Think Right, Not More: Test-Time Scaling for Numerical Claim Verification

Primakov Chungkham*
TU Delft
primakovch@gmail.com

Venktesh V*
Stockholm University
venktesh.viswanathan@dsv.su.se

Vinay Setty
University of Stavanger
vsetty@acm.org

Avishek Anand
TU Delft
avishek.anand@tudelft.nl

Abstract

Fact-checking real-world claims, particularly numerical claims, is inherently complex that require multistep reasoning and numerical reasoning for verifying diverse aspects of the claim. Although large language models (LLMs) including reasoning models have made tremendous advances, they still fall short on fact-checking real-world claims that require a combination of compositional and numerical reasoning. They are unable to understand nuance of numerical aspects, and are also susceptible to the *reasoning drift* issue, where the model is unable to contextualize diverse information resulting in misinterpretation and backtracking of reasoning process. In this work, we systematically explore scaling test-time compute (TTS) for LLMs on the task of fact-checking complex numerical claims, which entails eliciting multiple reasoning paths from an LLM. We train a verifier model (VERIFIERFC) to navigate this space of possible reasoning paths and select one that could lead to the correct verdict. We observe that TTS helps mitigate the *reasoning drift* issue, leading to significant performance gains for fact-checking numerical claims. To improve compute efficiency in TTS, we introduce an adaptive mechanism that performs TTS selectively based on the perceived complexity of the claim. This approach achieves **1.8**× higher efficiency than standard TTS, while delivering a notable **18.8%** performance improvement over single-shot claim verification methods. Our code and data can be found at `https://github.com/VenkteshV/VerifierFC`

1 Introduction

Automated verification of claims is critical to mitigate misinformation and disinformation. While many approaches focus on synthetically generated claims from Wikipedia (Jiang et al., 2020; Thorne et al., 2018; Aly et al., 2021), recent works have focused on fact-checking real-world claims (Augenstein et al., 2019; Schlichtkrull et al., 2023; Chen et al., 2022; V et al., 2024).

Real-world claims contain numerical (quantitative and temporal) elements (V et al., 2024; Alam et al., 2025) and verifying such claims necessitates systematically comparing numerical aspects within the claim against corresponding evidence provided. Consider the example in Figure 1, where verifying the claim about diaspora remittances requires ensuring that the claimed amount (KSh400 billion) and the specified year (2021) align correctly with the evidence, even when the evidence presents the

> **Mitigating *Reasoning drift* for Numerical FC**
>
> **Claim:** `Diaspora remittances "now stand at an all-time high of over KSh400 billion in 2021".`
> **[Evidence 1]**: diaspora remittances stand at an all-time-high at over ksh. 400 billion per annum as of 2021.
> **[Evidence2]**: Diaspora remittances were robust in 2021, with total remittances in 2021 reaching a record USD 3, 718 million. of daily users
> **[Verdict]**:True
>
> **LLM reasoning without VERIFIERFC**
> **[Reasoning]:** The evidence provides conflicting information about the exact amount of diaspora remittances in 2021, and some sources mention figures that are not directly comparable to KSh400 billion.
> **[Prediction]:** Conflicting
>
> **LLM reasoning with VERIFIERFC**
> **[Reasoning]:** The evidence supports the claim that diaspora remittances reached an all-time high of over KSh400 billion in 2021. However, there are minor variations in the wording and units used across different sources.
>
> **[Prediction]:** True

Figure 1: Example numerical claim demonstrating *reasoning drift* issue and mitigation offered by VERIFIERFC

*Both authors **contributed equally** to this work

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 24345–24363
November 4-9, 2025 ©2025 Association for Computational Linguistics

amount in a different currency (USD). Unlike binary verification tasks, real-world claims often contain contradictory or partially correct components, making verification more complex.

Standard automated fact-checking approaches employ a multistage pipeline involving fine-tuned NLI models. Recently, Large Language Models (LLMs) have emerged as promising alternatives to NLI-based methods for claim verification (Li et al., 2024). However, these approaches have been evaluated mainly in synthetic claims, and recent studies highlight significant shortcomings when applied to complex claims in the real world that involve numerical data (V et al., 2024; Setty, 2024; Schlichtkrull et al., 2023; Augenstein et al., 2024). Although prior work has acknowledged these limitations, they have not thoroughly analyzed why LLMs fail in realistic scenarios.

To address this gap, we analyze failure modes in both open-source and proprietary models of different sizes. Beyond noise in retrieval, we identify a critical reasoning challenge even in the presence of relevant evidence introduced specifically due to multi-aspect or inability to contextualize diverse information from evidence sources, termed **reasoning drift** due to issues in its ability for accurate analysis of claims and evidence. This differs fundamentally from the "overthinking" phenomenon observed in simpler Q&A contexts, where models unnecessarily elaborate on obvious details (Sui et al., 2025). While it can impact simple claims, reasoning drift is especially pronounced in complex cases with multiple aspects and diverse evidences, leading to unfocused reasoning that strays from the core verification task. Our analysis reveals that approximately **34%** of claims in the QuanTemp dataset suffer from this reasoning problem.

In the context of fact-checking, *Reasoning drift* stems from LLMs focusing on unimportant aspects of the claim or misinterpretation of information in diverse evidences as part of its reasoning process. For instance, in the example in Figure 1 the claim mentions an amount of *400 billion Ksh* (kenyan shillings). While one of the evidences confirm this figure, the other evidence shown in the example quotes an amount of 3718 mil. USD. Though this is equivalent to 400 billion ksh, the LLM interprets this quantity as not being equivalent to the one mentioned in the claim and falls short by billions, predicting its verdict as "Conflicting". To our knowledge, our work is the first to study the problem *reasoning drift* occuring due to **overanalysis** and explore test-time scaling as a solution to the same in the context of fact checking of real-world claims.

More formally reasoning drift due to overanalysis can be defined as the phenomenon in which:

"The LLM backtracks from it's current accurate reasoning trajectory due to misinterpretation of diverse information which may also stem from multi-aspect nature of claims and inability to contextualize information."

We introduce and investigate test-time scaling (TTS) (Zhang et al., 2025), an approach that allocates additional computational resources at inference time, enabling LLMs to explore multiple reasoning pathways to mitigate reasoning drift. To our knowledge, this is the first study to examine **test-time scaling** specifically for verifying complex factual claims. To effectively navigate and select from diverse reasoning paths, we train a dedicated verifier model (VERIFIERFC). Our experiments demonstrate that employing this verifier with a **best-of-N (BoN)** strategy significantly outperforms standard *majority-voting* (Wang et al., 2023) methods.

Since real-world claims are diverse, applying a fixed uniform computational effort across all claims is inefficient: simpler claims may consume unnecessary resources, and complex claims may not receive sufficient computational attention, resulting in suboptimal verification. To address this issue, we propose a novel *adaptive TTS* strategy that *dynamically* adjusts computational resources based on claim complexity, determined by analyzing *layerwise latent representations* within the LLMs. This adaptive method boosts verification performance by **18.8%** on complex numerical claims (QuanTemp) and **21.27%** on compositional claims (ClaimDecomp) compared to single-shot techniques, while also delivering **1.8x** greater computational efficiency than the standard TTS approach.

In this paper, we answer the following research questions through empirical validation:

RQ1: To what extent LLMs suffer from *reasoning drift* issue for complex numerical fact-checking tasks?

RQ2: How does test-time scaling impact the reasoning performance of LLMs for fact-checking complex numerical claims?

RQ3: How does test-time scaling compare to current strategies for fact-checking complex claims?

