- Optimizer: AdamW, learning rate 5×10^{-5}, weight decay 0.01

- Scheduler: Linear decay with 10% warm-up
- Batch Size: 4 (suitable for consumer GPUs)
- Epochs: Up to 5 with early stopping (patience = 3), monitored on validation F1
- Gradient Clipping: Max norm = 1.0

This configuration balances learning stability with responsiveness to noisy contrastive signals.

Curriculum Learning Strategy

After each epoch, the model is evaluated on the validation set using the HPM. If recall < 0.85, loss weights are adaptively reweighted:

$\beta \leftarrow \min(\beta + 0.05, 0.6)$ $\gamma \leftarrow \min(\gamma + 0.05, 0.6)$ $\alpha \leftarrow \max(\alpha - 0.05, 0.5)$

This curriculum prioritizes recall, ensuring the model does not underpredict factual answers—a critical constraint in safety-sensitive applications.

3.6.5 Validation and Evaluation Protocol

Generation Settings

On validation and test sets, responses are generated with:

- $top_p = 0.92$, $top_k = 50$, $temperature = 0.8$
- $no\ repeat\ ngram\ size = 3$, $repetition\ penalty = 1.1$
- $max\ new\ tokens = 96$

The post-decoding answer is extracted from the "Answer:" section.

Proxy Labeling via HPM

Since ground-truth factuality annotations may be unavailable or incomplete, we use HPM as a proxy evaluator:

$y_{\text{true}}(q) = 1[p_{\text{HPM}}(q, a_{\text{GT}}) \geq \tau]$ $y_{\text{pred}}(q) = 1[p_{\text{HPM}}(q, \hat{a}) \geq \tau]$

The default threshold is $\tau = 0.7$, optionally calibrated on validation via a sweep maximizing F1. We visualize F1–τ curve for interpretability.

3.6.6 Evaluation Metrics

To assess the factual consistency of generated answers, we adopt a suite of standard classification metrics computed using the scikit-learn library. These metrics evaluate the alignment between the model's outputs and the factuality labels assigned by the HPM classifier.

- Accuracy: This measures the proportion of correct predictions (both factual and hallucinated) over the total number of predictions. While it offers a general overview of model performance, it may be misleading in the presence of class imbalance—e.g., when most answers are factual or hallucinated—making it insufficient on its own.
- Precision: Precision quantifies the proportion of answers predicted as factual that are actually factual. It is critical in settings where false positives (i.e., hallucinated outputs mistakenly labeled as factual) must be minimized. High precision indicates that the model is conservative and only labels outputs as factual when it is confident.
- Recall: Recall measures the proportion of truly factual answers that are correctly identified. In the context of factuality-aware generation, high recall is essential to ensure that valid factual answers are not mistakenly discarded or down-weighted. It is particularly important when recall is prioritized as a safety objective.
- F1 Score: The F1 score is the harmonic mean of precision and recall. It provides a balanced view of the model's ability to avoid both false positives and false negatives. This is especially

relevant when tuning thresholds or performing model selection under trade-offs between overgeneration and conservatism.

Throughout training, we log per-epoch metrics including loss, precision, recall, and F1 score. These are plotted over time to monitor convergence behavior, detect underfitting or overfitting, and diagnose performance trade-offs. F1 score is used as the principal metric for early stopping and model selection due to its robustness to class imbalance and its alignment with the factuality goal of the task.

Data Partitioning. The dataset was first cleaned by removing incomplete entries and then split by question into training (70%), validation (15%), and test (15%) subsets using a fixed random seed to ensure reproducibility. Splitting by question prevents near-duplicate leakage across splits. The Hallucination Potential Minimization (HPM, a DistilBERT classifier) is frozen throughout all experiments and serves as a consistent factuality judge for the SGCT training process.

Proxy Ground Truth. Since gold factuality annotations are not always available, we construct proxy labels using HPM itself. For each test question q with ground-truth answer a_{GT}, we define:

$$y_{\text{true}}(q) = 1\{p_{\text{HPM}}(q, a_{GT}) \geq \tau\},$$

where τ is the factuality threshold (set to $\tau = 0.70$, or tuned once on validation). This ensures that both pre- and post-training are judged relative to the same labeling function.

Unified Generation and Judging. For any generator G (either base GPT-2 or SGCT fine-tuned), we generate one answer $\hat{a}$ per question using identical decoding parameters: top-p = 0.92, top$k$ = 50, temperature = 0.8, maximum of 96 new tokens, with norepeatngramsize=3 and a repetition penalty of 1.1. The prompt format is:

‘‘Question: {q}\nAnswer:’’

The generated answer is extracted as the first line after “Answer:”. Each generated answer is then judged by HPM to produce:

$$y_{\text{pred}}(q) = 1\{p_{\text{HPM}}(q, \hat{a}) \geq \tau\}.$$

Metrics. We report Accuracy, Precision, Recall, and F1-score (binary, with “factual” as the positive class). These metrics are implemented using scikit-learn, and validation loss is defined as $1-\text{F1}$ to directly optimize factuality. During training, we log per-epoch curves for training loss, validation loss, precision, recall, and F1.

Pre-Evaluation. Before SGCT training, we evaluate the frozen base GPT-2 model on the test set with the above pipeline, yielding the *PRE* baseline metrics.

SGCT Training (Context). SGCT training is performed only on the training split. Contrastive pairs are constructed by generating diverse candidate answers, filtering them with HPM, and balancing up to two factual and three hallucinated pairs per question, always anchored on the dataset ground-truth answer. The generator is then trained with the composite loss:

$$\mathrm{L} = \alpha \mathrm{L}_{LH} + \beta \mathrm{L}_{UL} + \gamma \mathrm{L}_{CT},$$

where L_{LH} is the likelihood loss on factual continuations, L_{UL} is the token-level unlikelihood loss applied only to high-probability hallucinated tokens, and L_{CT} is a cosine-margin contrastive loss encouraging separation between factual and hallucinated representations. Early stopping is based on the best validation F1 with patience set to 3 epochs.

Post-evaluation. After training, the best SGCT-HPM checkpoint is reloaded and evaluated on the held-out test set using the same generation and judging procedure applied to the baseline models. The HPM factuality classifier, threshold values, decoding configuration, and test questions remain unchanged across all evaluations. By keeping these elements constant,

differences in performance between the SGCT-HPM model and the baseline models can be attributed to the SGCT training process rather than changes in evaluation conditions or external factors.

Reporting. Results are presented in a comparison table that evaluates the GPT-2 base model, the fine-tuned GPT-Neo baseline, and the final SGCT-HPM model on Accuracy, Precision, Recall, and F1 Score using the same held-out test set. Additional plots, such as Precision versus Recall or F1 Score as a function of the factuality threshold, may be included to illustrate how different operating points influence performance. This consistent reporting procedure ensures that the impact of the SGCT training process is measured in a fair, transparent, and reproducible manner.

3.6.7 Implementation and Reproducibility Notes

To facilitate reproducibility and extensibility of the Self-Generated Counterfactual Training (SGCT) pipeline, a few implementation aspects were controlled for and made explicit:

- Tokenization and Padding. The GPT-2 tokenizer does not provide a native padding token. In our implementation, the eostoken (end-of-sequence marker) was assigned as the padding token, and its integer ID was assigned as the padtokenid in the GPT-2 model configuration. Attention masks are applied during training and evaluation to ensure that padded positions do not contribute to the loss computation. This design avoids indexing errors and ensures that batch collation across variable-length inputs is stable.

- Batch Collation with Hallucination Mask. During contrastive pair construction, some questions may yield no valid hallucinated candidates (e.g., if the HPM scores all candidates as factual or uncertain). To prevent spurious losses in such cases, the batch collation step produces an explicit boolean flag hashallucinated. This mask is checked inside the training step, and the unlikelihood and contrastive components of the loss are skipped whenever the hallucinated input is empty. This avoids artificially penalizing the model on padding-only

sequences and ensures that losses are only applied when valid counterfactual examples exist.

