Unifying Corroborative and Contributive Attributions in Large Language Models

Theodora Worledge* **Judy Hanwen Shen*** **Nicole Meister** **Caleb Winston** **Carlos Guestrin**[1,2]

Stanford University

***Abstract*—As businesses, products, and services spring up around large language models, the trustworthiness of these models hinges on the verifiability of their outputs. However, methods for explaining language model outputs largely fall across two distinct fields of study which both use the term "attribution" to refer to entirely separate techniques: citation generation and training data attribution. In many modern applications, such as legal document generation and medical question answering, both types of attributions are important. In this work, we argue for and present a unified framework of large language model attributions. We show how existing methods of different types of attribution fall under the unified framework. We also use the framework to discuss real-world use cases where one or both types of attributions are required. We believe that this unified framework will guide the use case driven development of systems that leverage both types of attribution, as well as the standardization of their evaluation.**

***Index Terms*—train data attribution, citation generation, large language model attribution, explainability**

I. Introduction

The rapid rise of large language models (LLMs) has been accompanied by a plethora of concerns surrounding the trustworthiness and safety of the LLM outputs. For example, these models can "hallucinate" or fabricate information in response to straightforward prompts (1). Beyond simply verifying that generated content can be trusted, knowing the source from which the output was generated is also crucial in many applications. In fact, Bommasani et. al. (2) highlight that "*Source tracing is vital for attributing ethical and legal responsibility for experienced harm, though attribution will require novel technical research*". The ubiquitous usage of LLMs in applied settings motivates the development of explanations that provide *both* sources that verify the model output and training sources that are influential in the generation of the output. Unfortunately, attributing an LLM output to sources has been mostly studied in two disjoint fields: citation generation and training data attribution (TDA). Verifying the correctness of model outputs, generally situated in the natural language processing and information retrieval communities, includes several different tasks such as fact-checking (3; 4), knowledge retrieval (5; 6; 7), attributed question answering (8), and verifiability in language generation (9). Training data attribution, generally situated in the core machine learning community, encompasses a variety of techniques to explain model behavior such as influence functions (10), data simulators (11), and data models (12). Meanwhile, the term "*attributions*" is used in both fields. When contemplating the two types of attributions, we can think of the former as external validity, which verifies that the output is correct according to external knowledge, and the latter as a certification of internal validity, which provides the source of the generated content. We can easily imagine applications where both types of validity are important for understanding LLM outputs. For instance, a potential criteria to use for identifying a case of model memorization is for a training source to exactly match the model output while also being highly influential in the generation of the output.

In this work, we argue for a unifying perspective of the citation generation and TDA forms of attribution, which we call **corroborative** and **contributive** attributions, respectively. We precisely define each type of attribution and discuss different properties that are desirable in different scenarios. Our work provides a first step towards a flexible, but well-defined notion of language attributions to encourage the development and evaluation of attribution systems capable of providing rich attributions of both types.

A. Our Contributions

1) We present an interaction model for LLM attributions that unifies corroborative and contributive attributions through their common components (Section IV).
2) To complete our unified framework, we outline properties relevant to both types of attributions (Section V).
3) We discuss existing implementations of corroborative and contributive attributions (Section VI).
4) We outline scenarios where attributions are important and discuss their desired properties (Sections VII, VIII).
5) We provide directions for future work on attributions (Section IX).

II. Motivation: The Necessity of a Unified Perspective

We argue for the study of LLM attributions through a unified perspective of corroborative and contributive attributions. First, we describe the limitations of the current fragmented approach to attributions and then we summarize the case for unification.