2 Related Work

2.1 Fact Checking Complex Real-World Claims

Automated fact-checking is key to mitigating growing misinformation and disinformation (Thorne et al., 2018; Aly et al., 2021; Guo et al., 2022). Existing works on automated fact-checking primarily focus on synthetic claims collected from Wikipedia (Thorne et al., 2018; Jiang et al., 2020; Aly et al., 2021; Nanekhan et al., 2025) that are not representative of real-world claims. More efforts have been made to build systems for real-world claims in domains like politics (Chen et al., 2022; Wang, 2017; Alhindi et al., 2018; Ostrowski et al., 2021; Nakov et al., 2025), science (Wadden et al., 2020; Vladika and Matthes, 2023; Wright et al., 2022), health (Kotonya and Toni, 2020) and climate (Diggelmann et al., 2021).

2.2 Reasoning drift due to overthinking in LLMs

LLMs with a large number of parameters are believed to exhibit emergent capabilities (Wei et al., 2022; Brown et al., 2020; OpenAI, 2023), including the ability to reason. Prior work shows that, when given few-shot examples, LLMs can produce step-by-step explanations (known as chains of thought or reasoning chains), resulting in improved performance on reasoning tasks (Wei et al., 2023; Huang and Chang, 2023; Yao et al., 2023b; Huang and Chang, 2023). However, there are known blindspots when used in few-shot and one-shot settings especially when numerical, temporal data is involved (Wallat et al., 2024a,b).

Moreover, recent studies show that large language models (LLMs) are prone to *overthinking* when they perform reasoning, a phenomenon where models generate unnecessarily long or elaborate reasoning chains that ultimately lead to incorrect conclusions (Sui et al., 2025; Chen et al., 2025). Overthinking has also been described as the tendency of LLMs to prioritize internal reasoning over external feedback (Cuadron et al., 2025). In knowledge-intensive tasks such as fact-checking, we observe that LLMs often backtrack or **overanalyze** their initial reasoning chain. This typically arises from the complexity of queries involving multiple aspects, or from misinterpreting conflicting information in the evidence. Crucially, this reasoning gap can occur regardless of the actual length of the reasoning chain. We refer to this problem as **reasoning drift**. While prior work has explored the application of LLMs to fact-checking (Li et al., 2024), none have explicitly addressed the reasoning drift phenomenon. Our findings suggest that reasoning drift presents a significant challenge for verifying complex numerical claims.

2.3 Test Time Scaling

Test-Time Scaling is a technique to improve the reasoning performance of LLMs by allocating additional inference-time compute (Zhang et al., 2025; Venktesh et al., 2025). Early approaches primarily focused on generating multiple outputs per query in parallel and selecting an answer via majority voting (Wang et al., 2023) or search-based methods (Yao et al., 2023a; Feng et al., 2024; Xie et al., 2023). Recent developments can be broadly categorized into two lines of work: (i) modifying the proposal distribution of LLM by fine-tuning (DeepSeek-AI et al., 2025; OpenAI, 2024; Singh et al., 2024; Madaan et al., 2023; Zelikman et al., 2022; Jin et al., 2025), and (ii) training a reward / verifier model to select the preferred answer from multiple candidates (Snell et al., 2024; Wang et al., 2024; Nichols et al., 2020; Cobbe et al., 2021; Uesato et al., 2022; Lightman et al., 2023). Our methodology focuses on the latter, often referred to as *search against a verifier*. It involves sampling multiple outputs and using a trained verifier model to select the best final output or the reasoning path that led to it.

Verifiers used to select the best solution, are divided into Outcome reward models (ORM) (Cobbe et al., 2021) or Process Reward Models (PRM) (Uesato et al., 2022; Lightman et al., 2023). We adopt a PRM based verifier and while the majority of the approaches use rejection sampling training only on positive reasoning paths, we also include incorrect solutions to provide better discriminative capacity to the verifier. Existing works also apply TTS uniformly to all test sample, while some claims may not require TTS, wasting compute resources. We are the first to explore applying TTS adaptively based on complexity level of the claim, drawing inspiration from adaptive RAG (Jeong et al., 2024).

3 Methodology

In this section, we present a test-time scaling (TTS) strategy aimed at mitigating reasoning drift, a rea-

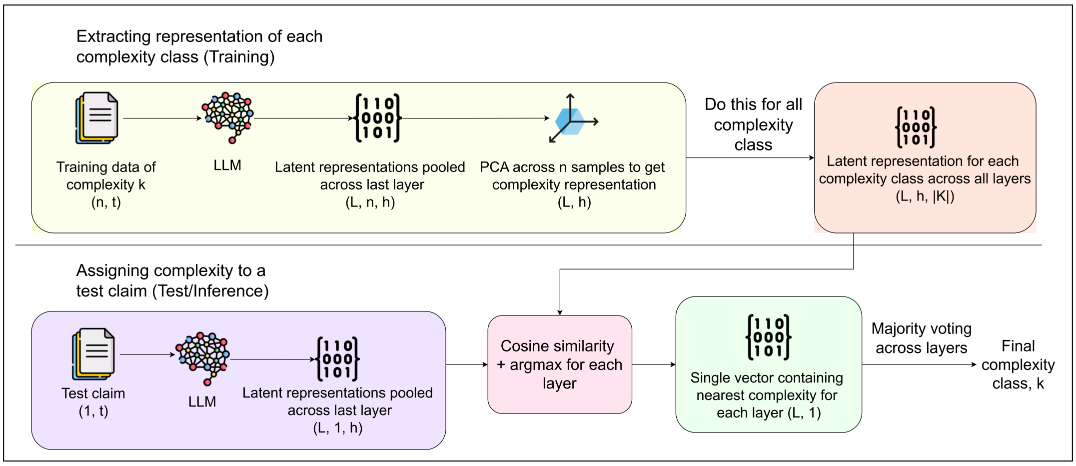

Figure 2: Training and inference of complexity assignment pipeline using latent representation-based method for adaptive decomposition.

soning failure wherein the model fails to prioritize relevant information or misconstrues aspects of the claim during inference. Our approach involves training a process verifier (Section 3.1) to select more accurate reasoning chains from multiple candidates. To further reduce unnecessary compute on straightforward claims, we explore an adaptive TTS mechanism that predicts claim complexity using layer-wise latent representations from the base LLM (Section 3.2). We adopt a standard fact-checking pipeline of BM25 retrieval + re-ranking to extract evidences for claim and employ LLMs for claim verification instead of standard NLI models.

3.1 Verifier for best path selection in TTS

Test-time scaling using a best-of-N strategy, in the context of fact-checking, entails generating multiple candidate reasoning paths for verifying a claim c, followed by selecting the best verdict using a verifier $\mathcal{V}$. We propose training a *process-based verification model* that scores reasoning paths rather than just final verdicts for a given claim, enabling the selection of the most insightful path for verification.

More formally, given a training corpus D of claims (c_i) and corresponding expert fact-checker verdicts (y_i), where $D = \{(c_1, y_1), \dots, (c_n, y_n)\}$, we first sample m different reasoning paths $\{(e_{11}, \dots, e_{1m}), \dots, (e_{n1}, \dots, e_{nm})\}$ from a pre-trained LLM $\mathcal{F}_{\text{LLM}}$ for each claim during the claim verification phase, along with their corresponding predicted verdicts $\{(\hat{y}_{11}, \dots, \hat{y}_{1m}), \dots, (\hat{y}_{n1}, \dots, \hat{y}_{nm})\}$. We then generate training data for the verifier by labelling each $(c_i, e_{ij}, \hat{y}_{ij})$ pair as 1 (positive sample) if the predicted verdict matches the ground-truth ($\hat{y}_{ij} == y_i$), and 0 otherwise (negative sample).

