TruthTorchLM:
A Comprehensive Library for Predicting Truthfulness in LLM Outputs

Duygu Nur Yaldiz[1]* Yavuz Bakman[1]* Sungmin Kang[1]
Alperen Ozis[2] Hayrettin Eren Yildiz[3] Mitash Shah [1]
Zhiqi Huang [4] Anoop Kumar[4] Alfy Samuel[4] Daben Liu[4]
Sai Praneeth Karimireddy[1] Salman Avestimehr[1]
[1]University of Southern California [2]Independent Researcher
[3]Bogazici University [4] Capital One
`{yaldiz, ybakman}@usc.edu`

Abstract

Generative Large Language Models (LLMs) inevitably produce untruthful responses. Accurately predicting the truthfulness of these outputs is critical, especially in high-stakes settings. To accelerate research in this domain and make truthfulness prediction methods more accessible, we introduce `TruthTorchLM`[1],[2],[3] an open-source, comprehensive Python library featuring over 30 truthfulness prediction methods, which we refer to as *Truth Methods*. Unlike existing toolkits such as `Guardrails` (guardrails-ai), which focus solely on document-grounded verification, or `LM-Polygraph` (Fadeeva et al., 2023), which is limited to uncertainty-based methods, `TruthTorchLM` offers a broad and extensible collection of techniques. These methods span diverse trade-offs in computational cost, access level (e.g., black-box vs. white-box), grounding document requirements, and supervision type (self-supervised or supervised). `TruthTorchLM` is seamlessly compatible with both HuggingFace and LiteLLM, enabling support for locally hosted and API-based models. It also provides a unified interface for generation, evaluation, calibration, and long-form truthfulness prediction, along with a flexible framework for extending the library with new methods. We conduct an evaluation of representative truth methods on three datasets, TriviaQA, GSM8K, and FactScore-Bio.

1 Introduction

Generative Large Language Models (LLMs) have been widely adopted in many real-world applications due to their remarkable performance across a range of tasks, from code generation to conversational agents (Band et al., 2021). Despite these successes, LLMs inevitably produce outputs that are factually or logically incorrect, commonly referred to as *hallucinations* (Ravi et al., 2024). Detecting such untruthful outputs is particularly crucial in high-stakes applications where reliability and correctness are essential.

In response, numerous methods have been proposed to assess the truthfulness of LLM-generated content to support reliable decision-making. These include uncertainty estimation techniques, agentic tool use, multi-LLM collaboration strategies, supervised classification models, and document-based verification approaches. Each method varies in terms of computational cost, required access to model internals, and reliance on external resources. As LLM usage continues to grow, developing new techniques and refining existing ones remains crucial, given that truthfulness is a core requirement for trustworthy language generation.

To support research in this domain, an open-source library that consolidates existing methods and offers a flexible development framework is essential. Current software tools only partially meet this need. For instance, `Guardrails` (guardrails-ai) is an open-source library that focuses on document-based guardrails to assess the truthfulness of LLM outputs. However, it lacks support for a broader range of methods that do not rely on external documents, such as uncertainty estimation techniques. Similarly, `LM-Polygraph` (Fadeeva et al., 2023) provides implementations of uncertainty estimation methods, but it does not include supervised approaches, document-checking techniques, or tool-based strategies. A more comprehensive and extensible toolkit is needed to facilitate systematic evaluation and innovation across the full spectrum of truthfulness prediction methods.

To facilitate research and address the limitations of existing software in the domain of truthfulness prediction, we introduce `TruthTorchLM` (TTLM), an open-source library that currently implements over 30 *truth methods* with diverse algorithmic ideas. The library provides an intuitive and ex-

[1] https://github.com/Ybakman/TruthTorchLM
[2] https://www.youtube.com/watch?v=dgovBgUYz3w
[3] https://pypi.org/project/TruthTorchLM/