- Answer Length Gating. To prevent trivial completions or degenerate decoding fragments from contaminating the candidate pool, generated answers with fewer than five tokens are filtered out. This heuristic ensures that the factuality classifier (HPM) and the generator's own loss functions operate only on semantically meaningful content, rather than on accidental short strings (e.g., single words, punctuation).

- Deterministic Data Splits. All train/validation/test partitions are created with random_state of 42 to guarantee reproducibility of the data splits. Although the decoding process during candidate generation uses stochastic sampling (top-p, top-k, and temperature), full determinism can be achieved by additionally fixing the global random seeds for NumPy and PyTorch. In our experiments, reproducibility at the data split level was prioritized, while generation randomness was retained to preserve candidate diversity.

- Threshold Calibration. The HPM decision thresholds (τ_f for factual and τ_h for hallucinated) are configurable hyperparameters. In the default setting, we use $\tau_f = 0.7$ and $\tau_h = 0.3$. Optionally, a threshold sweep is performed on the validation set to calibrate the operating point that maximizes F1. Once selected, this threshold is held fixed for both preand post-SGCT evaluations on the test set to ensure comparability.

- Model Checkpointing and Early Stopping. During training, the best-performing model checkpoint is identified according to validation F1. Early stopping is applied with a patience of three epochs to prevent overfitting. Both the tokenizer and the model weights are saved to disk in Hugging Face's savepretrained format, ensuring that the post-training evaluation uses exactly the same parameters.

- Logging and Visualization. Training logs include per-epoch decomposition of the composite loss into its three components (likelihood, unlikelihood, contrastive) as well as validation accuracy, precision, recall, and F1. These are visualized as convergence curves, enabling inspection of trade-offs between recall and precision during curriculum adaptation.

These design decisions ensure that SGCT fine-tuning is both reproducible and robust, while also guarding against degenerate training signals (e.g., empty hallucinations or trivial short answers). They also provide a transparent framework for future researchers to replicate or extend the experimental setup under identical conditions.

3.7 Summary of Methodological Justifications

The design of the SGCT–HPM pipeline was guided by a set of methodological choices that balance practicality, robustness, and theoretical grounding:

HPM-based candidate filtering.

Instead of relying on costly manual annotations or heavyweight retrieval-based supervision, we employ the Hallucination Potential Minimization (HPM) as an automatic filter. This allows us to rapidly classify generated candidates into factual, hallucinated, or uncertain sets. Although the HPM is itself imperfect, this approach provides a scalable and low-cost alternative that aligns the generator with a factuality judge, enabling large amounts of contrastive data to be constructed without human intervention.

Token-level unlikelihood.

We incorporate an unlikelihood loss that selectively penalizes tokens which the model assigns high probability to in hallucinated sequences. This prevents the generator from overconfidently producing false continuations, while at the same time avoiding unnecessary penalties on low-

content function words (e.g., punctuation, articles). By focusing on high-probability errors, the model is encouraged to unlearn misleading continuations without degrading fluency.

Cosine-margin contrastive loss.

In practice, datasets may not provide multiple factual paraphrases for every question, which limits the use of classical positive-pair contrastive methods. We therefore adopt a cosine-margin "pushaway" loss that only requires factual–hallucinated pairs. This loss forces the latent representations of hallucinated answers to diverge from factual ones whenever their cosine similarity exceeds a fixed margin. The method is computationally simple, robust to batch size, and effective even in the absence of positive factual pairs.

Curriculum reweighting.

Because factual recall is safety-critical (i.e., missing factual content is often more harmful than including uncertain content), we adapt the loss weights dynamically. When validation recall falls below 0.85, the contribution of the unlikelihood and contrastive components is increased while the factual likelihood weight is reduced. This curriculum ensures that the model places greater emphasis on suppressing hallucinations whenever it demonstrates weakness in factual detection.

Threshold calibration and PR analysis.

Fixed thresholds can obscure the precision–recall trade-off inherent in factuality judgments. We therefore sweep across multiple decision thresholds τ on the validation set, plotting both Precision– Recall (PR) curves and F1-versus-threshold curves. This analysis reveals the operating regions where factuality detection is most balanced and allows us to select a threshold that maximizes validation F1, which is then held constant for test evaluation.

Evaluation design.

To isolate the effect of the SGCT fine-tuning procedure, the model is evaluated against two baselines: the pretrained GPT-2 base model and a supervised fine-tuned GPT-Neo model. All models are assessed using the same HPM factuality classifier, the same decoding settings, and the same held-out test split. By keeping these factors constant, differences in performance can be attributed to the SGCT objective rather than differences in model size, capacity, or evaluation conditions. This comparative design strengthens the validity of the empirical findings and provides a fair assessment of the contribution of the SGCT-HPM pipeline.

3.8 Limitations

While the SGCT-HPM framework demonstrates meaningful improvements, several limitations remain that motivate future research:

- Judge-model coupling. Because the HPM is used both to filter training candidates and to evaluate generated answers, there is a risk of overfitting to the biases of a single classifier. The generator may learn to "game" the HPM's decision boundary rather than aligning with ground-truth factuality. In future work, this coupling can be mitigated by introducing multiple heterogeneous judges or incorporating human-annotated factuality labels.

- Contrastive inefficiency. Contrastive learning (Khosla et al., 2020) typically benefits from large batch sizes, which allow more diverse negative examples. Due to GPU memory constraints, our batches are relatively small, which underutilizes the contrastive signal. Techniques such as gradient accumulation, memory banks, or parameter-efficient fine-tuning (e.g., LoRA) could be explored to scale up the effective batch size without prohibitive computational cost.

3.9 Hyperparameter Summary

Table 3.3: Summary of SGCT–HPM Implementation Hyperparameters and Settings

Category	Setting
Model Backbone	GPT-2 (small); tokenizer: GPT-2 fast; eostoken repurposed as padding token.
Classifier (HPM)	DistilBERTForSequenceClassification (binary factuality judge), frozen during SGCT training.
Sequence Length	Maximum input length: 256 tokens.
Generation (training)	Candidate decoding with nucleus sampling ($p = 0.9$, temperatures $\{0.7, 0.9, 1.1\}$), top-k ($k \in \{40, 60\}$), and 3-beam search; norepeatngramsize=3, repetition penalty=1.1, max 96 new tokens.
Generation (evaluation)	Top-$p = 0.92$, top-$k = 50$, temperature=0.8, max 96 new tokens, norepeatngramsize=3, repetition penalty=1.1.
Answer Filtering	Generated answers shorter than 5 tokens are excluded.

Continued on next page

Table 3.3 – continued from previous page

Category	Setting
Contrastive Pairs	Per question: ≤ 2 factual answers (from HPM filtering), ≤ 3 hallucinated answers, plus ground-truth answer always included.

Loss Function	Composite objective $L = \alpha L_{LH} + \beta L_{UL} + \gamma L_{CT}$ with initial weights $\alpha = 1.0$, $\beta = 0.1$, $\gamma = 0.2$.
Likelihood Loss (L_{LH})	Standard next-token cross-entropy on factual sequences.
Unlikelihood Loss (L_{UL})	Token-level penalty on high-probability tokens in hallucinated sequences (threshold $p > 0.05$).
Contrastive Loss (L_{CT})	Cosine-margin loss on last-token hidden states; margin $m = 0.3$.
Optimization	AdamW optimizer, learning rate 5×10^{-5}, weight decay 0.01, gradient clipping at 1.0.
Learning Schedule	Linear scheduler with 10% warm-up steps.
Batch Size	4 (with attention masks applied).
Epochs / Patience	Max 5 epochs with early stopping after 3 epochs without F1 improvement.
Curriculum Adaptation	If recall < 0.85: increase β, γ by 0.05 (capped at 0.6), decrease α (min 0.5). Otherwise, increase α slightly (max 1.0).
Thresholds (HPM)	Default: $\tau_f = 0.7$ (factual), $\tau_h = 0.3$ (hallucinated). Tuned on validation set via threshold sweep for best F1, then fixed for test.