We use this constructed dataset of positive and negative reasoning paths to train the verifier model $\mathcal{V}(sc_i \mid (c_i, e_{ij}, \hat{y}_{ij}))$, which takes as input the original claim c_i, a reasoning path e_{ij}, and its corresponding predicted verdict $\hat{y}_{ij}$, and outputs a score based on the plausibility of the path leading to the correct verdict. The model is fine-tuned to assign higher scores to more appropriate reasoning paths by minimising a binary cross-entropy loss:

$$\begin{aligned}\mathcal{L}_{\mathcal{V}} = -\,&[y_i \cdot \log \mathcal{V}(sc_i \mid (c_i, e_{ij}, \hat{y}_{ij})) \\ &+ (1 - y_i) \cdot \log(1 - \mathcal{V}(sc_i \mid (c_i, e_{ij}, \hat{y}_{ij})))]\end{aligned}$$

The existing process verification approaches for training verifier primarily focus on rejection sampling when generating training data where only positive samples are considered. However, the supervision from training data generated above, helps the verifier account for nuances in the claim verification process by learning the ability to distinguish better between plausible and incorrect paths.

During inference for a test claim c_i we generate m such reasoning paths $\{(e_{11}, \hat{y}_{11})...(e_{1m}, \hat{y}_{1m}))\}$ and employ the verifier to score these reasoning paths by extracting logits score from output of the verifier $\mathcal{V}(sc_i|(c_i, e_{ij}, \hat{y_{ij}})..)$. The path with highest score is given as outcome of claim verification process. However, this best of N (BoN) strategy applies TTS for all claims uniformly not accounting for a scenario where TTS might not be useful. This is not compute-optimal and we propose an adaptive BoN strategy to circumvent this issue.

3.2 Adaptive Test-Time Scaling (TTS)

We adopt a semi-supervised approach to estimate the *complexity* of a claim by leveraging sub-question decomposition and clustering in the latent representation space of a language model. Our method is inspired by recent work on Adaptive RAG (Jeong et al., 2024; Mallen et al., 2023), which uses query complexity annotations to guide inference-time control in retrieval-augmented generation. In our case, we define complexity whether the claim requires decomposition to correctly verify its veracity. This decomposition signal serves as a proxy for reasoning difficulty, which we use to determine whether a claim should *trigger TTS*.

We construct the training data for adaptive TTS by annotating claims with sub-questions and track the number required to resolve the claim's veracity correctly. Claims are then categorized into two levels of complexity: level 0 includes claims that can be correctly verified without any decomposition; level 1 includes claims that require one or more sub-questions and are correctly verified after decomposition; and claims that remain misclassified even after decomposition are also assigned to level 1, reflecting unresolved ambiguity or model uncertainty.

Our adaptive decomposition decision algorithm draws inspiration from CTRLA (Liu et al., 2024), which posits that LLM latent representations encode semantically meaningful attributes. As shown in Figure 2, we extract latent representations and assign complexity labels accordingly.

We annotate a training set of claims $D_k = \{c_1, c_2, \ldots, c_n\}$ with complexity level k. Each claim is passed through a pretrained LLM $\mathcal{F}_{\text{LLM}}$, with L layers and latent size h, to extract token-level latent representations, following prior work on semantic properties of latent representations (Sun et al., 2025; Chuang et al., 2024). For an input claim c_i with t tokens, its latent representation $\mathbf{r}^{\mathbf{l}}$ for layer l is defined as:

$$\begin{aligned}\mathbf{r}^l &= \mathcal{F}^l_{\text{LLM}}(c_i) \\ &= [\mathbf{r}^{(l)}_{i,1}, \mathbf{r}^{(l)}_{i,2}, \ldots, \mathbf{r}^{(l)}_{i,t}] \in \mathbb{R}^{t\times h}\end{aligned}$$

More generally, the set of token representations across all layers can be expressed as:

$$\mathcal{F}_{\text{LLM}}(c_i) = \left\{\mathbf{r}^l_{i,j} \in \mathbb{R}^h \,\middle|\, 1 \le j \le t,\ 1 \le l \le L\right\},$$

where $\mathbf{r}^l_{i,j}$ is the hidden representation of the j^{th} token at layer l. Then, we summarize this representation using the last token as a proxy for the entire claim (following (Vaswani et al., 2023)). Formally, the pooled representation at layer l is defined as:

$$\mathbf{z}^l_i = \mathcal{P}(\{\mathbf{r}^l_{i,j}\}^t_{j=1}) = \mathbf{r}^l_{i,t} \quad \text{for } 1 \le l \le L,$$

where $\mathcal{P}$ denotes the pooling function over the token dimension. Generally, this can be represented as:

$$\mathcal{Z}_i = \left\{\mathbf{z}^l_i \in \mathbb{R}^h \,\middle|\, 1 \le l \le L\right\}.$$

We then apply Principal Component Analysis (PCA) to $\{\mathbf{z}^l_i\}^n_{i=1}$ for each layer l, and extract the first principal component:

$$\mathbf{u}^l_k = \text{PCA}_1\left(\left\{\mathbf{z}^l_i\right\}^n_{i=1}\right),$$

yielding complexity level representations:

$$\mathcal{U}_k = \left\{\mathbf{u}^l_k\right\}^L_{l=1},$$

For a test claim c, we extract its layerwise representations:

$$\mathcal{C} = \left\{\mathbf{c}^l \in \mathbb{R}^h \,\middle|\, 1 \le l \le L\right\}.$$

Cosine similarity is computed with each class prototype at each layer:

$$\begin{aligned}\mathcal{S} = \Big\{\mathbf{s}^l \in \mathbb{R}^{|K|} \,\Big|\, \mathbf{s}^l_k = \cos\left(\mathbf{c}^l, \mathbf{u}^l_k\right), \\ 1 \le k \le |K|,\ 1 \le l \le L\Big\}\end{aligned}$$

where $\mathbf{s}^l_k$ is the similarity with class k at layer l. We assign:

$$\hat{k}^l = \arg\max_k \ \mathbf{s}^l_k$$

and the final predicted complexity class is determined via majority voting across $\{\hat{k}^l\}^L_{l=1}$. This approach is similar to prototypical classification in ML literature (Snell et al., 2017; Sourati et al., 2024) or nearest centroid / rocchio classifier (Rocchio, 1971). The approach primarily pre-computes and indexes latent representations for complexity classes. Then at inference time we extract latent representations for the test claim and assign the complexity class based on the representation that is closest as measured by cosine similarity, similar to Rocchio classifier in IR literature. Based on the inferred complexity, we apply adaptive test-time scaling: one-shot inference for level-0 claims, and expanded reasoning for level-1 (moderate or high complexity) claims. This yields a compute-efficient TTS strategy for complex fact-checking, as shown in Section 5.

4 Experimental Setup

Datasets. We evaluate our methods on **QuanTemp** (V et al., 2024), which is the first benchmark for complex numerical claims. It comprises 9935 training, 3084 validation and 2495 test claims. We employ their training set to train the verifier model and create representations for the complexity classes, which are used for adaptive TTS.

ClaimDecomp (Chen et al., 2022) is another benchmark dataset that focuses on compositional real-world claims. We use its evaluation set (200 claims) to test the generalization capabilities of our verifier (trained on QuanTemp) to other types of complex claims.