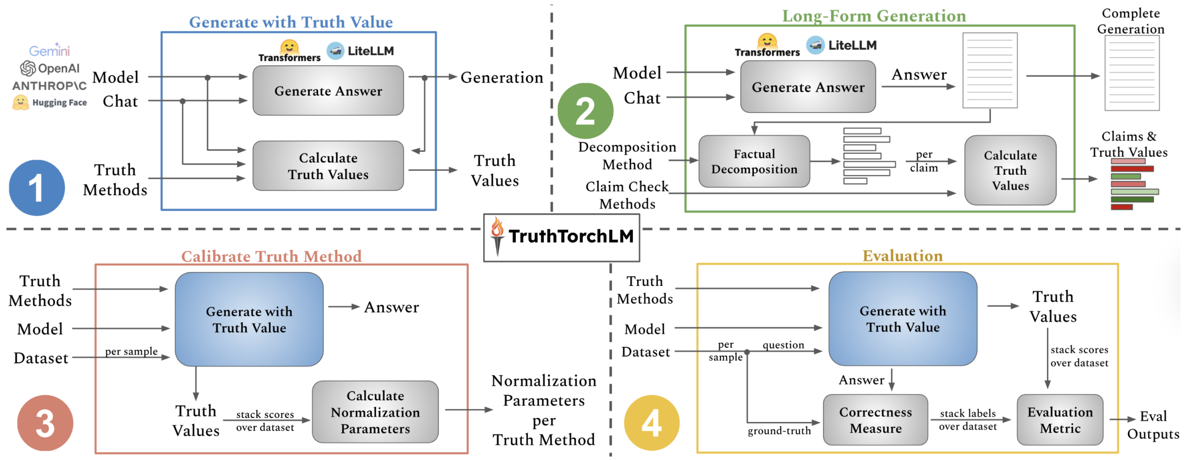

Figure 1: Overview of TruthTorchLM functionalities.

tensible interface for integrating new methods, enabling researchers to prototype and evaluate novel approaches with ease. `TruthTorchLM` is seamlessly compatible with both HuggingFace (Wolf et al., 2020) and LiteLLM (BerriAI), two of the most widely used frameworks for deploying LLMs in local and hosted environments. Beyond method implementation, the library offers comprehensive evaluation tools for benchmarking performance and includes calibration utilities to produce more interpretable truthfulness scores. Importantly, `TruthTorchLM` supports the application of truth methods to long-form generations, where multiple factual claims may be present and each claim requires individual assessment. This long-form setting represents a challenging and underexplored area of research, which is currently underserved by existing libraries.

Our contributions can be summarized as follows:

1. We release `TruthTorchLM (TTLM)`, an open-source library that implements over 30 truthfulness prediction methods, fully compatible with both HuggingFace and LiteLLM frameworks.
2. `TruthTorchLM` provides a unified interface for generation, evaluation, calibration, and long-form extension of existing truth methods, along with a flexible framework for adding new methods.
3. We conduct a comprehensive evaluation of a representative truth methods across three diverse datasets, TriviaQA, GSM8K, and FactScore-Bio, using both an open-weight model (LLaMA-3-8B) and a closed-weight model (GPT-4o-mini).

2 System Design and Features of TTLM

`TTLM` is designed around a central abstraction: *truth methods*, which are methods for predicting the truthfulness of LLM-generated outputs. Using TTLM, users can generate responses for any input query and apply one or more truth methods to assess the reliability of these outputs, whether they are short-form answers containing a single claim or long-form responses with multiple factual assertions. In addition to prediction, TTLM enables users to evaluate and calibrate the outputs of truth methods with just a few lines of code. In the following sections, we detail each of TTLM's core features and explain how they support robust and scalable research in truthfulness assessment.

2.1 Truth Methods

Truth Methods are methods designed to estimate the truthfulness or correctness of an LLM's response to a given query. These methods operate in an *off-the-shelf* manner, meaning they do not interfere with the generation process itself, but instead assign a post hoc truthfulness score (referred to as a *truth value*) after the response has been produced. Each Truth Method can optionally define its own parameters and must implement a standardized `forward` function, which takes as input the relevant generation-time information, such as generated token ids, the LLM and tokenizer objects, and returns a truth value. All methods inherit from the `TruthMethod` base class and follow a consistent interface, making the library easily extensible for users to implement custom methods.

Truthfulness estimation can be approached in a variety of ways, each with distinct trade-offs. The first major axis of variation is the use of external context: for example, Natural Language Inference (NLI) (Lei et al., 2023) methods assess truthfulness relative to supporting documents, while Uncertainty Quantification (UQ) methods rely solely on the model's output probabilities or internal states and do not require any external resources.

Table 1: Categorization of a representative subset of available methods in TruthTorchLM.

Truth Methods	Document-Grounding	Supervised	Access Level	Sampling-Required
LARS (Yaldiz et al., 2025)	✗	✓	Grey-box	✗
MARS (Bakman et al., 2024)	✗	✗	Grey-box	✗
SelfDetection (Zhao et al., 2024)	✗	✗	Black-box	✓
PTrue(Kadavath et al., 2022)	✗	✗	Grey-box	✗
AttentionScore (Sriramanan et al., 2024)	✗	✗	White-box	✗
CrossExamination (Cohen et al., 2023)	✗	✗	Black-box	✓
Eccentricity (Lin et al., 2024)	✗	✗	Black-box	✓
GoogleSearchCheck (Chern et al., 2023)	✓	✗	Black-box	✗
Inside (Chen et al., 2024)	✗	✗	White-box	✓
KernelLanguageEntropy (Nikitin et al., 2024)	✗	✗	Black-box	✓
MiniCheck (Tang et al., 2024)	✓	✗	Black-box	✗
Matrix-Degree (Lin et al., 2024)	✗	✗	Black-box	✓
SAPLMA (Azaria and Mitchell, 2023)	✗	✓	White-box	✗
SemanticEntropy (Kuhn et al., 2023)	✗	✗	Grey-box	✓
MultiLLMCollab (Feng et al., 2024)	✗	✗	Black-box	✓
SAR (Duan et al., 2024)	✗	✗	Grey-box	✓
VerbalizedConfidence (Tian et al., 2023)	✗	✗	Black-box	✗
DirectionalEntailmentGraph (Da et al., 2024)	✗	✗	Black-box	✓