A. Gaps in existing approach to language model attributions

a) ***Misalignment between TDA methods and their use cases:*** Most training data attribution (TDA) papers present

*Equal contribution. Correspondence to: {worledge, jhshen}@stanford.edu
[1]Chan Zuckerberg Biohub
[2]Stanford Institute for Human-Centered Artificial Intelligence (HAI)

their methods as standalone solutions for motivating use cases such as identifying mislabeled data points (10; 13; 14; 15; 16), debugging domain mismatch (10), and understanding model behavior (17). While discriminative model outputs may be adequately explained by contributive attributions, explanations for generative model outputs may require the additional notion of corroboration. In the setting of language models, TDA methods may not be a comprehensive solution; training sources that are irrelevant to the content of a model output may be flagged as influential by TDA methods (17). This is undesirable because the semantic meaning of a flagged training source can indicate its importance in generating the semantic meaning of the output. For instance, when searching for misleading training sources in a Question Answering (QA) language model, it is important to understand which of the sources flagged by TDA methods corroborate the misinformation in the output. This is also the case in other practical applications, such as debugging toxic outputs. Without carefully considering the types of attribution needed in generative language models, we risk investing in methods that, while establishing essential foundations, may not align with practical use.

b) ***Citation generation methods do not explain model behavior****:* Corroborative methods (e.g., fact checking (4), citation generation (5)) are not designed to explain model behavior. For example, the verifying the truthfulness of outputted facts using sources from an external corpus does little to explain why the model generated such an output. When outputted facts are found to be incorrect, there is limited recourse for correcting model behavior. Thus, corroborative attributions alone cannot address all the challenges of explaining the outputs of language models.

c) ***Emergent usage of language models require a richer notion of attributions****:* The emerging use of LLMs in domains such as health care and law involves tasks such as document generation and domain-specific QA that require both explanations of whether the output is correct and where the output came from. As an example, in the legal domain, different products based on LLMs such as legal QA, immigration case document generation, and document summarization are currently under development.[1] In this setting, *corroborative* attributions are important to ensure that a generated legal document follows local laws. The sources for such corroborative attributions need not be in the training data. Simultaneously, *contributive* attributions are important for understanding the training documents from which the generated legal document is borrowing concepts. In the legal setting, context and subtle changes in wording matter (2).

B. Motivating a unified framework of attributions

Developing a standardized language to describe different types of attribution will improve the (1) *clarity* and (2) *simplicity* of scholarly discussion around attributions. Furthermore, identifying the common components of all attributions provides (3) *modularity* for improving individual components and better (4) *reproducibility* of results. Looking ahead to future work, a unified perspective motivates the (5) *hybrid development* of both corroborative and contributive attributions.

a) ***"Attribution" is an overloaded, ambiguous term****:* The term "attribution" is overloaded in machine learning literature. Moreover, recent works have attempted to provide both types of attribution for language models under the vague umbrella term of “attributions” (8; 17; 18). While existing work recognizes the importance of both corroborative and contributive attribution (19), comparing these two notions is difficult without precisely delineating between them while also acknowledging their similarities. A unified perspective of both types of attributions improves the *clarity* of technical progress on attributions.

b) ***Attribution methods exist concurrently in disjoint fields****:* The two dominant interpretations of attributions for language model outputs come from the natural language processing (NLP) and information retrieval (IR) communities, as well as the core ML explainability community. In NLP and IR literature, attributing a model output to a source generally refers to identifying a source that corroborates the output (3; 4; 7; 8; 9; 20). We refer to this as *corroborative attribution*. This differs from TDA work, where attributing a model output to a source refers to identifying a training source that highly influenced the model to produce that output (10; 11; 18; 21). We refer to this as *contributive attribution*. While there has been work on contributive attributions in the NLP venues (22) and corroborative attributions in ML venues (23), there remains little overlap between the two communities. To the best of our knowledge, there is no established framework that unifies these different types of attributions. Furthermore, methods to achieve both types of attribution and metrics to evaluate them have been developed separately. Our goal is to introduce *simplicity* in understanding the vast landscape of prior work by creating a shared language to discuss attribution methods across different tasks.

c) ***Attributions have common components****:* Despite these two types of attribution being studied in different fields, there are commonalities in system components, properties, metrics, and evaluation datasets. For example, fact-checking using corroborative attributions has significant overlap with fact-tracing using contributive attributions, in terms of metrics and evaluation datasets (22). Defining the shared components of different types of attributions introduces *modularity* that better enables the improvement of individual components of attribution systems. Furthermore, precise definitions of properties shared across different attributions allow for better *reproducibility* in implementations of attribution systems.

d) ***A unifying perspective enables the development of richer attribution systems****:* Because both notions of attribution are relevant to use cases that improve the safety and reliability of language models as information providers, both are often simultaneously relevant in application settings. There are real-world use cases of attribution that require careful reasoning and differentiating between these two interpretations; some use cases even require both notions of attribution. These use

[1] Y-Combinator companies in this area include Casehopper, Lexiter.ai, DocSum.ai, and Atla AI.

cases should motivate the *hybrid development* of methods that provide both citation and TDA for LLM outputs. Furthermore, methods used in one type of attribution may be leveraged to develop other types of attributions.