Fact-checking pipeline: We use a standard fact-checking pipeline as adopted in (V et al., 2024) which comprises an initial retrieval of top-100 results for each claim using BM25 (Elasticsearch). We then re-rank using paraphrase-MiniLM-L6-v2 (Reimers and Gurevych, 2019) to select top-3 evidence snippets for each claim.

Metrics: We use per-class F1 scores, macro and weighted F1 score as evaluation metrics.

LLMs used for claim verification: We use Llama 3.1 (8B) and gpt-4o-mini as LLMs for claim verification stage and Deepseek-r1 which internally does TTS as baseline. We employ a temperature of 0.45 for generating diverse reasoning paths in TTS setting. We use Few-shot Chain of Thought (COT) prompting (in **Figure 10**) for eliciting reasoning for all baselines and our approach. We experiment with different number of reasoning paths (m=1,3,5,10) as elaborated in Section 5.

Verifier model for TTS: The Verifier model used for Best-of-N in TTS setting is based on the `Llama-3.2-3B` model. It is fine-tuned for classification using LoRA adapters (Hu et al., 2021) with a rank of 8 and a scaling factor (α) of 16. The model is fine-tuned for 3 epochs with a batch size of 32, using the AdamW optimizer with $\epsilon = 1 \times 10^{-8}$ and an initial learning rate of 1×10^{-3}.

Prompts: All prompts for claim verification, decomposition and LLM based annotations can be found in Appendix C.

5 Experimental Results

5.1 LLM Reasoning Drift - Manual Analysis

To answer **RQ1**, we conduct a systematic study of *reasoning drift* in numerical claim verification using QuanTemp. From a stratified sample of 100 claims, we use a system prompt (Figure 8) and a few-shot prompt with annotated examples (Figure 9, Appendix C) to instruct an LLM judge (gpt-4o-mini) to label each reasoning path as exhibiting *reasoning drift* (1) or not (0).

Two human annotators (graduate volunteers) then independently verify whether the LLM's label is correct, following the guidelines in Appendix A. We report a Cohen's Kappa of **74.13** indicating substantial agreement and LLM judge is **69%** accurate. Based on this procedure, after correcting the LLM judge outcomes, we find that approximately **34%** of sampled claims exhibit *reasoning drift*, suggesting it is a common and non-trivial failure mode. Examples are shown in Figure 1, Table 3, and Figures 6, 7 (Appendix B).

5.2 Effectiveness of test-time scaling for Claim Verification

To address **RQ2**, we evaluate TTS as a mitigation strategy (Section 3) with results shown in Table 1. We first establish a theoretical upper bound, where a perfect verifier selects the correct reasoning path and verdict whenever present.

As shown in Table 1, the upper bound achievable via TTS—across different LLMs—surpasses the oracle performance of fine-tuned NLI models reported in V et al. (2024). The *Oracle fine-tuned* row reflects state-of-the-art NLI models (e.g., RoBERTa-large) trained on human-written justifications and provided those at inference. In contrast, the TTS upper bound relies entirely on LLM-generated reasoning, without any human-authored justifications. This highlights LLMs' ability to elicit appropriate reasoning processes for complex fact-checking when guided by TTS.

Self-consistency (Wang et al., 2023), which aggregates multiple reasoning paths via majority voting, performs only marginally better than—or sometimes worse than—top-1 decoding. This is because it assumes the most frequent verdict is correct, overlooking the nuanced reasoning often required in fact-checking, especially for conflicting claims with diverse evidence.

As shown in Table 1, our trained verifier in both BoN and Adaptive BoN settings effectively selects the best reasoning path by leveraging task-specific signals. Adaptive BoN, which invokes TTS based on the predicted claim complexity, improves over top-1 decoding by **18.8%** and self-consistency by **16.69%** (Macro-F1 on QuanTemp). Adaptive BoN

Method	QuanTemp					ClaimDecomp				
	T-F1	F-F1	C-F1	**M-F1**	**W-F1**	T-F1	F-F1	C-F1	**M-F1**	**W-F1**
Other Baselines										
DeepSeek-R1 (7B) (DeepSeek-AI et al., 2025)	39.40	58.80	28.80	42.3	47.9	36.89	52.56	25.29	38.25	41.21
Llama-3.1 8B										
Top 1 Decoding (Wei et al., 2023)	33.5	66.4	34.7	44.8	52.5	19.75	44.03	44.44	36.07	37.15
Self-Consistency (Wang et al., 2023)	34.9	68.6	35.1	46.2	54.2	16.22	44.00	44.30	34.84	36.39
Best of N	44.20	75.30	40.30	53.20	**61.0**	22.22	48.00	46.91	39.05	40.22
Adaptive BoN (VERIFIERFC)	44.27	73.52	**43.93**	**53.91**	60.84	21.43	56.21	**49.70**	**42.44**	**44.13**
Upper bound	63.9	85.1	69.60	72.9	77.4	63.92	82.21	83.10	76.41	77.21
Gpt-4o-mini										
Top 1 decoding (Wei et al., 2023)	40.68	63.14	37.99	47.27	52.84	42.11	48.92	39.06	43.36	44.46
Self-Consistency (Wang et al., 2023)	29.70	63.20	42.33	45.07	51.83	47.41	50.00	34.19	43.87	45.35
Best of N	26.30	74.82	46.51	49.21	58.81	45.45	59.60	39.32	48.12	50.39
Adaptive BoN (VERIFIERFC)	**36.16**	72.61	44.22	**51.00**	**58.88**	49.18	57.93	**46.62**	**51.24**	**52.53**
Upper bound	45.34	85.37	67.19	65.97	73.41	58.21	66.67	61.68	62.19	62.91
Oracle (fine-tuned) (V et al., 2024)	56.86	82.92	48.79	62.85	69.79	66.67	66.67	45.76	59.69	60.50

Table 1: Claim verification performance of the models. The reward model is fine-tuned llama-3.2-3b and number of reasoning paths considered, m=10. The gains method is computed w.r.t the *top 1 Decoding* baseline.

Decomposition	T-F1	F-F1	C-F1	M-F1	W-F1
Majority Voting					
w/o decomp	34.89	68.63	35.09	46.20	54.18
with decomp	37.48	69.33	32.27	46.36	53.51
BoN					
w/o decomp	**44.16**	75.27	**40.27**	**53.23**	**60.97**
with decomp	43.77	**76.32**	35.21	51.77	60.29

Table 2: Analysis on impact of claim decomposition in test-time scaling setup on complex claim verification (using Llama-3.1 8B as base model).

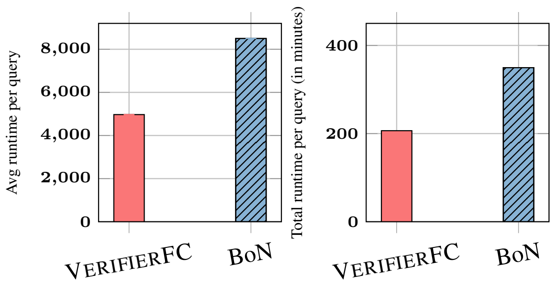

Figure 3: (Left) Avg. runtime per query (in milliseconds) comparison and (Right) Total **Runtime** comparison: Adaptive BoN (VERIFIERFC) vs BoN (Best of N).

also outperforms **DeepSeek-R1** by **27.44%**, highlighting the importance of a trained verifier for TTS in complex fact-checking.