In addition to document-grounding (1), we categorize truth methods along three further dimensions: (2) whether the method is supervised or self-supervised, that is, whether it requires separate training or can operate in a zero-shot fashion; (3) the level of access to the underlying model, ranging from black-box (output only), to gray-box (output probabilities), to white-box (internal representations); and (4) whether the method requires sampling, some approaches explore the output space to assign truth values, while others operate directly on a single response, which often reflects their computational cost. A representative subset of currently available methods and their categorization is provided in Table 1.

2.2 Unified Generation Interface

TTLM provides a unified generation interface that supports both locally hosted models via HuggingFace and API-based models through LiteLLM. This interface enables seamless integration of truth prediction with model inference, regardless of deployment type. The core function, generate_with_truth_value, accepts a chat history formatted as a list of message dictionaries (including system prompts, user queries, and prior exchanges), along with a list of predefined truth methods. It also supports standard generation parameters such as temperature, sampling strategy, and maximum token limits, with full compatibility with both HuggingFace and LiteLLM generation arguments.

The function returns the generated output alongside the assigned truth values for each specified truth method and each truth methods' specific details if desired. This streamlined interface enables effortless evaluation of generation reliability across diverse model backends. Figure 1.1 illustrates the function's architecture, and a code example is shown below.

Listing 1: Usage of generate_with_truth_value

```python
import TruthTorchLM as ttlm
# Define truth methods
lars = ttlm.truth_methods.LARS()
confidence = ttlm.truth_methods.Confidence()
self_detection = ttlm.truth_methods.
    SelfDetection(number_of_questions=5)
truth_methods = [lars, confidence,
    self_detection]

# Define chat input
chat = [{"role": "system", "content": "You are a
     helpful assistant."},
   {"role": "user", "content": "What is the
        capital city of France?"}]

# Generate with a HuggingFace model
output_hf_model= ttlm.generate_with_truth_value(
   model=model, tokenizer=tokenizer,
   messages=chat,
   truth_methods=truth_methods,
   max_new_tokens=100, temperature=0.7)

# Generate with an API-based model
output_api_model=ttlm.generate_with_truth_value(
   model="GPT-4o", messages=chat,
   truth_methods=truth_methods)
```

2.3 Evaluation of Truth Methods

Truth methods assign a scalar score, referred to as the *truth value*, to each model-generated output or individual claim. In short-form question answering tasks, evaluation follows a simple principle: if the generation is correct with respect to the ground truth, the assigned truth value should be high; if incorrect, it should be low. We explain the evaluation in long-form generations in Section 2.5.

Since free-form generations may vary lexically even when correct, we employ both traditional and modern correctness evaluators. Classical approaches include string-based metrics such as ROUGE (Lin, 2004), Exact Match, and BLEU (Papineni et al., 2002), while more recent methods such as *Model-as-a-Judge* leverage large language models to assess semantic correctness(Lin et al., 2024; Yaldiz et al., 2025). `TruthTorchLM` supports all of these evaluation criteria out-of-the-box. Once correctness labels are assigned, the performance of a truth method can be measured using both threshold-independent metrics, such as AUROC and PRR, and threshold-dependent metrics like F1 score, accuracy, precision, and recall. In Figure 1.4, we provide the design of evaluation functionality and, below is an example illustrating how to run evaluation using TTLM on the TriviaQA dataset:

Listing 2: Evaluating truth methods on TriviaQA

```
# Define correctness evaluator
model_judge = ttlm.evaluators.ModelJudge('gpt-4o-
    mini')

# Use built-in or custom datasets for evaluation
results = ttlm.evaluate_truth_method(
    dataset='trivia_qa',
    model=model, tokenizer=tokenizer,
    truth_methods=truth_methods,
    eval_metrics=['auroc', 'prr', 'accuracy'],
    correctness_evaluator=model_judge,
    size_of_data=1000, max_new_tokens=64)
```

2.4 Calibration of Truth Methods

Different truth methods may produce scores on varying ranges. For example, some methods output values between 0 and 1, while others produce unbounded negative scores (e.g., in the range $(-\infty, 0]$). As a result, directly comparing or interpreting these raw truth values can be challenging.

To address this, TTLM supports the calibration of truth method outputs. Calibration maps the original score range into a normalized interval, typically $[0, 1]$, where 0 represents minimal likelihood of truthfulness and 1 represents maximal likelihood. This enables both meaningful comparison across methods and the possibility of ensembling multiple truth scores into a unified signal, as demonstrated in prior work (Bakman et al., 2025).