III. Related Work

The majority of prior work has focused on corroborative and contributive attributions separately. Works that have considered both types of attribution in the same setting often do so for specific case studies or experiments without attempting to provide a conceptual unification. This section discusses existing attribution frameworks, as well as works that simultaneously employ notions of corroborative and contributive attributions.

a) ***Corroborative attribution frameworks****:* Previous work has proposed and leveraged frameworks for attributions that identify supporting sources for model outputs. Notably, (9) define a specific notion of corroborative attribution; a model output is transformed into an interpretable standalone proposition s, which is then attributed to a source P if it passes the human intuitive test that "According to P, s". Their *attributable to identified sources* (AIS) evaluation framework evaluates both steps of this definition with human annotators who first evaluate the interpretability of the model output and then whether it satisfies the aforementioned intuitive test for a particular source. Bohnet et. al. (8) applies the AIS framework to the QA setting. Gao et. al. (6) extends the AIS framework to evaluating LLMs that output citations alongside standard text generations. Another line of work focuses on building and using automated AIS evaluations (24; 25). In contrast to prior work, we generalize the definition of corroborative attribution beyond the notion of an "intuitive test" and construct a framework to unify these attributions with contributive attributions.

b) ***Contributive attribution frameworks****:* Existing TDA work has revealed a common framework for contributive attributions. This shared framework, explicitly defined as data attribution in (18), specifies that given a model, list of training data instances, and input, a data attribution method is a function that produces a scalar score for each training instance, indicating the importance of that training instance to the model's output generated from the input. Several lines of work fit under this framework, including influence functions, which make great efforts to scale implementations in the face of significant computational requirements (10; 15; 16; 17; 18). Surveys summarizing this area include broad categorizations across gradient-based and retraining-based methods (26) and language-specific summaries (27).

c) ***Shared settings for corroborative and contributive attributions****:* Even without a shared framework, attributions that are simultaneously corroborative and contributive have naturally appeared. The first of these settings is *fact tracing* (22), which recovers the training sources that cause a language model to generate a particular fact. (22) propose FTRACE-TREx, a dataset and evaluation framework with the explicit goal of identifying corroborative training sources using contributive attribution methods. (18) also uses FTRACE-TREx as a benchmark for different TDA methods. Another shared setting of corroborative and contributive attributions is the TF-IDF filtering employed in (17). Here, TF-IDF scores (28) are used to filter the training data to a manageable number of sources for influence estimation. While the ultimate objective of this heuristic in (17) is to overcome the bottleneck of training source gradient calculations, the TF-IDF filtering ensures that all of the sources examined are semantically related, which we consider a corroborative notion, to the model input. As the models and training dataset sizes of LLMs continue growing larger, filtering strategies built on notions of corroboration may become the norm. Lastly, (19) discuss attributions to non-parametric content, meaning corroborative sources, and attributions to parametric content, meaning contributive sources. While it is perhaps the closest existing work to ours in that it makes explicit the value of both corroborative and contributive attributions, (19) largely focuses on roadblocks to practical implementations and pitfalls of attributions in LLMs; a formal unifying framework for the different types of attribution is not proposed.