We observe similar improvements on Claimdecomp eval set where Adaptive BoN (VERIFIERFC) outperforms Top-1 decoding significantly, demonstrating that our verifier can also generalize to other types of complex claims apart from numerical claims. We observe that this is primarily because the verifier used in BoN and adaptive versions is trained on data that considers positive and negative reasoning paths / process paths from the LLM for claim verification. This is unlike conventional Process or Outcome Reward Models which employ rejection sampling and only consider positive samples. This offers more discriminative capacity to the verifier that is able to distinguish the correct reasoning path among diverse paths even when their frequency is low. For instance, on second example in Table 3, the LLM is unable to reason that similar party manifesto was also adopted earlier in 1928 and was further expanded in 1956 and hence the claim is accurate. Instead, the model overanalyzes, backtracks from correct reasoning and passes the verdict as “Conflicting" due to mention of 1928 in one of the evidence not being in line with 1956 mentioned in claim. We observe that Self consistency also select “Conflicting" as verdict as 9/10 reasoning paths lead to this verdict. Rather BoN and adaptive BoN was able to rank the one correct reasoning path as the best solution leading to correct verdict.

We also observe that BoN and adaptive BoN with our trained process verifier helps mitigate the reasoning drift issue as evident from examples in Table 3. For instance, in the first example, one of the evidences exactly support the remittance value of 400 KSh and the second evidence quotes the equivalent value in US dollars. Under top-1-decoding, reasoning drift causes the LLM to classify the claim as "Conflicting" owing to the differing figures (ignoring the currency units) which

Method	Evidences
Claim	**Diaspora remittances "now stand at an all-time high of over KSh400 billion in 2021".**
	[Evidence 1]: diaspora remittances have increased tenfold over the last decade; and they now stand at an all-time-high at over ksh. 400 billion per annum as of 2021. [Evidence 2]: diaspora remittances were robust in 2021, with total remittances in 2021 reaching a record usd3,718 million. of daily users.
Top-1 decoding	[Reasoning]: The evidence provides conflicting information about the exact amount of diaspora remittances in 2021, and some sources mention figures that are not directly comparable to KSh400 billion. [Verdict]: Conflicting
Self Consistency	[Final Verdict]: Conflicting
VERIFIERFC (ours)	[Reasoning]: The evidence supports the claim that diaspora remittances reached an all-time high of over KSh400 billion in 2021. However, there are minor variations in the wording and units used across different sources. [Final Verdict]: True
Claim	**The 1956 Republican Party platform supported equal pay, the minimum wage, asylum for refugees, protections for unions and more.**
	[Evidence 1]: 1928 the gop platform read: the party favors freedom in wage we support an equitable minimum wage for american workers one providing fair wages [Evidence 2]: the 1956 republican platform pledged to provide assistance; protect social security; provide asylum for refugees; extend the minimum wage; strengthen labor laws to support union; assure equal pay...
Top-1 decoding	[Reasoning]: The evidence provided supports many of the claims made in the original statement, but some quotes are from earlier years or mention specific policies, and not all points are explicitly stated in the 1956 platform. [Final Verdict]: Conflicting
Self Consistency	[Final Verdict]: Conflicting
VERIFIERFC (ours)	[Reasoning]: The claim is supported by quotes from the 1956 Republican platform, which mention equal pay, minimum wage, asylum for refugees, protections for unions,. While there are minor nuances it is supported [...] [Final Verdict]: True

Table 3: Qualitative analysis of reasoning process by Self consistency and VERIFIERFC

is resolved when inferenced using BoN strategy (VERIFIERFC).

5.3 Efficiency of Adaptive Test-Time Scaling

We evaluate the efficiency of Adaptive BoN based TTS strategy by measuring total runtime on QuanTemp evaluation set and also record the average per claim processing time. The efficiency results are shown in plots in Figure 3 and claim verification performance in Table 1. We observe that Adaptive Bon is less time intensive (**1.8**$\times$) as it adaptively scales inference computation based on the complexity of the claims. We also observe from the plots that Adaptive scaling of test-time compute also reduces overall running time significantly, while achieving gains over plain TTS based BoN strategy, leading to judicious use of computational resources without sacrificing performance. We observe that on QuanTemp most of the claims assigned complexity level 0 belong to **statistical** category which are easy to verify as noted by authors in (V et al., 2024). We also independently evaluate the complexity consistency on its agreement with human notion of complexity by asking two annotators to annotate complexity levels for 100 claims sampled in a stratified manner. We observe a Cohen's kappa of **72.89** between annotators (Guidelines in Appendix A.2). Evaluating the classifier with annotations as ground truth, yields weighted precision of **71%**.

Method	QuanTemp		ClaimDecomp	
	M-F1	W-F1	M-F1	W-F1
Llama-3.1 8B				
m=3	50.60	57.47	37.96	39.71
m=5	51.49	58.39	39.64	41.25
m=10	**53.91**	**60.84**	**42.44**	**44.13**
Gpt-4o-mini				
m=3	47.98	54.82	44.15	45.28
m=5	49.26	56.82	48.24	49.24
m=10	**51.00**	**58.88**	**51.24**	**52.53**

Table 4: Performance with different number of reasoning paths m=3,5,10.

5.4 Impact of Decomposition on TTS

Claim decomposition is a popular approach for addressing complex claims with multiple aspects (V et al., 2024; Chen et al., 2022). Hence, to answer **RQ3** we analyze the impact of decomposing claim into sub-questions in combination with test-time scaling to understand the impact on final claim verification. We use ClaimDecomp (Chen et al., 2022) style yes/no sub-questions as they are found to have superior performance and easy to comprehend (V et al., 2024; Chen et al., 2022). The results are as shown in Table 2. We observe that claim decomposition surprisingly leads to negative effects on claim verification performance. On analysis, we observe this is primarily due to reasoning drift issue from LLM being elicited due to certain sub-questions focusing on aspects that are not central to facts being checked, causing drift.

Additionally, we observe that Adaptive BoN and BoN leads to diverse aspects of the claim being considered. Since this is the core objective of decomposition, TTS renders claim decomposition redundant, leading to diminishing returns (Tables 1, 2 and 3).

5.5 Ablations

We perform ablations by varying the scaling dimension of number of reasoning paths (m=3,5,10) using LLM. The results are as shown in Table 4. We observe that the performance increases with increase in number of reasoning paths, reaching the best performance at m=10. We do not scale beyond 10 due to computational and cost constraints.

6 Conclusion

We identify and study the *reasoning drift* issue in LLM reasoning for complex and numerical claims. To address this, we propose a Best-of-N (BoN) test-time scaling strategy, which outperforms Self-Consistency and Top-1 decoding. For improved compute-efficiency, we introduce Adaptive BoN (VERIFIERFC). VERIFIERFC achieves better performance than BoN while reducing computational cost. In future work, we aim to incorporate stronger verifier models and further optimize VERIFIERFC for performance and efficiency.

7 Limitations

While VERIFIERFC shows strong improvements over baselines, a notable gap remains compared to the upper bound. This may be addressed by integrating stronger verifier models with deeper linguistic and numerical understanding. Another promising direction is training verifiers on higher-quality, human-annotated data, which has been shown to improve reasoning performance. Additionally, our current approach leverages existing state-of-the-art retrieval and ranking approaches. However, noisy evidence can introduce bias that may affect both the performance and the validity of our analysis. Future work could incorporate more robust retrieval and re-ranking pipelines (Rathee et al., 2024) to reduce this noise, enabling a fairer and more accurate evaluation of veracity prediction methods.

8 Ethical Considerations and Risks

This work focuses on improving factual consistency in language model outputs through test-time scaling and verifier-guided reasoning. While our methods aim to avoid reasoning drift, they are subject to several ethical considerations.