We provide several calibration techniques, including Isotonic Regression (Han et al., 2017), and simple min-max normalization. Some calibration methods require labeled data for supervision, while others can operate in an unsupervised manner using only queries. Figure 1.3 provides the system overview of the calibration feature.

2.5 Predicting Truthfullness in Long Form Generation

Most questions require long-form generations that contain multiple factual claims, some correct, others incorrect. Evaluating the truthfulness of such outputs is non-trivial. Assigning a single truthfulness score to the entire generation lacks granularity and is not intuitive. To address this, we assign truth values to each individual factual claim within the generation, a strategy also adopted in prior work (Farquhar et al., 2024; Wei et al., 2024; Fadeeva et al., 2024; Zhang et al., 2024; Min et al., 2023).

To extract individual claims from a generation, the long-form text must first be decomposed. The quality of the decomposition process is critical for reliable truthfulness assessment. Each extracted claim must be self-contained and contextually coherent to enable accurate evaluation. TTLM's *Decomposition Methods* use language models with carefully designed prompts to ensure high-quality results across a wide range of topics. Users can choose any capable model and optionally enforce a structured output format to prevent parsing issues.

Next step is the assessment of truthfulness of the decomposed claims. However, most truth methods are designed for short-form generations and are not inherently applicable to long-form outputs. To address this limitation, TTLM introduces *Claim Check Methods*. These methods operate on individual claims extracted from long-form generations and assign truth scores to each claim. Similar to truth methods, each claim check method can define its own parameters and must implement a standardized forward function, which takes a claim along with relevant generation-time information and returns a truth value.

Claim check methods serve two main purposes: 1. Wrapper functionality: They adapt existing truth methods for claim-level checks (e.g., claim-specific question generation). In this case, the claim check method is initialized with one or more truth methods as input. TTLM provides three such wrapper methods by default. 2. Claim-level evaluation: These methods are specifically designed for assessing individual claims directly. All claim check methods inherit from the `ClaimCheckMethod` base class, ensuring a consistent, extensible interface.

To generate a long-context output with

corresponding truth values, TTLM contains `long_form_generation_with_truth_value`. This function accepts a chat history, a set of claim check methods, and a decomposition method. First, it generates a response ant this process is fully compatible with both Hugging Face and LiteLLM, supporting their respective generation configurations. The output is then decomposed into individual factual claims, each of which is evaluated using the specified claim check methods. The function returns the full generation, the set of claims with their associated truth values, and optionally, detailed metadata describing the decomposition and truth assessment processes. Figure 1.2 provides an overview of the long-form generation functionality, with a code example shown below.

Listing 3: Long-form generation with truth values

```
import TruthTorchLM.long_form_generation as LFG
#define a decomposition method
decomposition_method = LFG.decomposition_methods.
    StructuredDecompositionAPI(model="gpt-4o-
    mini", decomposition_depth=1)

#claim check method that apply truth methods
qa_generation = LFG.claim_check_methods.
    QuestionAnswerGeneration(model="gpt-4o-mini"
    , truth_methods=[confidence, lars])

#claim check methods designed for this purpose
ac_entailment = LFG.claim_check_methods.
    AnswerClaimEntailment( model="gpt-4o-mini",
    num_questions=3, num_answers_per_question=2)

#define a chat history
chat = [{"role": "system", "content": "You are a
     helpful assistant."},
       {"role": "user", "content": "Who is Ryan
           Reynolds?"}]

# Generate with an API-based model
out = LFG.long_form_generation_with_truth_value(
       model="gpt-4o-mini", messages=chat,
       decomp_method=decomposition_method,
       claim_check_methods=[qa_generation,
           ac_entailment])
```

TTLM evaluates claim check methods, either individually or in combination with truth methods, at the claim level within long-form generations. Since ground truth labels are typically unavailable for claims extracted from long-form outputs, we adopt the SAFE algorithm (Wei et al., 2024), which estimates claim correctness via Google Search. Once correctness labels are established, each (claim, truth value) pair is treated as a distinct evaluation sample, and assessment is conducted across all claims in the dataset. As in short-form evaluation, both threshold-dependent and threshold-independent metrics can be used to measure performance. A code sample for evaluating long-form generation is included in Appendix B.

3 Related Works

The most closely related open-source libraries to `TruthTorchLM` are `GuardrailsAI` (guardrails-ai) and `LM-Polygraph` (Fadeeva et al., 2023). `GuardrailsAI` implements guardrail mechanisms for safe and structured LLM outputs, primarily through document-grounded verification. `LM-Polygraph`, on the other hand, focuses on uncertainty quantification methods for generative language models.`TruthTorchLM` distinguishes itself from both in an important way. TTLM is explicitly designed for truthfulness prediction and aims to unify a wide spectrum of methods, ranging from uncertainty-based to supervised, document-grounded, and LLM-collaboration approaches. In contrast, `GuardrailsAI` is limited to document-grounded verification, while `LM-Polygraph` covers only uncertainty-based techniques, which represent a subset of the methods included in TTLM.