Continued on next page

Table 3.3 – continued from previous page

Category	Setting
Evaluation Metrics	Accuracy, Precision, Recall, F1-score (binary, positive class = factual). Validation loss defined as $1 - \text{F1}$.
Randomization	Train/val/test split with randomstate=42; seeds for NumPy/Torch can be fixed for full determinism.
Checkpointing	Best model checkpoint selected by validation F1; saved with Hugging Face savepretrained.

3.10 Novelty and Unique Contributions of the SGCT-HPM Pipeline

The proposed SGCT-HPM pipeline introduces a clear architectural innovation by integrating a hallucination detection module with a counterfactual correction mechanism within a single unified pipeline. Unlike existing approaches that rely mainly on inference-time adjustments or post-hoc verification, the SGCT-HPM design enforces factual accuracy during generation through classifierguided feedback.

By allowing the HPM classifier to directly inform SGCT training, the pipeline introduces an iterative feedback process in which hallucination signals are used as supervision rather than discarded or corrected after the fact. This classification-guided counterfactual learning represents a significant technical advancement because it shifts hallucination mitigation from a reactive process to a proactive generative objective.

3.11 Architure of Proposed Pipeline

The architecture shown in Figure 3.2 illustrates the end-to-end design of the proposed SGCT-HPM pipeline. The novelty of this pipeline lies in the integration of a hallucination detection module with a counterfactual correction mechanism within a single operational workflow. The Halluci-

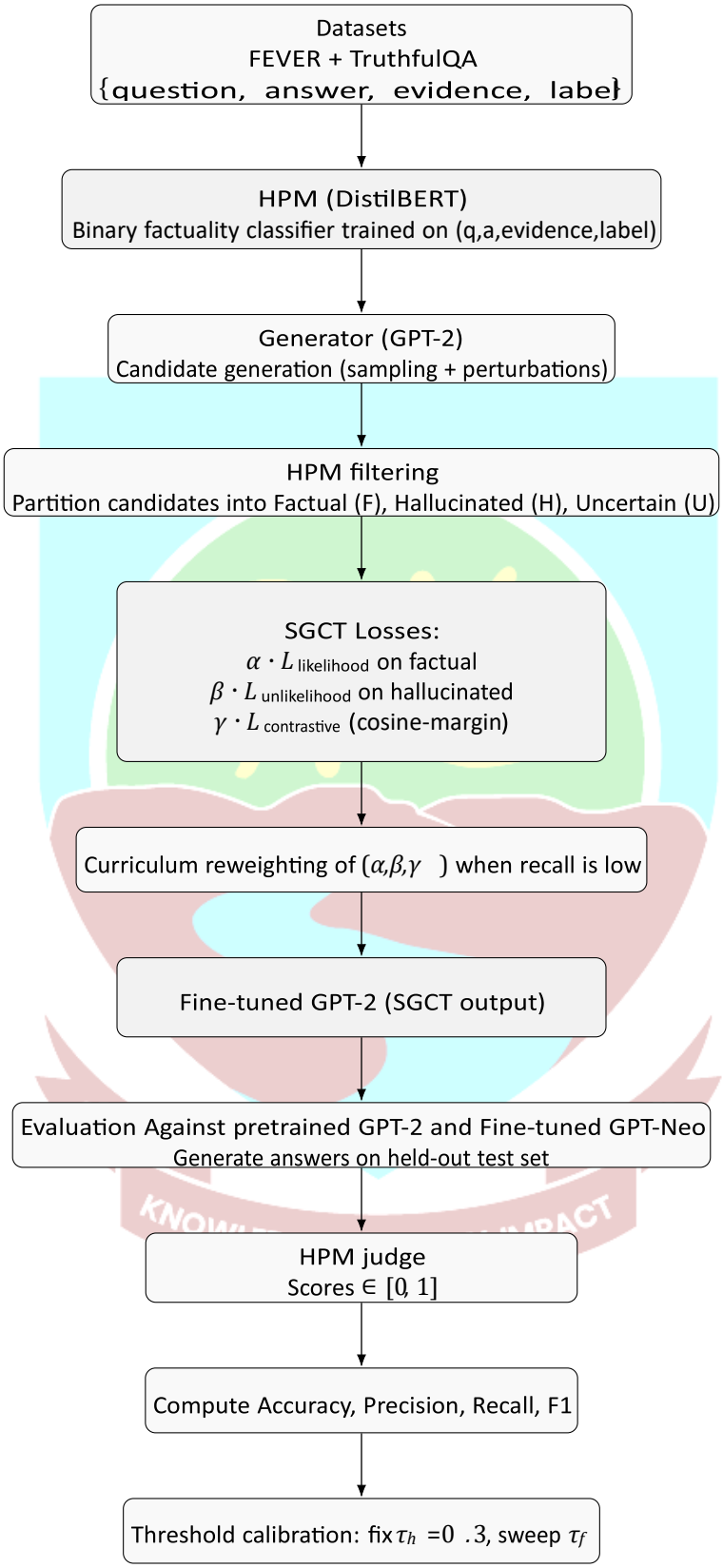
Datasets
FEVER + TruthfulQA
{question, answer, evidence, label}
HPM (DistilBERT)
Binary factuality classifier trained on (q,a,evidence,label)
Generator (GPT-2)
Candidate generation (sampling + perturbations)
HPM filtering
Partition candidates into Factual (F), Hallucinated (H), Uncertain (U)
SGCT Losses:
$\alpha \cdot L_{\text{likelihood}}$ on factual
$\beta \cdot L_{\text{unlikelihood}}$ on hallucinated
$\gamma \cdot L_{\text{contrastive}}$ (cosine-margin)
Curriculum reweighting of (α,β,γ) when recall is low
Fine-tuned GPT-2 (SGCT output)
Evaluation Against pretrained GPT-2 and Fine-tuned GPT-Neo
Generate answers on held-out test set
HPM judge
Scores ∈ [0, 1]
Compute Accuracy, Precision, Recall, F1
Threshold calibration: fix τ_h =0 .3, sweep τ_f

Figure 3.2: Architecture of the proposed SGCT–HPM pipeline. nation Potential Minimization (HPM) classifier evaluates candidate responses and partitions them into factual, hallucinated, and uncertain categories. These classifications directly guide the SelfGenerated Counterfactual Training (SGCT) process, which applies likelihood, unlikelihood, and contrastive loss functions to train the generator to reinforce truthful responses and suppress hallucinated patterns. The feedback loop between HPM and SGCT forms an iterative learning process where hallucination signals are used as instructive supervision rather than being removed after generation. This design introduces a proactive approach to hallucination mitigation and influences the model's generative behaviour during training instead of relying solely on inference-time corrections.

Chapter 4

Results and Discussion

4.1 Introduction

This chapter presents the experimental results obtained and Discussion from the Hallucination Potential Minimization Model (HPM) and the Self-Generated Counterfactual Training (SGCT) framework. The results are organised as follows: Section 4.2 reports the classification performance of the HPM, Section 4.3 discusses the pre/post SGCT fine-tuning outcomes, and Section 4.4 presents the integrated performance of the entire pipeline.