IV. Formal Problem Statement

A. Interaction Model

To frame our discussion of attributions for LLMs, we first define the relevant components of an attribution. We build upon the *Attributable to Identified Sources* definition introduced by Rashkin et. al. (9) to introduce a general framework for different types of attributions. We define 6 high-level components of the attribution system interaction: the input, model, output, attributable unit, attribution domain and evaluator that allow us to construct an attribution set. As a running example throughout the paper, we consider the use case of attributions for QA in which a model provides a short-form output for a given input.

a) ***Input****:* The input is the query provided to the model (x). Following the requirements for input interpretability proposed in (9), we assume that x contains the wall-clock time at which it was used to query the model. We consider a variety of different input queries including knowledge queries and generative queries. Knowledge queries are questions that can be answered with the correct piece of information; this is analogous to the QA task. Our scope includes both *Open-book QA* and *Closed-book QA* (29). Generative queries may have many different answers but may nevertheless require attribution. For example: "Plan a fun weekend in San Francisco" and "Write me a Python program to approximate pi" are both generative queries that require verification before a model can be trusted. While we do not directly consider other interactive settings where there are multiple inputs (e.g., information-seeking dialog (30) and social-oriented settings (e.g., chit-chat dialog), these are important future directions in which our framework for attribution should extend.

Example: `"What is the diameter of the moon?"`

b) ***Output****:* The output (y) is the response of a language model to the input (x).

Example: `"3,475 kilometers"` [2]

[2] https://nightsky.jpl.nasa.gov/club/attachments/Fun_Facts_About_the_Moon.pdf

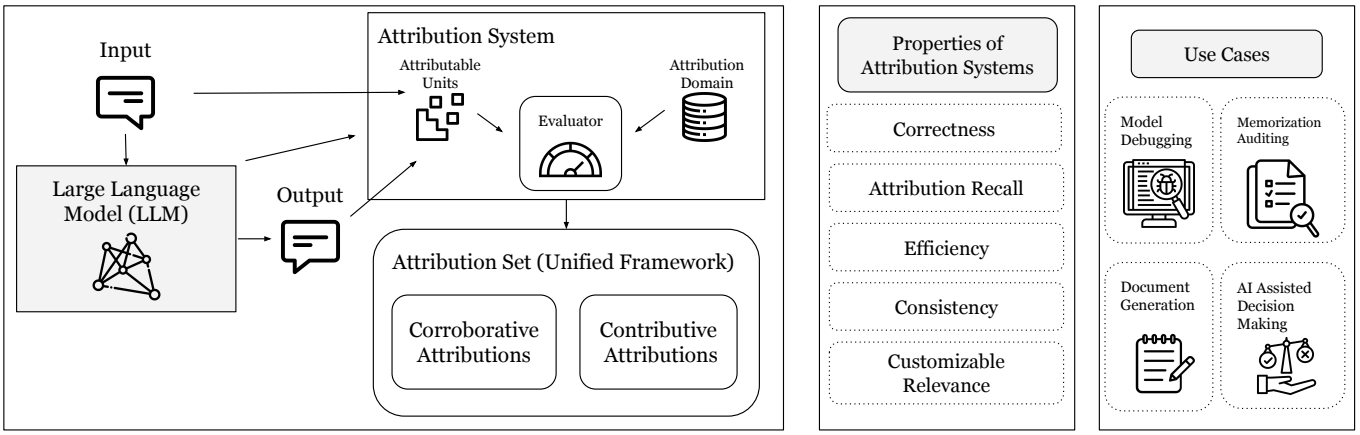

Fig. 1. Overview of our proposed unified framework for large language model attributions. We include tasks that require both contributive and corroborative attributions and properties that apply to both types of attributions.

*c) **Model**:* The base language model M takes an input x and generates the output y. We note that in practice, some models jointly output attributions with the answer y. However, when defining an attribution under our framework, we consider the output generation and attribution generation separately, even if they are generated by the same model. Therefore, for inputs $x \in \mathcal{X}$ and outputs $y \in \mathcal{Y}$, we define *the model* as $M : \mathcal{X} \rightarrow \mathcal{Y}$.

Example: LLM

*d) **Attributable Unit**:* In some cases, the full output is used to create an attribution. However, in other cases, a sentence may contain many clauses that need to be independently attributed to achieve the desired level of granularity for the attribution. We define an attributable unit $z = (x, y, i, j)$ where i and j are the beginning and end indices of tokens in y which require attribution. We define the set of all the attributable units as $Z = z_1, ..., z_n$ for x and y.