4 Experiments

We evaluate the performance of a subset of available truth methods listed in Table 1 using our proposed library, `TruthTorchLM`. In this section, we present the details of our experimental setup and provide a discussion of the results.

Datasets Our primary evaluation focuses on short-form question answering, a standard benchmark for assessing truthfulness. We use 1,000 samples from TriviaQA (Joshi et al., 2017) and GSM8K (Cobbe et al., 2021) for open-ended and mathematical reasoning questions, respectively. For long-form evaluation, we use FactScore-Bio (Min et al., 2023), which targets biographical questions with multi-fact generations.

Models We conduct evaluations using both open- and closed-weight language models. Specifically, we use `LLaMA-3-8B` (AI@Meta, 2024), an open-source model that enables full access to internal states, and `GPT-4o-mini` (OpenAI, 2023), a closed-weight API model. Note that white-box truth methods are not applicable to `GPT-4o-mini`.

Metrics As discussed in Section 2.4, different truth methods produce scores on varying numerical scales, which complicates the use of fixed thresholds for evaluation. While threshold-based metrics

Table 2: AUROC and PRR performance of truth methods on TriviaQA, GSM8K, and FactScore-Bio, across two models: LLaMA-3 8B and GPT-4o-mini.

	LLaMA-3 8B						**GPT-4o-mini**					
	TriviaQA		GSM8K		FactScore-Bio		TriviaQA		GSM8K		FactScore-Bio	
Truth Methods	**AUROC**	**PRR**	**AUROC**	**PRR**	**AUROC**	**PRR**	**AUROC**	**PRR**	**AUROC**	**PRR**	**AUROC**	**PRR**
LARS	0.861	0.783	0.834	0.719	0.677	0.391	0.852	0.766	0.840	0.686	0.640	0.294
MARS	0.763	0.635	0.730	0.488	0.660	0.367	0.792	0.668	0.735	0.480	0.655	0.405
SelfDetection	0.780	0.590	0.556	0.090	0.687	0.369	0.799	0.587	0.736	0.421	0.671	0.313
PTrue	0.727	0.485	0.654	0.307	0.670	0.368	0.772	0.509	0.833	0.636	0.658	0.372
AttentionScore	0.523	0.092	0.503	-0.024	0.644	0.263	–	–	–	–	–	–
CrossExamination	0.664	0.377	0.585	0.187	0.683	0.361	0.718	0.483	0.768	0.551	0.635	0.289
Eccentricity	0.809	0.645	0.703	0.450	0.695	0.415	0.817	0.632	0.754	0.455	0.671	0.421
GoogleSearchCheck	0.672	0.470	–	–	–	–	0.779	0.673	–	–	–	–
Inside	0.711	0.478	0.689	0.354	0.636	0.221	–	–	–	–	–	–
KernelLanguageEntropy	0.792	0.596	0.662	0.296	0.680	0.396	0.820	0.635	0.706	0.349	0.678	0.397
SAPLMA	0.850	0.726	0.815	0.642	0.651	0.347	–	–	–	–	–	–
SemanticEntropy	0.799	0.652	0.699	0.417	0.682	0.403	0.813	0.673	0.735	0.464	0.681	0.447
MultiLLMCollab	0.632	0.350	0.689	0.320	0.681	0.347	0.778	0.565	0.933	0.879	0.671	0.399
SAR	0.804	0.679	0.768	0.590	0.674	0.389	0.835	0.724	0.764	0.512	0.671	0.433
VerbalizedConfidence	0.759	0.547	0.579	0.234	0.698	0.460	0.836	0.740	0.652	0.369	0.717	0.514
DirectionalEntailmentGraph	0.745	0.513	0.731	0.501	0.659	0.347	0.778	0.532	0.736	0.439	0.658	0.380

such as accuracy can be informative, they require method-specific thresholds, introducing potential bias or instability in comparison.

To mitigate this issue, we primarily report threshold-free metrics, following prior work (Kuhn et al., 2023; Bakman et al., 2025). Specifically, we use the Area Under the Receiver Operating Characteristic Curve (AUROC) and the Prediction Rejection Ratio (PRR). AUROC measures a method's ability to distinguish between truthful and untruthful outputs across all possible thresholds, with values ranging from 0.5 (random performance) to 1.0 (perfect discrimination). PRR quantifies the relative precision gain obtained by rejecting low-confidence predictions and ranges from 0.0 (random rejection) to 1.0 (perfect rejection).

Correctness Measure Since our tasks involve free-form generation, the model outputs may be semantically correct even if they do not lexically match the ground truths. To account for this, we adopt the *LLM-as-a-judge* paradigm, following prior work (Bakman et al., 2025; Farquhar et al., 2024). Specifically, we prompt a language model with the question, the generated answer, and the reference answer, and ask it to provide a binary correctness judgment (0 or 1). For long-form generations in FactScore-Bio, where reference ground truths are unavailable we use the SAFE algorithm (Wei et al., 2024) to automatically extract and assess the correctness of individual factual claims within the generated text.