First, our reliance on LLM-generated reasoning paths introduces potential risks of reinforcing model biases or factual errors during verifier training. Although the verifier is trained to select plausible reasoning, it may inherit systematic flaws from the base LLM. Second, our approach does not currently include fairness or bias mitigation across demographic attributes, which may be relevant for politically or socially sensitive claims.

Our datasets (QuanTemp, ClaimDecomp) are drawn from public sources (with create commons license) and include no personally identifiable or sensitive information. The claims focus on public data and domain-specific knowledge rather than private individuals. Human annotators involved in evaluation were presented with reasoning paths and asked to assess model performance under clear, task-specific instructions (Appendix A). No potentially harmful or offensive content was encountered during the annotation process.

9 Acknowledgements

We would like to thank Sowmya AS for her help in evaluating quality of claim difficulty estimator. We would also like to acknowledge Factiverse AI for contributing hardware for the experiments reported in this work.

References

Firoj Alam, Julia Maria Struß, Tanmoy Chakraborty, Stefan Dietze, Salim Hafid, Katerina Korre, Arianna Muti, Preslav Nakov, Federico Ruggeri, Sebastian Schellhammer, et al. 2025. The clef-2025 checkthat! lab: Subjectivity, fact-checking, claim normalization, and retrieval. In *European Conference on Information Retrieval*, pages 467–478. Springer.

Tariq Alhindi, Savvas Petridis, and Smaranda Muresan. 2018. Where is your evidence: Improving fact-checking by justification modeling. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 85–90, Brussels, Belgium. Association for Computational Linguistics.

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. The fact extraction and VERification over unstructured and structured information (FEVEROUS) shared task. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, pages 1–13, Dominican Republic. Association for Computational Linguistics.

Isabelle Augenstein, Timothy Baldwin, Meeyoung Cha, et al. 2024. Factuality challenges in the era of large language models and opportunities for fact-checking. *Nature Machine Intelligence*, 6:852–863.

Isabelle Augenstein, Christina Lioma, Dongsheng Wang, Lucas Chaves Lima, Casper Hansen, Christian Hansen, and Jakob Grue Simonsen. 2019. MultiFC: A real-world multi-domain dataset for evidence-based fact checking of claims. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4685–4697, Hong Kong, China. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jifan Chen, Aniruddh Sriram, Eunsol Choi, and Greg Durrett. 2022. Generating literal and implied subquestions to fact-check complex claims. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3495–3516, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025. Do not think that much for 2+3=? on the overthinking of o1-like llms.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R. Glass, and Pengcheng He. 2024. Dola: Decoding by contrasting layers improves factuality in large language models. In *The Twelfth International Conference on Learning Representations*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems.

Alejandro Cuadron, Dacheng Li, Wenjie Ma, Xingyao Wang, Yichuan Wang, Siyuan Zhuang, Shu Liu, Luis Gaspar Schroeder, Tian Xia, Huanzhi Mao, Nicholas Thumiger, Aditya Desai, Ion Stoica, Ana Klimovic, Graham Neubig, and Joseph E. Gonzalez. 2025. The danger of overthinking: Examining the reasoning-action dilemma in agentic tasks.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.

Thomas Diggelmann, Jordan Boyd-Graber, Jannis Bulian, Massimiliano Ciaramita, and Markus Leippold. 2021. Climate-fever: A dataset for verification of real-world climate claims.

Xidong Feng, Ziyu Wan, Muning Wen, Stephen Marcus McAleer, Ying Wen, Weinan Zhang, and Jun Wang. 2024. Alphazero-like tree-search can guide large language model decoding and training.

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards reasoning in large language models: A survey. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1049–1065, Toronto, Canada. Association for Computational Linguistics.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park. 2024. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity.

Yichen Jiang, Shikha Bordia, Zheng Zhong, Charles Dognin, Maneesh Singh, and Mohit Bansal. 2020. HoVer: A dataset for many-hop fact extraction and claim verification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3441–3460, Online. Association for Computational Linguistics.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning.

Neema Kotonya and Francesca Toni. 2020. Explainable automated fact-checking for public health claims.

Miaoran Li, Baolin Peng, Michel Galley, Jianfeng Gao, and Zhu Zhang. 2024. Self-checker: Plug-and-play modules for fact-checking with large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 163–181, Mexico City, Mexico. Association for Computational Linguistics.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step.

Huanshuo Liu, Hao Zhang, Zhijiang Guo, Jing Wang, Kuicai Dong, Xiangyang Li, Yi Quan Lee, Cong Zhang, and Yong Liu. 2024. Ctrla: Adaptive retrieval-augmented generation via inherent control.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, volume 36, pages 46534–46594. Curran Associates, Inc.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories.

Preslav Nakov, Federico Ruggeri, Sebastian Schellhammer, Vinay Setty, Megha Sundriyal10, Konstantin Todorov11, and V Venktesh. 2025. Overview of the clef-2025 checkthat! lab: Subjectivity, fact-checking, claim normalization, and retrieval. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 16th International Conference of the CLEF Association, CLEF 2025, Madrid, Spain, September 9–12, 2025, Proceedings*, page 199. Springer Nature.

Kevin Nanekhan, Erik Martin, Henrik Vatndal, Vinay Setty, and Avishek Anand. 2025. Flashcheck: Exploration of efficient evidence retrieval for fast fact-checking. In *European Conference on Information Retrieval*.

Eric Nichols, Leo Gao, and Randy Gomez. 2020. Collaborative storytelling with large-scale neural language models.

OpenAI. 2023. Gpt-4 technical report.

OpenAI. 2024. Learning to reason with llms. `https://openai.com/index/learning-to-reason-with-llms/`. Accessed: 2025-04-26.

Wojciech Ostrowski, Arnav Arora, Pepa Atanasova, and Isabelle Augenstein. 2021. Multi-hop fact checking of political claims. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3892–3898. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Mandeep Rathee, Sean MacAvaney, and Avishek Anand. 2024. Quam: Adaptive retrieval through query affinity modelling. *arXiv preprint arXiv:2410.20286*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

J. J. Rocchio. 1971. Relevance feedback in information retrieval. In G. Salton, editor, *The Smart retrieval system - experiments in automatic document processing*, pages 313–323. Englewood Cliffs, NJ: Prentice-Hall.

Michael Schlichtkrull, Zhijiang Guo, and Andreas Vlachos. 2023. Averitec: A dataset for real-world claim verification with evidence from the web.

Vinay Setty. 2024. Surprising efficacy of fine-tuned transformers for fact-checking over larger language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 2842–2846, New York, NY, USA. Association for Computing Machinery.

Avi Singh, John D. Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J. Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron Parisi, Abhishek Kumar, Alex Alemi, Alex Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet, Gamaleldin Elsayed, Hanie Sedghi, Igor Mordatch, Isabelle Simpson, Izzeddin Gur, Jasper Snoek, Jeffrey Pennington, Jiri Hron, Kathleen Kenealy, Kevin Swersky, Kshiteej Mahajan, Laura Culp, Lechao Xiao, Maxwell L. Bileschi, Noah Constant, Roman Novak, Rosanne Liu, Tris Warkentin, Yundi Qian, Yamini Bansal, Ethan Dyer, Behnam Neyshabur, Jascha Sohl-Dickstein, and Noah Fiedel. 2024. Beyond human data: Scaling self-training for problem-solving with language models.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4080–4090, Red Hook, NY, USA. Curran Associates Inc.

Zhivar Sourati, Darshan Girish Deshpande, Filip Ilievski, Kiril Gashteovski, and Sascha Saralajew. 2024. Robust text classification: Analyzing prototype-based networks. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12736–12757, Miami, Florida, USA. Association for Computational Linguistics.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Hanjie Chen, and Xia Hu. 2025. Stop overthinking: A survey on efficient reasoning for large language models.