Example attributable set: `[("What is the diameter of the moon?", "3,475 kilometers", 0, 15)]`

*e) **Attribution Domain**:* A crucial component of our attribution framework is the domain from which sources (i.e. $s_1, ..., s_m \in D$) for attribution are drawn; we call this the *attribution domain* D. There are different promises and limitations when the attributions are drawn from the training data compared to other data not necessarily included in the training. In the practical application and deployment of language models, there are even more domains such as in-context data and fine-tuning data.[3]

Example: LLM training data

*f) **Evaluator**:* Each attribution is identified with an evaluation function we call an *evaluator*. Different evaluators lead to different types of attribution. Given an attributable unit $z \in \mathcal{Z}$ and source $s \in \mathcal{D}$, an evaluator $v : \mathcal{Z} \times \mathcal{D} \rightarrow \mathbb{R}$ provides a score that represents the extent to which the given source is an attribution for the attributative unit. In some cases, this value is binary and in others it is continuous. For instance, exact match (EM) is an example of a binary evaluator, which is defined as:

$$v_{\text{EM}}(z, s) = \begin{cases} 1 & \text{If } y[i:j] \text{ exists word-for-word within } s, \\ 0 & \text{otherwise.} \end{cases}$$

Implementations of v are denoted as $\hat{v}$. An implemented evaluator $\hat{v}$ is not infallible, making it important to evaluate the evaluator against other evaluators on common ground, i.e., potentially using another implementation of the evaluator to compute relevant metrics (see Section V-B). Past work has used human annotators for $\hat{v}$ (9; 20; 31), but the high cost in time and resources of human evaluation has motivated model-based implementations of $\hat{v}$ (4).

Example: If seeking a corroborative attribution, we can use the textual entailment evaluator, v_{TE}, as defined in Definition 4. If seeking a contributive evaluator, we can use the counterfactual textual entailment evaluator, v^{M}_{CTE}, as defined in Definition 5.

B. Attribution Sets

Having defined the different components of an attribution system, we now present a definition for an attribution.

Definition 1. *[Attribution Set] Given an attributable set Z, source domain D, evaluator v, and evaluator cutoff $\alpha \in \mathbb{R}$, an attribution set $\mathcal{A}$ is the following set of attributions, or pairs of attributable units and sources:*

$$\mathcal{A}(Z, D, v, \alpha) = \{(z, s) \mid z \in Z, s \in D, v(z, s) \geq \alpha\}$$

We present this definition as a class of explanations for language model outputs. The type of attributions provided in the set depends primarily on the evaluator v and attribution domain D. Prior work from (9) proposes the AIS framework where the evaluator v seeks to satisfy the intuitive test "According to s, z" for some source s and sentence z. Our definition differs from AIS in several ways. Significantly, the evaluator

[3]While we leave the complexities of these domains for future work, we discuss in-context data as an attribution domain in Appendix A.

v of our framework is not restricted to the intuitive test and the attributable unit z of our framework is not restricted to sentence-level explicatures. The flexibility of our framework is important in unifying different approaches to attribution.

C. Attribution Sets with Customizable Source Relevance

Definition 1 of an attribution set considers all sources that satisfy the evaluator cutoff for a given attributable unit as equal in value. Sometimes, however, it is important to value certain sources over others, even if all are valid attributions. Different use-cases demand different notions of relevance; among others, the field of information retrieval has studied multiple manifestations of relevance (32). To accommodate for this, our definition of a relevance function below allows for custom orders of priority among sources.