4.1 Discussion

The results are summarized in Table 2. Since each method entails different trade-offs, such as computational overhead, model access level, and supervision requirements, their performance varies accordingly. In short-form QA tasks (TriviaQA and GSM8K), LARS and SAPLMA achieve the highest performance, except on GSM8K with GPT-4o-mini, which is expected given that both are trained on labeled data. Among self-supervised methods, SAR performs best on both TriviaQA and GSM8K for the LLaMA-3-8B model. For GPT-4o-mini, Verbalized Confidence achieves the best results on TriviaQA, while MultiLLMCollab leads on GSM8K.

FactScore-Bio evaluates long-form generation, which typically involves multiple factual claims and thus presents a more challenging setting for truthfulness detection. On this task, performance generally drops across methods compared to short-form QA. Verbalized confidence achieves the best results on both models. Eccentiricity and Semantic Entropy performs next best as sampling based methods, with Semantic Entropy showing stronger results for GPT-4o-mini.

5 Conclusion

In this work, we introduced `TruthTorchLM`, an open-source library for evaluating and developing truthfulness prediction methods for large language models. TTLM unifies a diverse set of techniques under a common interface, supports both short- and long-form generation tasks, and includes tools for evaluation, calibration, and extensibility. We hope TTLM serves as a valuable resource for the community and accelerates research in building more trustworthy and reliable language models.

Ethics Statement

We acknowledge the ethical considerations associated with the development and release of truthfulness prediction tools for large language models (LLMs). Our library, TruthTorchLM, is designed to assist researchers and practitioners in systematically evaluating and improving the truthfulness of LLM outputs. It does not generate content on its own; any harmful or incorrect content produced by language models is not the product of this library. Our goal is to help detect and reduce untruthful outputs.

All experiments in this work were conducted on publicly available datasets (TriviaQA, GSM8K, and FactScore-Bio) that do not contain personally identifiable or sensitive information. No private or user-generated data was collected or used during development or evaluation. We encourage responsible use of our library and caution that automated truthfulness prediction should complement, not replace, human oversight, especially in high-stakes domains such as health, law, and finance.

References

AI@Meta. 2024. Llama 3 model card.

Amos Azaria and Tom Mitchell. 2023. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976, Singapore. Association for Computational Linguistics.

Yavuz Bakman, Duygu Nur Yaldiz, Sungmin Kang, Tuo Zhang, Baturalp Buyukates, Salman Avestimehr, and Sai Praneeth Karimireddy. 2025. Reconsidering llm uncertainty estimation methods in the wild. *Preprint*, arXiv:2506.01114.

Yavuz Faruk Bakman, Duygu Nur Yaldiz, Baturalp Buyukates, Chenyang Tao, Dimitrios Dimitriadis, and Salman Avestimehr. 2024. MARS: Meaning-aware response scoring for uncertainty estimation in generative LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7752–7767, Bangkok, Thailand. Association for Computational Linguistics.

Neil Band, Tim G. J. Rudner, Qixuan Feng, Angelos Filos, Zachary Nado, Michael W Dusenberry, Ghassen Jerfel, Dustin Tran, and Yarin Gal. 2021. Benchmarking Bayesian deep learning on diabetic retinopathy detection tasks. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

BerriAI. litellm.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. 2024. INSIDE: LLMs' internal states retain the power of hallucination detection. In *The Twelfth International Conference on Learning Representations*.

I Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, Pengfei Liu, et al. 2023. Factool: Factuality detection in generative ai–a tool augmented framework for multi-task and multi-domain scenarios. *arXiv preprint arXiv:2307.13528*.

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. *arXiv preprint arXiv:2110.14168*.

Roi Cohen, May Hamri, Mor Geva, and Amir Globerson. 2023. LM vs LM: Detecting factual errors via cross examination. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12621–12640, Singapore. Association for Computational Linguistics.

Longchao Da, Tiejin Chen, Lu Cheng, and Hua Wei. 2024. Llm uncertainty quantification through directional entailment graph and claim level response augmentation. *Preprint*, arXiv:2407.00994.

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. 2024. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5050–5063, Bangkok, Thailand. Association for Computational Linguistics.

Ekaterina Fadeeva, Aleksandr Rubashevskii, Artem Shelmanov, Sergey Petrakov, Haonan Li, Hamdy Mubarak, Evgenii Tsymbalov, Gleb Kuzmin, Alexander Panchenko, Timothy Baldwin, Preslav Nakov, and Maxim Panov. 2024. Fact-checking the output of large language models via token-level uncertainty quantification. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9367–9385, Bangkok, Thailand. Association for Computational Linguistics.