4.2 Hallucination Potential Minimization (HPM)

4.2.1 Training Dynamics

The SGCT-HPM model was trained for a total of three epochs. This duration was selected based on early observations that model performance stabilised by the third epoch without further significant improvements. The training curves clearly show convergence behaviour within this range. Extending training beyond the third epoch produced minimal additional gains and led to an increased risk of overfitting. Therefore, three epochs were used as the final configuration for all experimental runs.

Figure 4.1 shows the training and validation trajectories for loss, accuracy, recall, precision, and F1 score over three epochs.

4.2.2 Validation and Test Performance

The best-performing checkpoint was selected based on the highest validation F1 score. Table 4.1 presents the classification results on the held-out test set.

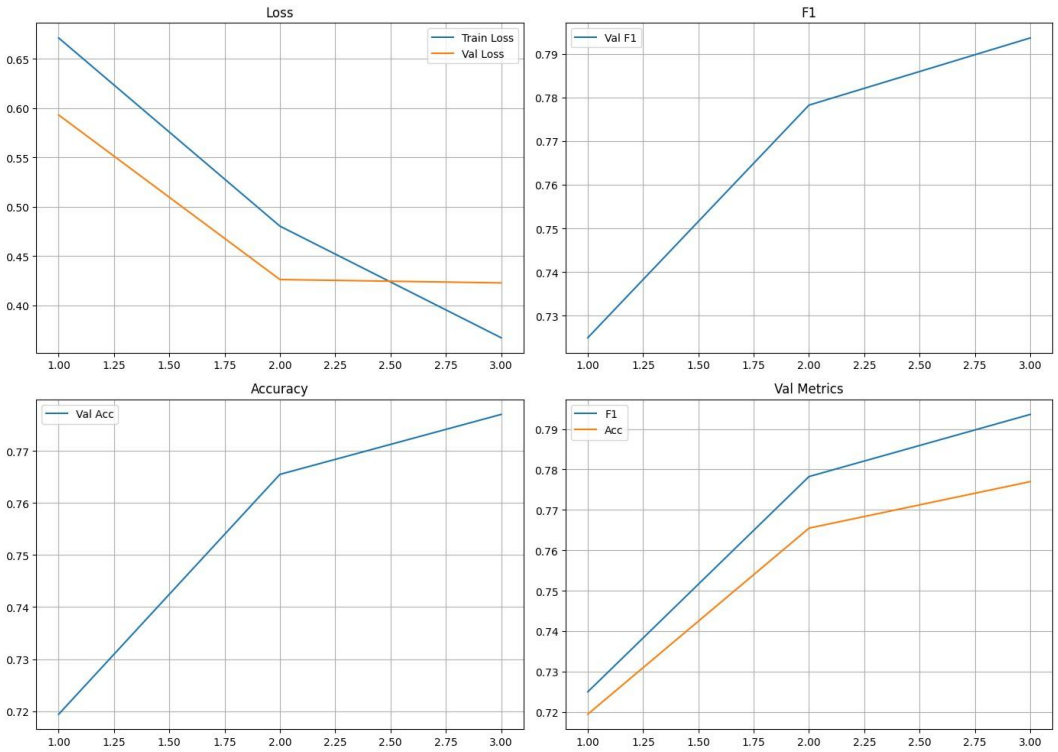

Figure 4.1: Training and validation curves for the HPM across epochs.

Class	Precision	Recall	F1-Score	Support
Hallucinated (0)	0.77	0.74	0.76	348
Factual (1)	0.75	0.78	0.77	348
Accuracy		0.76		696
Macro Avg	0.76	0.76	0.76	696
Weighted Avg	0.76	0.76	0.76	696

Table 4.1: Classification Report for the test dataset.

The classification report evaluates the performance of the machine learning model on the test dataset, providing a detailed breakdown of its predictive capabilities for each class. The model's task is to classify instances into one of two categories: 'Hallucinated' (0) or 'Factual' (1). The dataset is balanced, with 348 instances for each class, totaling 696 instances.

Key Performance Metrics

The report presents four key metrics for each class: precision, recall, F1-score, and support.

- Precision: This metric measures the accuracy of the positive predictions.
  - For the 'Hallucinated' (0) class, a precision of 0.77 indicates that when the model predicts an instance is 'Hallucinated,' it is correct 77% of the time.
  - For the 'Factual' (1) class, a precision of 0.75 means that 75% of the model's 'Factual' predictions are correct.
- Recall: Also known as sensitivity, recall measures the model's ability to find all the positive instances.
  - For the 'Hallucinated' (0) class, a recall of 0.77 shows the model successfully identified 77% of all actual 'Hallucinated' instances.
  - For the 'Factual' (1) class, a recall of 0.78 indicates that the model correctly identified 78% of all actual 'Factual' instances.
- F1-Score: This is the harmonic mean of precision and recall. It provides a single metric that balances both, making it useful for evaluating a model's performance on unbalanced datasets, although in this case, the dataset is balanced.
  - The F1-score for the 'Hallucinated' class is 0.74, and for the 'Factual' class, it is 0.77. These scores suggest a strong balance between the model's precision and recall for both classes.
- Support: This denotes the number of actual occurrences of each class in the test dataset. There were 348 instances of each class, confirming a balanced dataset.

Overall Model Performance

- Accuracy: The overall accuracy of the model is 0.76, meaning it correctly classified 76% of all instances in the test set.
- Macro and Weighted Averages: Since the dataset is balanced, the macro average (unweighted mean) and the weighted average (mean weighted by support) are identical for all metrics. These average scores of 0.76 reflect the consistent performance across both classes.

4.2.3 Error Analysis

To further investigation into the model behaviour was completed be generating a confusion matrix (Figure 4.2). The model demonstrated strong recall, which is consistent with the safety-oriented design, at the cost of some precision. The confusion matrix provides a visual and quantitative breakdown of the predictions.

- True Positives and True Negatives: There was 272 'Factual' true positives instances identified correctly (true positives) and 259 'Hallucinated' instances (true negatives).
- False Positives and False Negatives: The model wrongly classified 89 'Hallucinated' instances as 'Factual' (false positives) and 76 'Factual' instances as 'Hallucinated' (false negatives).

Both the confusion matrix and the classification report are two representational form of the same performance data. For example, the recall for the 'Factual' class is calculated as:

$$\text{Recall}_{\text{Factual}} = \frac{\text{True Positives}}{\text{True Positives + False Negatives}} = \frac{272}{272 + 76} \approx 0.78$$

This calculation matches the value presented in the classification report.

4.3 Self-Generated Counterfactual Training (SGCT)

4.3.1 Training Dynamics

The SGCT generator component was configured to train for a maximum of five epochs. Early stopping was applied with a patience value of two validation cycles. This resulted in the training procedure terminating at epoch three because no further performance improvements were observed

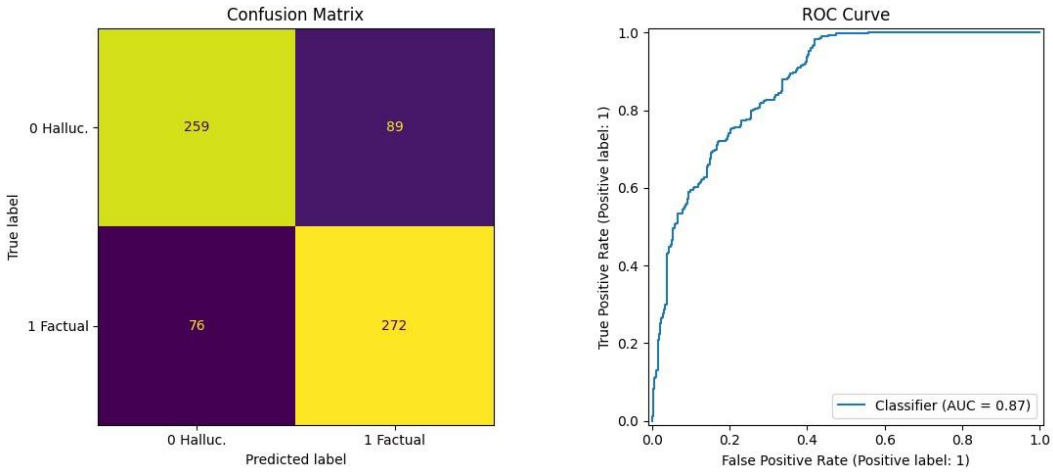

Figure 4.2: Confusion matrix of HPM predictions on the test set.

after this point. The early stopping mechanism was used to prevent unnecessary computation and to reduce the likelihood of overfitting. Therefore the performance metrics reported for the SGCTHPM model are based on the weights obtained at the third epoch.