Qi Sun, Marc Pickett, Aakash Kumar Nain, and Llion Jones. 2025. Transformer layers as painters.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process- and outcome-based feedback.

Venktesh V, Abhijit Anand, Avishek Anand, and Vinay Setty. 2024. Quantemp: A real-world open-domain benchmark for fact-checking numerical claims. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 650–660, New York, NY, USA. Association for Computing Machinery.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need.

V Venktesh, Avishek Anand, et al. 2025. Trust but verify! a survey on verification design for test-time scaling. *arXiv preprint arXiv:2508.16665*.

Juraj Vladika and Florian Matthes. 2023. Scientific fact-checking: A survey of resources and approaches.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.

Jonas Wallat, Abdelrahman Abdallah, Adam Jatowt, and Avishek Anand. 2024a. A study into investigating temporal robustness of llms. *Findings of the Association for Computational Linguistics*.

Jonas Wallat, Adam Jatowt, and Avishek Anand. 2024b. Temporal blind spots in large language models. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 683–692.

Peiyi Wang, Lei Li, Zhihong Shao, R. X. Xu, Damai Dai, Yifei Li, Deli Chen, Y. Wu, and Zhifang Sui. 2024. Math-shepherd: Verify and reinforce llms step-by-step without human annotations.

William Yang Wang. 2017. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426, Vancouver, Canada. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Dustin Wright, David Wadden, Kyle Lo, Bailey Kuehl, Arman Cohan, Isabelle Augenstein, and Lucy Lu Wang. 2022. Generating scientific claims for zero-shot scientific fact checking.

Yuxi Xie, Kenji Kawaguchi, Yiran Zhao, Xu Zhao, Min-Yen Kan, Junxian He, and Qizhe Xie. 2023. Self-evaluation guided beam search for reasoning.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. Tree of thoughts: Deliberate problem solving with large language models.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023b. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. 2022. Star: Bootstrapping reasoning with reasoning.

Qiyuan Zhang, Fuyuan Lyu, Zexu Sun, Lei Wang, Weixu Zhang, Wenyue Hua, Haolun Wu, Zhihan Guo, Yufei Wang, Niklas Muennighoff, et al. 2025. A survey on test-time scaling in large language models: What, how, where, and how well? *arXiv preprint arXiv:2503.24235*.

Annotator Guidelines for Claim Complexity Annotation

Annotation Instructions:
Please review the given claim in context of provided expert fact-checker justification and verdict. Trace the process of fact-checking by yourself and analyze how many aspects need to be verified in the claim. if the claim has more than one aspect that has to be verifier using multiple sources annotate it as 1 else annotate it as 0.

Figure 4: Annotator Guidelines for Claim Complexity Annotation.

A Annotation Guidelines

A.1 Guidelines for Manual analysis of reasoning drift issue

Figure 5 presents the guidelines to the annotator qualitatively evaluating the samples identified by judge LLMs for *reasoning drift*. Depending on the human judgements, they are instructed to assign a category `1` if they agree with the LLM's judgement and `0` otherwise. This assignment is used to compute Cohen's Kappa to evaluate inter-annotator agreements.

A.2 Annotation Guidelines for Claim Complexity

Figure 4 shows the guidelines for annotating complexity level for the claims.

B More examples exhibiting reasoning drift

Figure 6 and Figure 7 provides more qualitative examples where LLM exhibits *reasoning drift* during reasoning. This is mitigated by selecting the *best* reasoning sequence using VERIFIERFC.

C Prompts

This section describes the system and user prompts employed during different phases of our experiments. Figures 8 and 9 show the system prompt and a sample user prompt used to determine whether verifier-based reasoning reduces *reasoning drift* compared to its Top 1 Decoding counterpart. Figure 10 illustrates an example prompt designed to guide the fact-checking LLM in predicting a verdict and providing reasoning for a claim-evidence pair

Annotator Guidelines for Evaluating Reasoning Drift

Please review the LLM's judgment regarding the presence or absence of reasoning drift, along with its justification. Then, annotate whether you believe the label is correct and whether it aligns with our definition of reasoning drift.
Definition of Reasoning Drift: Reasoning Drift is defined as a behavior inherent to LLMs, where—particularly in the context of fact-checking—the model overfits to an aspect of the claim that is not central to verifying its truth. This can be triggered by irrelevant details or by diverse contexts.
Examples:

- An LLM might fail to recognize that two quantities expressed in different units (e.g., KSh and dollars) represent the same value.
- It might not realize that the same political manifesto could have been adopted by a party in both 1928 and a later year, leading to unnecessary doubt.

Annotation Instructions: If the LLM judge has determined that reasoning drift is present (or absent), and you agree that this decision aligns with the definition above, mark it as **1** (i.e., correct). Otherwise, mark it as **0** (i.e., incorrect). We will use this to compute inter-annotator agreement.

Figure 5: Guidelines for human annotators evaluating the reasoning drift prediction made by the judge LLM.

Mitigating *reasoning drift* for Numerical FC

Claim: `There's 100 million people that have preexisting conditions.`

[Evidence]: a 2018 analysis by avalere, a health care consulting firm, estimated that 102 million americans have preexisting conditions. a 2019 analysis by the left-leaning center for american progress suggested 135 million people.. also, it is unclear how many people with preexisting conditions would be at risk of losing their insurance entirely, or facing higher costs or having their conditions excluded from coverage. though bidens number is certainly within this range, he would need to provide more detail to support such a definitive number. the hhs issue brief, published in january 2017, estimated that between 61 million and 133 million americans have a preexisting condition.

[Verdict]: True

LLM reasoning without VERIFIERFC
[Reasoning]: The evidence suggests a range of estimates for the number of people with pre-existing conditions, with Biden's number being within this range but not universally accepted.
[Prediction]: Conflicting

LLM reasoning with VERIFIERFC
[Reasoning]: The claim is supported by multiple credible sources, including a former Vice President and health care consulting firms, with estimates ranging from 61 to 135 million Americans having pre-existing conditions. Biden's statement of 100 million falls within this range.

[Prediction]: True

Figure 6: Example numerical claim demonstrating *reasoning drift* issue and mitigation offered by VERIFIERFC

within a few-shot learning framework. Lastly, Figure 11 presents the system prompt used to generate sub-questions by breaking down a claim, following the methodology detailed in the CLAIMDECOMP paper (Chen et al., 2022).

Mitigating *reasoning drift* for Numerical FC

Claim: `Egas Moniz, the inventor of the controversial lobotomy technique for treating mentally ill patients (particularly people with schizophrenia), was awarded a Nobel Prize in 1949 for his work.`

[Evidence]: antnio egas moniz (1874-1955) of portugal was an ambitious and multitalented person. ... he had hoped and perhaps expected to receive the nobel prize for this work, and was disappointed when he realized he would not. in 1935 at an international neurology conference he saw a ... loses something by this operation, some spontaneity, some sparkle, some flavor of the personality." in 1942 they published an influential book promoting the practice. in the united states the number of lobotomies performed per year went from 100 in 1946 to 5,000 in 1949. that year moniz won the nobel prize in physiology/medicine for his contribution .... reduced the perceived need for most lobotomies.