Ekaterina Fadeeva, Roman Vashurin, Akim Tsvigun, Artem Vazhentsev, Sergey Petrakov, Kirill Fedyanin, Daniil Vasilev, Elizaveta Goncharova, Alexander Panchenko, Maxim Panov, Timothy Baldwin, and Artem Shelmanov. 2023. LM-polygraph: Uncertainty estimation for language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 446–461, Singapore. Association for Computational Linguistics.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630.

Shangbin Feng, Weijia Shi, Yike Wang, Wenxuan Ding, Vidhisha Balachandran, and Yulia Tsvetkov. 2024. Don't hallucinate, abstain: Identifying LLM knowledge gaps via multi-LLM collaboration. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14664–14690, Bangkok, Thailand. Association for Computational Linguistics.

guardrails-ai. Guardrails.

Qiyang Han, Tengyao Wang, Sabyasachi Chatterjee, and Richard J. Samworth. 2017. Isotonic regression in general dimensions. *The Annals of Statistics*.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. Language models (mostly) know what they know. *Preprint*, arXiv:2207.05221.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*.

Deren Lei, Yaxi Li, Mengya Hu, Mingyu Wang, Vincent Yun, Emily Ching, and Eslam Kamal. 2023. Chain of natural language inference for reducing large language model ungrounded hallucinations. *Preprint*, arXiv:2310.03951.

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

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2024. Generating with confidence: Uncertainty quantification for black-box large language models. *Transactions on Machine Learning Research*.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Alexander V Nikitin, Jannik Kossen, Yarin Gal, and Pekka Marttinen. 2024. Kernel language entropy: Fine-grained uncertainty quantification for LLMs from semantic similarities. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

OpenAI. 2023. GPT-4 Technical Report. *Preprint*, arXiv:2303.08774.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Selvan Sunitha Ravi, Bartosz Mielczarek, Anand Kannappan, Douwe Kiela, and Rebecca Qian. 2024. Lynx: An open source hallucination evaluation model. *Preprint*, arXiv:2407.08488.

Gaurang Sriramanan, Siddhant Bharti, Vinu Sankar Sadasivan, Shoumik Saha, Priyatham Kattakinda, and Soheil Feizi. 2024. LLM-check: Investigating detection of hallucinations in large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Liyan Tang, Philippe Laban, and Greg Durrett. 2024. MiniCheck: Efficient fact-checking of LLMs on grounding documents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8818–8847, Miami, Florida, USA. Association for Computational Linguistics.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442, Singapore. Association for Computational Linguistics.

Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Zixia Hu, Jie Huang, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, Cosmo Du, and Quoc V Le. 2024. Long-form factuality in large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Duygu Nur Yaldiz, Yavuz Faruk Bakman, Baturalp Buyukates, Chenyang Tao, Anil Ramakrishna, Dimitrios Dimitriadis, Jieyu Zhao, and Salman Avestimehr. 2025. Do not design, learn: A trainable scoring function for uncertainty estimation in generative LLMs. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 691–713, Albuquerque, New Mexico. Association for Computational Linguistics.

Caiqi Zhang, Fangyu Liu, Marco Basaldella, and Nigel Collier. 2024. LUQ: Long-text uncertainty quantification for LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5244–5262, Miami, Florida, USA. Association for Computational Linguistics.

Yukun Zhao, Lingyong Yan, Weiwei Sun, Guoliang Xing, Chong Meng, Shuaiqiang Wang, Zhicong Cheng, Zhaochun Ren, and Dawei Yin. 2024. Knowing what LLMs DO NOT know: A simple yet effective self-detection method. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7051–7063, Mexico City, Mexico. Association for Computational Linguistics.

A Datasets Statistics

TriviaQA contains question-answer pairs authored by trivia enthusiasts. Among the 17.2k samples in the test split, we use a random subset of 1000 samples in our evaluations. GSM8K is composed of grade school math word problems. It contains 1.32k samples in the test set. Similar to GSM8K, we use a subset of 1000 samples in our experiments. Lastly, FactScore-Bio contains biography queries about people from Wikipedia. We used a random subset of 50 questions from this dataset. After generation decomposition, the total number of claims is 1290 for GPT-4o-mini and 1764 for Llama-3-8B. We provide sample questions from each dataset in Table 3.

B Additional Code Snippets

Below is an example illustrating how to calibrate a set of truth methods on the TriviaQA dataset:

Listing 4: Calibrating multiple truth methods using Isotonic Regression.

```
# Assign a calibrator to each method
for truth_method in truth_methods:
    truth_method.set_normalizer(ttlm.normalizers.
        IsotonicRegression())

# Calibrate using labeled evaluation data
calib_results = ttlm.calibrate_truth_method(
    dataset=’trivia_qa’,
    model=model, tokenizer=tokenizer,
    truth_methods=truth_methods,
    correctness_evaluator=model_judge,
    size_of_data=1000, max_new_tokens=64)
```

We provide a code sample below that evaluates truth methods in long-form generation setting:

Listing 5: Evaluation on long-form generation.

```
#Define claim evaluator
safe = LFG.ClaimEvaluator(
    rater=’gpt-4o-mini’,
    tokenizer = None,
    max_steps = 5,
    max_retries = 10,
    num_searches = 3)

# Use built-in or custom datasets for evaluation
results = LFG.evaluate_truth_method_long_form(
    dataset=’longfact_objects’,
    model=model, tokenizer=tokenizer,
    sample_level_eval_metrics=[’f1’],
    dataset_level_eval_metrics=[’auroc’, ’prr’],
    decomp_method=decomposition_method,
    claim_check_methods=[qa_generation],
    claim_evaluator = safe)
```

C Generation Decomposition

We provide a sample output of our decomposition process in Table 4. The model used for decomposition is GPT-4o-mini and the output structure is forced by instructor[4].

[4] https://python.useinstructor.com/

	Question	Ground Truth
TriviaQA	David Lloyd George was British Prime Minister during the reign of which monarch?	King George V
	How many symphonies did Jean Sibelius compose?	Seven
	The capital of Brazil was moved from Rio de Janeiro to the purpose-built capital city of Brasilia in what year?	1960
GSM8K	Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May?	72
	Julie is reading a 120-page book. Yesterday, she was able to read 12 pages and today, she read twice as many pages as yesterday. If she wants to read half of the remaining pages tomorrow, how many pages should she read?	42
	Mr. Sam shared a certain amount of money between his two sons, Ken and Tony. If Ken got $1750, and Tony got twice as much as Ken, how much was the money shared?	5250
FactScore-Bio	Tell me a bio of Vaira Vīķe-Freiberga.	-
	Tell me a bio of Ji Sung.	-
	Tell me a bio of Baltasar Corrada del Río.	-
	Tell me a bio of Henry Santos.	-

Table 3: Data samples from the datasets we use in our evaluations: TriviaQA, GSM8K, and FactScore-Bio

Question: Tell me a bio of Andrew Shue.

Generation: Andrew Shue is an American actor, producer, and former professional soccer player. He was born on January 20, 1967, in South Orange, New Jersey, USA.

Shue began his career as a professional soccer player, playing for the United States national team and the New York/New Jersey MetroStars in Major League Soccer. However, he suffered an injury that forced him to retire from professional soccer in 1994.

After his soccer career, Shue turned to acting and began appearing in various television shows and films. Some of his notable roles include:

* "The Breakfast Club" (1985) - Shue played the role of Andrew Clark, a popular high school student.
* "Mallrats" (1995) - Shue starred alongside Jason Lee and Jeremy London in this comedy film.
* "The Adventures of Pete & Pete" (1992-1996) - Shue played the role of C.J. Lane, the older brother of the main characters, in this Nickelodeon television series.
* "Sister, Sister" (1994-1999) - Shue had a recurring role as Eddie Stevens, the love interest of one of the main characters, in this ABC sitcom.

Shue has also worked as a producer and has been involved in several business ventures, including the development of the social networking site, MySpace.

Shue is married to Amy Robach, a journalist and news anchor, and they have two daughters together.

Claims:

[‘Andrew Shue is an American actor.’,
‘Andrew Shue is a producer.’,
‘Andrew Shue is a former professional soccer player.’,
‘Andrew Shue was born on January 20, 1967.’,
‘Andrew Shue was born in South Orange, New Jersey, USA.’,
‘Andrew Shue began his career as a professional soccer player.’,
‘Andrew Shue played for the United States national soccer team.’,
‘Andrew Shue played for the New York/New Jersey MetroStars in Major League Soccer.’,
‘Andrew Shue suffered an injury that forced him to retire from professional soccer in 1994.’,
‘Andrew Shue turned to acting after his soccer career.’,
‘Andrew Shue began appearing in various television shows.’,
‘Andrew Shue began appearing in films.’,
‘Andrew Shue played the role of Andrew Clark in "The Breakfast Club" in 1985.’,
‘Andrew Shue starred in "Mallrats" in 1995.’,
‘Andrew Shue starred alongside Jason Lee in "Mallrats".’,
‘Andrew Shue starred alongside Jeremy London in "Mallrats".’,
‘From 1992 to 1996, Andrew Shue played the role of C.J. Lane in "The Adventures of Pete & Pete".’,
‘From 1994 to 1999, Andrew Shue had a recurring role as Eddie Stevens in "Sister, Sister".’,
‘Andrew Shue has worked as a producer.’,
‘Andrew Shue has been involved in several business ventures.’,
‘Andrew Shue has been involved in the development of the social networking site MySpace.’,
‘Andrew Shue is married to Amy Robach.’,
‘Amy Robach is a journalist.’,
‘Amy Robach is a news anchor.’,
‘Andrew Shue and Amy Robach have two daughters together.’]

Table 4: Output of long-text decomposition. The question is from FactScore-Bio and the model used to generate the answer is LLaMa-3-8B.