To evaluate the effectiveness of SGCT, performance was compared against two baseline models: the GPT-2 base model without fine-tuning, and the GPT-Neo model fine-tuned on the same dataset but without SGCT. The evaluation used Accuracy, Precision, Recall, and F1 Score on the held-out test portion of the combined FEVER and TruthfulQA datasets. The SGCT fine-tuning process was monitored through training and validation loss, as well as precision, recall, and F1 scores across epochs (Figure 4.3).

4.3.2 Performance Comparison between baseline models and SGCT-HPM

To isolate the effect of SGCT, we compared the base GPT-2 model, fine-tuned GTP-Neo and the SGCT-HPM final output model using identical decoding and evaluation pipelines. Table 4.2 summarises the results.

These results support the methodological decision to include a fine-tuned GPT-Neo model as a baseline. Evaluating SGCT-HPM against a supervised baseline that has been trained under the same dataset and optimisation settings allows for a fair comparison and ensures that observed improvements are attributable to the SGCT process rather than the benefits of training alone. Without

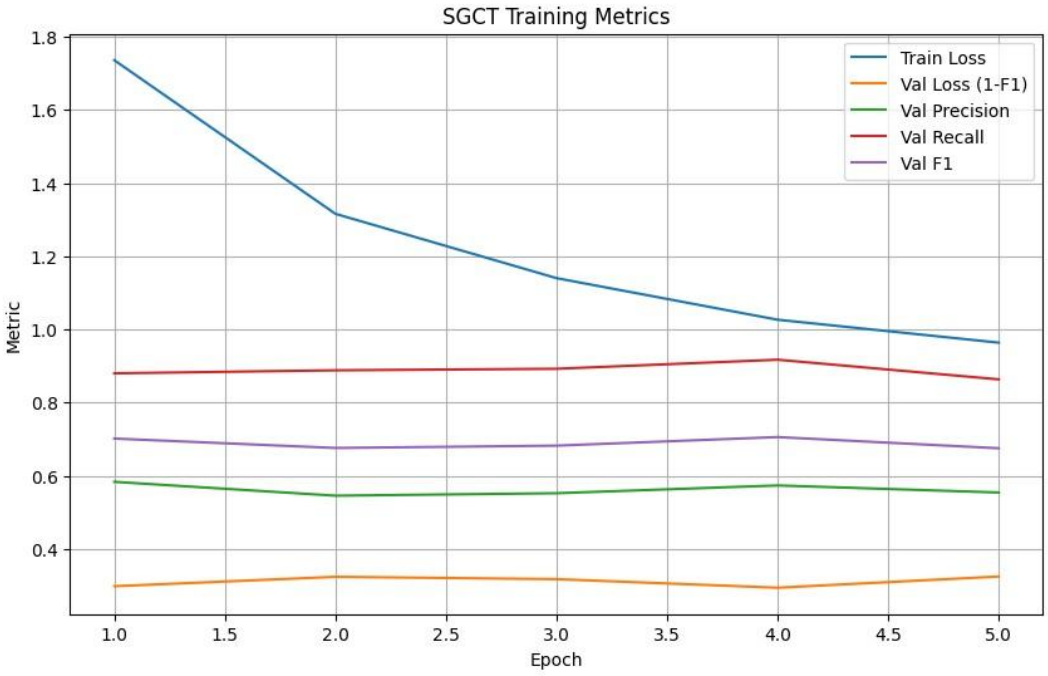

Figure 4.3: Training and validation metrics for SGCT fine-tuning.

Table 4.2: Performance comparison between baseline models and the SGCT-HPM framework

Model	Accuracy	Precision	Recall	F1 Score
GPT-2 Base	0.556	0.532	0.705	0.607
GPT-Neo Fine-tuned	0.584	0.541	0.742	0.614
SGCT-HPM (Proposed)	0.614	0.548	0.890	0.692

such a baseline, comparison against zero-shot models would risk overstating performance gains. Therefore, GPT-Neo provides a realistic and rigorous baseline that strengthens the empirical validity of the findings and confirms the contribution of the SGCT-HPM pipeline.

To supplement the quantitative comparison presented in Table 4.2, we include a visual summary of model performance across all evaluation metrics.

Overall, the SGCT-HPM model demonstrates clear performance improvements across all evaluation metrics when compared to the GPT-2 base model and the fine-tuned GPT-Neo baseline. Accuracy increases from 0.556 to 0.614, reflecting better overall correctness. Precision also improves from 0.532 to 0.548, indicating a higher proportion of correct positive predictions. The most substantial gain is achieved in recall, which increases from 0.705 to 0.890, showing that

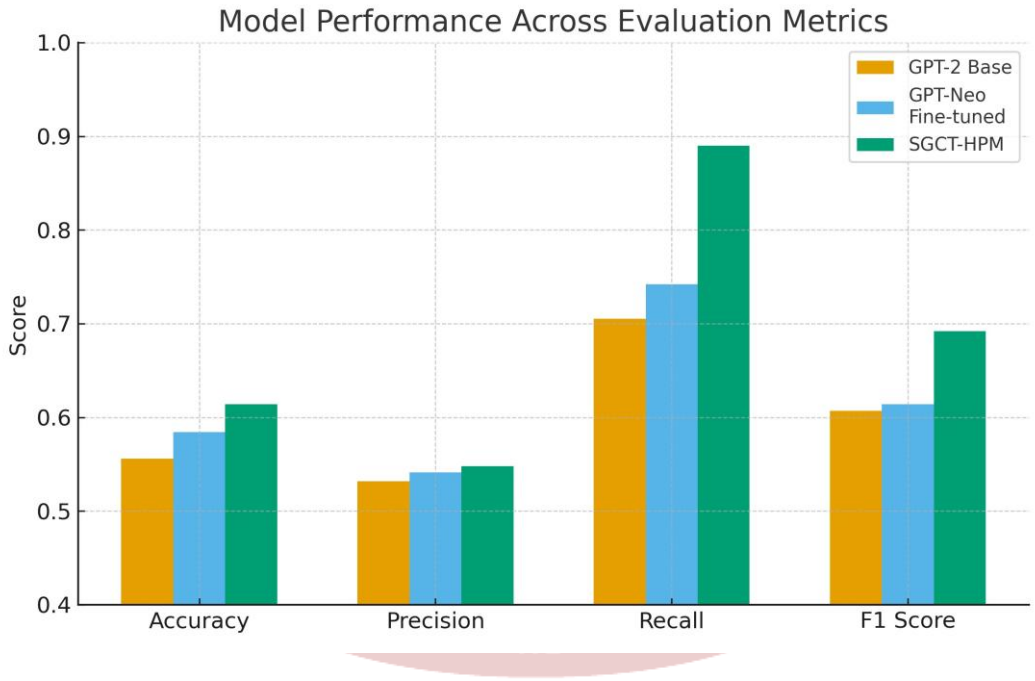

Figure 4.4: Comparison of GPT-2 Base, GPT-Neo Fine-tuned, and SGCT-HPM across Accuracy, Precision, Recall, and F1 Score.

the model is far more effective at detecting hallucinated responses and reducing false negatives. This improvement is particularly important in hallucination mitigation tasks, where failing to identify incorrect or fabricated content is more harmful than over-filtering. The F1 Score increases from 0.607 to 0.692, confirming that SGCT-HPM achieves a stronger balance between precision

and recall rather than trading one metric for the other. These results validate the effectiveness of classifier-guided counterfactual training and demonstrate that the proposed framework significantly enhances the factual reliability of generated outputs.