[Verdict]: True

LLM reasoning without VERIFIERFC
[Reasoning]: The text states that Moniz was disappointed when he realized he would not receive the Nobel Prize for his work on visualizing blood vessels in the brain, but later won the Nobel Prize in Physiology or Medicine in 1949. This suggests that there is a discrepancy between the information provided.
[Prediction]: Conflicting

LLM reasoning with VERIFIERFC
[Reasoning]: The text states that Moniz was disappointed when he realized he would not receive the Nobel Prize for his work on visualizing blood vessels in the brain, but later won the Nobel Prize in Physiology/Medicine in 1949 for his contribution to psychosurgery.

[Prediction]: True

Figure 7: Example numerical claim demonstrating *reasoning drift* issue and mitigation offered by VERIFIERFC

System Prompt to Identify reasoning drift

Instruction: You are a fact-checking expert tasked with evaluating whether the reasoning generated using a verifier model mitigates a common issue in language model inference known as *reasoning drift*.
You will be provided with:

- A **claim**
- Its corresponding **evidence**
- The **verdict** for the claim
- A reasoning and prediction generated **without** a verifier model
- A reasoning and prediction generated **with** a verifier model

Goal: Determine whether the verifier-based reasoning successfully mitigates reasoning drift. reasoning driftis defined as:

- Focusing on details in the evidence that are not pertinent to the claim's veracity.
- Being overly strict with numerical or factual precision that could be interpreted more leniently.
- Verifying sub-questions individually but failing to aggregate them into a coherent, accurate final verdict.

Verifier reasoning is considered to mitigate reasoning driftif:

- It focuses only on evidence that is relevant to the claim.
- It avoids unnecessary scrutiny of minor or irrelevant details.
- It improves label accuracy by reasoning holistically about the claim.

Your task: Return a binary label based on your assessment:

- `1` — reasoning driftis mitigated in the verifier-based reasoning.
- `0` — reasoning driftis not mitigated (i.e., both approaches suffer equally or the verifier reasoning also overanalyzes).

Output format:

1. **Label:** `<0 or 1>`
2. **Explanation:** `<Brief justification referencing key differences in reasoning and use of evidence>`

Figure 8: System prompt provided to the judge LLM to evaluate and identify reasoning driftbetween reasoning with and without verifier support.

An example user prompt to identify reasoning drift

Example:

[Claim]: Joe Biden says, “There’s 100 million people that have preexisting conditions.”

[Evidence]: a 2018 analysis by avalere, a health care consulting firm, estimated that 102 million americans have preexisting conditions. ... hhs issue brief, published in january 2017, estimated that between 61 million and 133 million americans have a preexisting condition.

[Verdict]: True

LLM reasoning without verifier

[Reasoning]: The evidence suggests a range of estimates for the number of people with pre-existing conditions, with Biden’s number being within this range but not universally accepted.

[Prediction]: Conflicting

LLM reasoning with verifier

[Reasoning]: The claim is supported by multiple credible sources, including a former Vice President and health care consulting firms, with estimates ranging from 61 to 135 million Americans having pre-existing conditions. Biden’s statement of 100 million falls within this range.

[Prediction]: True

LLM judgement

[Label]: 1

[Explanation]: The reward model-based reasoning effectively mitigates overthinking by focusing on the relevant evidence that supports the claim, specifically highlighting that Biden’s figure of 100 million falls within a credible range of estimates (61 to 135 million). In contrast, the reasoning without the reward model gets bogged down in the conflicting nature of the estimates without recognizing that Biden’s number is still valid within that context. The reward model reasoning arrives at the correct label (TRUE) more reliably by synthesizing the evidence holistically rather than getting lost in the details of conflicting statements.

Input:

[Claim]: Diaspora remittances “now stand at an all-time high of over KSh400 billion in 2021”.

[Evidence]: Diaspora remittances were robust in 2021, with total remittances in 2021 reaching a record USD 3,718 million.

[Verdict]: True

LLM reasoning without verifier

[Reasoning]: The evidence provides conflicting information about the exact amount of diaspora remittances in 2021, and some sources mention figures that are not directly comparable to KSh400 billion.

[Prediction]: Conflicting

LLM reasoning with verifier

[Reasoning]: The evidence supports the claim that diaspora remittances reached an all-time high of over KSh400 billion in 2021. However, there are minor variations in the wording and units used across different sources.

[Prediction]: True

Figure 9: An example user prompt to the judge LLM to identify reasoning driftbetween the reasoning without verifier and with verifier

An example prompt to generate reasoning sequences for a claim - evidence pairs

System prompt:

You are a fact-checking assistant tasked with evaluating claims based on provided evidence. Your task:

1. Carefully analyze the important details from both the claim and the evidence provided.
2. Reason through the evidence step-by-step, synthesizing relevant information to assess the overall veracity of the claim.
3. Finally Classify the claim's veracity into one of the following categories:
   (a) True: The evidence fully supports the claim.
   (b) False: The evidence directly contradicts the claim.
   (c) Conflicting: The evidence contains contradictory elements or is inconclusive.

Response Format:
- [Label]: (True, False, or Conflicting)
- [Justification]: The reasoning steps used that led to your classification of the claim.

User prompt:

Example 1

[Claim]: Argentina's Central Bank is considering putting Lionel Messi's image on the 1000 pesos bank note.

[Evidence]: in fact, such was his influence in the argentina camp, the country's central bank are thinking about ... governing body are considering immortalising world cup winner lionel messi by putting his face on a 1000-peso note, according to reports.

[Verdict]: False

[Justification]: Is it true? The claim is false. Neither does the BCRA's website show Messi's picture on any banknote, nor has it made such an announcement.

Example 2

[Claim]: Politico did not report that 20 million Ukrainians would die to win war.

[Evidence]: in the ongoing russia-ukraine conflict, losses reported by different sources vary by orders of magnitude. sergei shoigu reported 5,937 russian soldier ... russian soldier killed, russian sources report only 0.3 losses. on the contrary, for every ukrainian soldier killed, russian sources report 4.3 ukrainian soldiers killed.

[Verdict]: True

[Justification]: Politico did not publish an article saying the deaths of 20 million Ukrainians were needed to win Russia 2019s .... The report cannot be seen when searching through Politico 2019s social media accounts.

Input prompt

Similar to the given examples, fact check the following claim using the evidence. Pay additional attention to numerical spans in claim and evidence and fact check by thinking step by step and output the label by performing entailment. Classify the entire claim strictly into one of the following categories: TRUE, FALSE or CONFLICTING.

[Claim]: Diaspora remittances "now stand at an all-time high of over KSh400 billion in 2021".

[Evidence]: Diaspora remittances were robust in 2021, with total remittances in 2021 reaching a record USD 3,718 million.

Give the reply in the following format:

[Label]: (SUPPORTS, REFUTES or CONFLICTING).
[Justification]:

Figure 10: An example prompt to generate veracity prediction and corresponding reasoning sequence for a claim, evidence pair.

Prompt to decompose a claim

System prompt:

You are tasked with generating a set of questions to break down and evaluate the veracity of a given fact-checking claim. Follow these strict guidelines when creating the subquestions:

1. **Comprehensive Coverage:** The subquestions should comprehensively address all relevant aspects of the claim to enable a thorough fact-check.
2. **Non-redundancy:** Ensure there is no overlap between the subquestions. Each question should be unique in meaning and not semantically repetitive.
3. **Relevance:** Every question must be relevant in determining the veracity of the claim. Avoid irrelevant or tangential questions.
4. **Clarity and Grammar:** Write questions that are clean, concise, and grammatically correct to maintain clarity.
5. **Yes/No Format:** Each question must be strictly answerable with a simple "Yes" or "No" response.

Figure 11: System prompt to generate sub-questions given a claim, adhering to the standards defined in the CLAIMDECOMP paper.