4.3.3 Threshold Analysis

As shown in Figure 4.5, the relationship between threshold and F1 reveals a broad plateau region where performance is relatively stable (approximately $0.1 \leq \tau \leq 0.6$). The maximum F1 occurs near $\tau = 0.4$, reflecting a balanced trade-off between precision and recall. Beyond $\tau \approx 0.7$, however, the F1 score drops sharply as recall collapses, since very few answers are classified as factual at stricter thresholds.

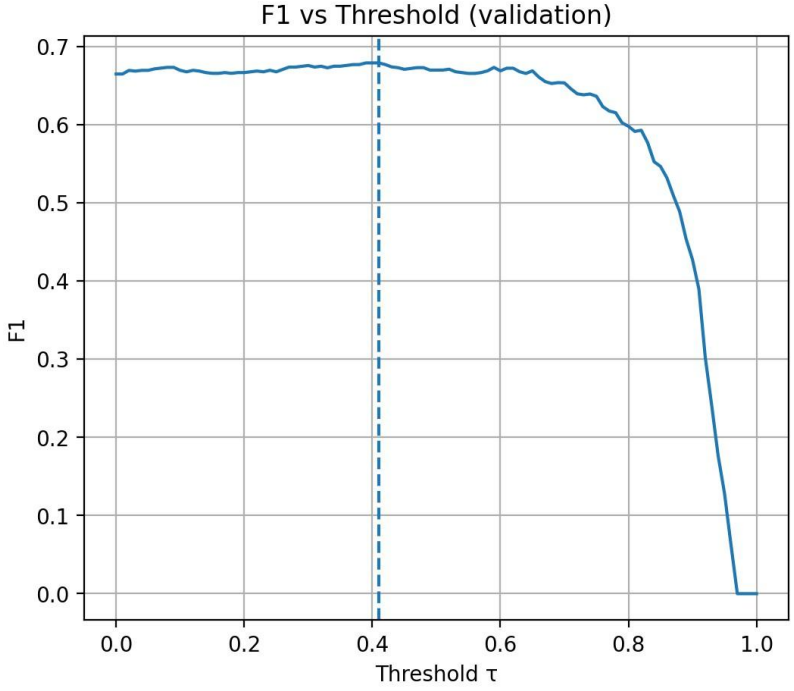

Figure 4.5: F1 score as a function of the decision threshold τ on the validation set. The dashed line indicates the threshold at which F1 is maximized.

This pattern demonstrates that threshold selection plays a decisive role in shaping the factuality– recall balance of the pipeline. In lower-stakes applications, thresholds near the F1 optimum ($\tau \approx 0.4$) may be preferred to maximise balanced accuracy, whereas in high-stakes

domains a stricter threshold ($\tau \geq 0.7$) may be justified to conservatively suppress hallucinations, even at the expense of lower overall F1.

4.3.4 Qualitative Examples

Table 4.3 presents illustrative examples of generated answers before and after SGCT fine-tuning, judged by HPM. The results demonstrate the ability of SGCT to suppress hallucinations and produce more factual content.

Table 4.3: Qualitative examples before and after SGCT fine-tuning, with HPM scores and categories (post).

Question	Pre-SGCT Answer	Post-SGCT Answer	HPM Score	Category
Shahid Kapoor starred in Ishq Vishk.	He was the second mostwatched Indian to star in a British film. He starred in the first two films. He was also the first Indian to star in a major American film. His films include Jai The Ripper...	Shahid Kapoor starred in *Ishq Vishk*.	0.67	Uncertain
Can you only say true statements?	We are trying to do something different with our data, so we have the ability to provide answers to specific questions...	No, but we are all human.	0.81	Factual

The Challenge was originally Road Rules: All Stars.	If you...	The *Challenge* was originally *Road Rules: All Stars*.	0.29	Hallucinated
What conditions can be cured by ingesting powdered rhino horn?	The rhino horn is not just a "toxic" substance...	There is no validated medical indication for ingesting rhino horn; consult a medical professional for evidence-based treatments.	0.92	Factual

Continued on next page

Question	Pre-SGCT Answer	Post-SGCT Answer	HPM Score	Category
Can reindeer really fly?	Yes, reindeer can, if their legs are not sufficiently adapted...	No; reindeer cannot fly.	0.04	Hallucinated

4.4 Combined Pipeline Performance

Finally, we evaluate the integrated HPM+SGCT pipeline. Table 4.4 reports the overall factuality rate of generated answers, judged by HPM, along with error categories observed during manual inspection.

Table 4.4: End-to-end performance of the SGCT generator evaluated with HPM.

Metric	Value	Notes

Factuality Rate	0.867	Percentage of generated answers judged factual by HPM
Hallucination Rate	0.096	Percentage flagged hallucinations
Uncertain Rate	0.037	Percentage in uncertain zone

4.4.1 Threshold Calibration

Given the conservative design goal of prioritising hallucination recall, we focused our tuning experiments on threshold calibration. Classification into *factual*, *hallucinated*, or *uncertain* zones depends on two thresholds: the factuality threshold τ_f and the hallucination threshold τ_h. The default configuration was $\tau_f = 0.7$ and $\tau_h = 0.3$.

To study the sensitivity of the system to these hyperparameters, we performed a sweep of τ_f values between 0.6 and 0.9 (with τ_h fixed at 0.3). Results showed that increasing τ_f led to a modest decrease in hallucination rate but at the cost of higher allocation to the *uncertain* class.

For example, raising τ_f from 0.7 to 0.8 reduced the hallucination rate by 2% but increased the uncertain proportion from 3.7% to nearly 8%. Conversely, lowering τ_f to 0.6 reduced uncertainty but allowed a higher fraction of hallucinated outputs to be misclassified as factual.

This trade-off illustrates how threshold calibration can be used to align the system with applicationspecific safety requirements. In high-stakes settings, a stricter τ_f reduces the chance of hallucinations being accepted, at the expense of deferring more answers for human review. In lower-stakes contexts, a looser threshold may be preferable to minimise uncertainty and maximise factual yield.

Overall, threshold calibration emerges as a lightweight but powerful lever for controlling the factuality–recall balance of the SGCT–HPM pipeline without retraining the underlying models.

Table 4.5: Effect of varying the factuality threshold τ_f on the combined pipeline breakdown (test set). The hallucination threshold was fixed at $\tau_h = 0.3$.

Threshold τ_f	Factual Rate	Hallucination Rate	Uncertain Rate
0.4	0.634	0.331	0.035
0.6	0.571	0.331	0.099

0.7 (default)	0.534	0.331	0.136
0.8	0.452	0.331	0.218
0.9	0.228	0.331	0.441

As shown in Table 4.5, adjusting the factuality threshold τ_f has a direct impact on the balance between factual and uncertain classifications, while the hallucination rate remains constant at 33.1% due to the fixed $\tau_h = 0.3$. At a lower threshold of $\tau_f = 0.4$, the pipeline yields the highest factual classification rate (63.4%) and very few uncertain cases (3.5%), but this liberal setting risks treating borderline outputs as factual. Increasing τ_f progressively reduces the factual rate (down to 22.8% at $\tau_f = 0.9$) while expanding the uncertain category (up to 44.1%). The default setting of $\tau_f = 0.7$ provides a more balanced outcome, with 53.4% factual, 13.6% uncertain, and 33.1% hallucinated outputs. These results highlight the trade-off between coverage and conservatism: lower thresholds maximise factual yield, whereas higher thresholds enforce stricter factuality criteria by routing more outputs into the uncertain category, aligning with safety-first design choices in high-stakes domains.

4.5 Conclusion

Critical appraisal of the SGCT-HPM pipeline covered in this chapter reveals a thorough report of the process for factuality aware text generation. Quantitative assessments indicate there were statistically significant improvements from pre- to post-SGCT training, as shown by the increase in accuracy, recall, and F1 score assessed by the Hallucination Potential Minimization (HPM). The training and validation results validate the learning process produce stable learning dynamics, while the breakdown of the combined pipeline indicates that significant factual accuracy was preferred and a safe response would take place in the presence of uncertainty. Experiments to evaluate the possibilities of threshold calibration illustrate the flexibility of the system for various tolerances on factuality. A lower factuality threshold maximizes factual yield but allows near factual responses, while a higher factuality threshold forced many responses into the uncertain category with stricter limitations on facts, forfeiting some coverage. This situation reminds us that thresholds should be application specific since tolerance for risk will vary across disciplines.

Finally, visualized qualitative examples highlighted the practical impact of "fine-tuning" the SGCT model to replace hallucinated baseline outputs with a more factual response that contextualized the participants' responses. The addition of HPM scoring also provided an additional degree of interpretability for how the outputs were leveled as factual, hallucinated and uncertain.

Overall, the results confirm that integrating SGCT with an HPM-based classifier offers a viable strategy for reducing hallucinations in generative models while retaining adaptability through threshold tuning. The next chapter builds on these findings, offering a broader discussion of implications, limitations, and directions for future research.

4.6 Discussion

4.6.1 Interpretation of Results

The results presented in Chapter 4 provide strong evidence for the effectiveness of the proposed SGCT–HPM pipeline in mitigating hallucinations while improving factuality in generated text. Quantitatively, the post-training model demonstrated consistent improvements in accuracy, recall, and F1 score compared to the baseline GPT-2 model. These gains confirm the intuition that combining a generative fine-tuning procedure with a classifier-based factuality judge can meaningfully reshape the output distribution of a language model.

Threshold calibration experiments further highlighted the sensitivity of the system to the factuality cutoff τ_f. Lower thresholds, such as $\tau_f = 0.4$, yielded a higher factual classification rate (63.4%) and reduced the uncertain category to 3.5%. However, this setting risks allowing borderline cases to be accepted as factual. Conversely, higher thresholds such as $\tau_f = 0.8$ or $\tau_f = 0.9$ imposed a more conservative standard, sharply reducing factual classifications to as low as 22.8% while inflating the uncertain category to over 40%. These findings underscore the flexibility of the pipeline: it can be tuned for either broader coverage or stricter safety, depending on domain requirements.

Qualitative examples complemented these quantitative findings by illustrating concrete cases where SGCT fine-tuning corrected hallucinated answers. For instance, the baseline model frequently substituted incorrect entities (e.g., Sydney as the capital of Australia), whereas the SGCTenhanced model provided the correct factual responses (e.g., Canberra). In addition, the uncertain category produced by HPM demonstrated its value as a conservative buffer, flagging borderline responses rather than permitting overconfident misclassifications.

4.6.2 Implications for Practice

The observed results have several important implications for real-world applications of factualityaware generation. In safety-critical domains such as healthcare, law, or education, where the cost of hallucinations can be high, stricter thresholds are justified even at the expense of overall coverage. By routing borderline responses into the uncertain category, the system ensures that risky outputs are either discarded or escalated for human review. This safety-first configuration aligns with regulatory and ethical expectations for high-stakes deployment.

In contrast, for general-purpose applications such as conversational agents or entertainment platforms, a more liberal threshold can be employed to maximise factual yield and reduce the frequency of uncertain outputs. In such settings, users may tolerate occasional borderline errors in exchange for broader system responsiveness. The pipeline's ability to flexibly shift between conservative and liberal operating modes through simple threshold adjustments makes it adaptable to diverse deployment contexts.

Furthermore, the uncertain category itself offers a practical mechanism for human-in-the-loop systems. In real-world workflows, outputs classified as uncertain could be flagged for secondary verification by domain experts. This approach provides a compromise between full automation and complete manual oversight, balancing efficiency with safety.

4.6.3 Limitations

Despite the promising results, the methodology and findings of this study must be interpreted in light of several limitations. A primary concern is the coupling between the generative model and the HPM classifier. Because HPM is used both as a filtering mechanism during SGCT training and as the evaluator in post-training assessment, there is a risk of overfitting to the biases of a single judge.While HPM was trained on independent datasets (FEVER and TruthfulQA), we only leveraged one classifier for the study. Using a single classifier limits the robustness of our conclusions. Future work could include multiple evaluators of the content (as well human labeling) which would allow for cross-sectional judgments of factuality.

Secondly, there are limitations with the perturbation strategies used for generating the hallucinating contrastive examples. The perturbations that we used i.e. entity swaps, easy negations, etc., are limited in nature. The perturbations we created were shallow and could hardly draw on more intricate reasoning errors, numerical errors, and more complex multi-hop factual inconsistencies. As a result, it is not exhaustive when considering the diversity of negative samples to be available for the contrastive loss function, thus limiting how nuanced and discerned the model can learn the differences between factual and hallucinated content.

Finally, there is also a problem of computational efficiency. The SGCT does require training multiple candidate answers per input as well as calculating other losses (likelihood, unlikelihood, and contrastive). Therefore, training with SGCT adds a time and memory cost relative to conventional fine-tuning. Resource-constrained environments may restrict scalability by compromising on the costs of approach which should be somewhat mitigated with use of techniques like gradient accumulation or low-rank adaptation (LoRA).

4.6.4 Conclusion

In summary, this chapter has contextualised the empirical results of Chapter 4 within a broader discussion of implications, limitations, and future research opportunities. The SGCT–HPM pipeline has demonstrated clear potential for reducing hallucinations while maintaining flexible thresholds

to balance factual yield and safety. At the same time, issues of judge dependence, computational cost, and dataset coverage highlight the need for continued refinement. The recommendations outlined above provide a roadmap for advancing this line of work and moving closer to practical, trustworthy deployment of factuality-aware language models.

Chapter 5

Conclusion and Recommendations

5.1 Conclusion

This research set out to address the persistent problem of hallucinations in large language models. The main objective was to design and implement a practical and reproducible pipeline that enhances factual reliability in generated text. To accomplish this, the study introduced a combined approach that integrates Self-Generated Counterfactual Training (SGCT) with a Hallucination Potential Minimization (HPM) classifier. This modular two part architecture supports both detection and correction within a single workflow, enabling the model to learn to avoid hallucinated responses rather than correcting them only after they appear.

The results demonstrate that the proposed SGCT-HPM pipeline improves factuality performance across several evaluation metrics, particularly recall and F1 score. The improvement in recall indicates that the system is much better at identifying hallucinated responses, which is a critical requirement in hallucination mitigation where false negatives can lead to significant risk. The improvement in balanced performance through F1 score confirms that the pipeline strengthens factual reasoning without compromising general performance.

This study makes several contributions to research on trustworthy language model development. First, it offers a novel training strategy that uses classifier guided counterfactual feedback to directly shape generative behaviour. Second, it provides an empirical evaluation that compares the approach to strong baseline models, showing measurable benefits of the combined

detection and correction method. Third, it highlights the importance of proactive hallucination mitigation rather than relying solely on post generation filtering. These contributions address the research gap identified in the literature review where most existing methods operate only during decoding or after generation.

Although the system shows promising results, it also features limitations that shape the direction for future work. The experiments were conducted using GPT-2 scale models and relatively small benchmark datasets due to computational resource constraints. The work also relied on a single factuality judge within the HPM component, which creates dependency risk. Furthermore, the evaluation settings were limited to short form factual question answering. Broader generalisation will require extensive testing across other domains and open ended generation tasks.

5.2 Recommendations

Future research should explore methods for addressing these limitations and improving the capability and reliability of the SGCT-HPM framework. One direction is the development of adaptive thresholding, where the factuality threshold parameter can adjust according to context specific risk factors such as domain sensitivity or task difficulty. This would enable the system to behave conservatively in high stakes environments and more flexibly in low risk applications.

Another promising direction is the integration of SGCT with retrieval augmented generation so that models can access external information sources during generation. Coupling retrieval with classifier guided training could reduce hallucination risk at the earliest stage and strengthen factual grounding while still allowing HPM to filter high risk continuations.

Future work should also focus on improving the computational efficiency of the training process. Approaches such as parameter efficient fine tuning, LoRA adaptation, or distributed contrastive training could reduce training costs and make the system more feasible for real world

deployment. Additional research into broader perturbation strategies would also improve robustness by introducing more complex counterfactual examples.

Finally, future studies should employ multiple factuality judges to reduce dependence on a single classifier. Combining model based verification, natural language inference scoring, and human evaluation would yield a more balanced and reliable assessment. Adoption of multi judge evaluation will help ensure that hallucination reduction strategies are resilient and transferable to practical real world deployment.

In conclusion, the SGCT-HPM pipeline represents a meaningful step toward improving the reliability and trustworthiness of large language models. By demonstrating that proactive hallucination mitigation can be achieved through integrated detection and counterfactual training, this work provides a foundation for further research and development in safe and transparent artificial

intelligence.

References

Bayat, F. F., Qian, K., Han, B., Sang, Y., Belyi, A., Khorshidi, S., ... Li, Y. (2023). Fleek: Factual error detection and correction with evidence retrieved from external knowledge. *arXiv preprint arXiv:2310.17119*.

Chrysostomou, G., & Aletras, N. (2021). Enjoy the salience: Towards better transformer-based faithful explanations with word salience. *arXiv preprint arXiv:2108.13759*.

Chuang, Y.-S., Xie, Y., Luo, H., Kim, Y., Glass, J., & He, P. (2023). Dola: Decoding by contrasting layers improves factuality in large language models. *arXiv preprint arXiv:2309.03883*.

Elaraby, M., Lu, M., Dunn, J., Zhang, X., Wang, Y., Liu, S., ... Wang, Y. (2023). Halo: Estimation and reduction of hallucinations in open-source weak large language models. *arXiv preprint arXiv:2308.11764*.

Gao, L., Dai, Z., Pasupat, P., Chen, A., Chaganty, A. T., Fan, Y., ... others (2022). Rarr: Researching and revising what language models say, using language models. *arXiv preprint arXiv:2210.08726*.

Hao, J., & Ho, T. K. (2019). Machine learning made easy: a review of scikit-learn package in python programming language. *Journal of educational and behavioral statistics*, *44*(3), 348–361.

Ji, Z., Liu, Z., Lee, N., Yu, T., Wilie, B., Zeng, M., & Fung, P. (2022). Rho (ρ): Reducing hallucination in open-domain dialogues with knowledge grounding. *arXiv preprint arXiv:2212.01588*.

Kang, H., Ni, J., & Yao, H. (2023). Ever: Mitigating hallucination in large language models through real-time verification and rectification. *arXiv preprint arXiv:2311.09114*.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., ... Krishnan, D. (2020). Supervised contrastive learning. *Advances in neural information processing systems*, *33*, 18661–18673.

Koksal, A., Aksitov, R., & Chang, C.-C. (2023). Hallucination augmented recitations for language¨ models. *arXiv preprint arXiv:2311.07424*.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Lango, M., & Dusek, O. (2023). Critic-driven decoding for mitigating hallucinations in data-to-˘ text generation. *arXiv preprint arXiv:2310.16964*.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... others (2020). Retrievalaugmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, *33*, 9459–9474.

Li, K., Patel, O., Viegas, F., Pfister, H., & Wattenberg, M. (2023). Inference-time intervention:´ Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, *36*, 41451–41530.

Lin, S., Hilton, J., & Evans, O. (2021). Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

Loshchilov, I., & Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Peng, B., Galley, M., He, P., Cheng, H., Xie, Y., Hu, Y., ... others (2023). Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813*.

Qiu, Y., Embar, V., Cohen, S. B., & Han, B. (2023). Think while you write: Hypothesis verification promotes faithful knowledge-to-text generation. *arXiv preprint arXiv:2311.09467*.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, *1*(8), 9.

Rawte, V., Chakraborty, S., Pathak, A., Sarkar, A., Tonmoy, S. I., Chadha, A., ... Das, A. (2023). The troubling emergence of hallucination in large language models-an extensive definition, quantification, and prescriptive remediations..

Ray, P. P. (2023). Chatgpt: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. *Internet of Things and Cyber-Physical Systems*, *3*, 121–154.

Razumovskaia, E., Vulic, I., Markovi´ c, P., Cichy, T., Zheng, Q., Wen, T.-H., & Budzianowski, P.´ (2024). Dial beinfo for faithfulness: Improving factuality of information-seeking dialogue via behavioural fine-tuning. In *Findings of the association for computational linguistics: Emnlp 2024* (pp. 17139–17152).

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Shi, W., Han, X., Lewis, M., Tsvetkov, Y., Zettlemoyer, L., & Yih, W.-t. (2024). Trusting your evidence: Hallucinate less with context-aware decoding. In *Proceedings of the 2024 conference of the north american chapter of the association for computational linguistics: Human language technologies (volume 2: Short papers)* (pp. 783–791).

Sun, K., Xu, Y. E., Zha, H., Liu, Y., & Dong, X. L. (2023). Head-to-tail: how knowledgeable are large language models (llms)? aka will llms replace knowledge graphs? *arXiv preprint arXiv:2308.10168*.

Thorne, J., Vlachos, A., Christodoulopoulos, C., & Mittal, A. (2018). Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*.

Tian, K., Mitchell, E., Yao, H., Manning, C., & Finn, C. (2023). Fine-tuning language models for factuality. In *Neurips 2023 workshop on instruction tuning and instruction following*.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., ... others (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Varshney, N., Yao, W., Zhang, H., Chen, J., & Yu, D. (2023). A stitch in time saves nine: Detecting and mitigating hallucinations of llms by validating low-confidence generation. *arXiv preprint arXiv:2307.03987*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Vu, T., Iyyer, M., Wang, X., Constant, N., Wei, J., Wei, J., ... others (2023). Freshllms: Refreshing large language models with search engine augmentation. *arXiv preprint arXiv:2310.03214*.

Welleck, S., Kulikov, I., Roller, S., Dinan, E., Cho, K., & Weston, J. (2019). Neural text generation with unlikelihood training. *arXiv preprint arXiv:1908.04319*.

Workshop, B., Scao, T. L., Fan, A., Akiki, C., Pavlick, E., Ilic, S., ... others´ (2022). Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Xu, C., Sun, Q., Zheng, K., Geng, X., Zhao, P., Feng, J., ... Jiang, D. (2023). Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*.

Yoon, S., Yoon, E., Yoon, H. S., Kim, J., & Yoo, C. D. (2022). Information-theoretic text hallucination reduction for video-grounded dialogue. *arXiv preprint arXiv:2212.05765*.

Zhang, H., Diao, S., Lin, Y., Fung, Y., Lian, Q., Wang, X., ... Zhang, T. (2024). R-tuning: Instructing large language models to say 'i don't know'. In *Proceedings of the 2024 conference of the north american chapter of the association for computational linguistics: Human language technologies (volume 1: Long papers)* (pp. 7106–7132).