Definition 2. *[Relevance Function] Given attributable units $z \in Z$, attribution domain sources $s \in D$, evaluator v, and evaluator cutoff $\alpha \in \mathbb{R}$, a relevance function is defined as $\phi : Z \times D \to \mathbb{R} \in [0, 1]$ such that if $v(z, s_1) \geq \alpha$, $v(z, s_2) \geq \alpha$, and $\phi(z, s_1) > \phi(z, s_2)$, then s_1 is considered to be a better attribution for z than is s_2.*

Adding this additional component of source relevance to an attribution set allows for an ordering of sources within the source domain. While this notion of relevance is not integral to an attribution, it is particularly useful for certain applications. We build off of an attribution set to define the following:

Definition 3 (r-Relevant Attribution Set)**.** *Given an attributable set Z, source domain D, evaluator v, evaluator cutoff $\alpha \in \mathbb{R}$, relevance function ϕ, and relevance threshold $r \in \mathbb{R}$, an r-relevant attribution set $\mathcal{A}$ is the following set of attributions, or pairs of attributable units and sources:*

$$\mathcal{A}(Z, D, v, \alpha, \phi, r) = \{(z, s) \mid z \in Z, s \in D, v(z, s) \geq \alpha, \phi(z, s) \geq r\}$$

Note that the relevance of a source document for an attribution is a function of the attributable unit. Including a relevance threshold in an attribution set is a way to place priority on certain sources within the attribution domain.

V. Properties of Attributions

The central question of *why did a language model provide this answer?* can be answered in many different ways. We present two types of attributions that correspond to different ways of explaining a model output. Furthermore, we build on existing properties of explanations of LLM outputs to define properties that are relevant to language model attributions.

A. Corroborative and Contributive Attributions

a) ***Corroborative Attributions****:* A vast literature exists around corroborative attributions. Prior works refer to these as *citations* in open-domain QA and retrieval settings (33; 34).

An attribution set (Definition 1) is corroborative if its evaluator is corroborative. Corroborative evaluators compare the information content between an attributable unit and a source drawn from the attribution domain. Formally, we define a corroborative evaluator as follows:

Definition 4. ***Corroborative Evaluator****. Let $s \in D$ be a source in the attribution domain and $z = (x, y, i, j) \in Z$ be an attributable unit of the input-output pair. A corroborative evaluator is a binary evaluator such that:*

$$v_{corr}(z, s) = \begin{cases} 1 & \textit{If } s \textit{ corroborates } z, \\ 0 & \textit{otherwise.} \end{cases}$$

Moreover, v_{corr} is a class of different possible evaluators where "corroborate" can have different meanings. Three common corroborative evaluators are:

- ***Exact Match****: v_{EM} verifies whether there is an exact match between: $y[i : j]$ and a clause in source s.*
- ***Valid Paraphrase****: v_{VP} verifies that $y[i : j]$ written as a declarative sentence in the context of x, y is a valid paraphrase of content in s; i.e., the declarative sentence is a rewriting of content in s that preserves its truth conditions.*
- ***Textual Entailment****: v_{TE} verifies that $y[i : j]$, in the context of x, y, logically follows from the source s.*[4]

The study of linguistics has long recognized the inherent *fuzziness* of natural language and so asserts that logical operations are relaxed to *approximate reasoning* when applied to natural language (35). Therefore, the logical operations involved in the valid paraphrase and textual entailment evaluators are actually instances of *approximate reasoning*. In practice, the textual entailment evaluator is either implemented through human reasoning or through automated systems capable of natural language inference (NLI), as discussed further in Section VI.

For the valid paraphrase and textual entailment evaluators, the context provided by the original input x and the rest of the output $y \setminus y[i : j]$ may be important. To this end, the spans $y[i : j]$ of each attributable unit can be chosen to correspond to sentence-level (9) or clause-level *explicatures* (see Appendix B). Rewriting a span as an explicature allows the span $y[i : j]$ to be interpreted in the context of x and y. In particular, attributable units corresponding to clause-level explicatures within one sentence of the output allow the sentence to be corroborated through more than one source, rather than requiring a single source to corroborate everything in the sentence. In practice, the attributable set is already predefined in many existing tasks and benchmarks (34).

In general, the attribution domain of a corroborative attribution may contain any document regardless of whether it was used to train the model or not. The corroborative attribution set for a model output is independent of the model itself; if another model were to produce the same output, the original corroborative attribution set would still be applicable.

[4]We assume $s \in D$ has been chosen such that there are no degenerate cases where s contradicts itself, as this would permit an attribution to any z.

*b) **Contributive Attributions**:* A contributive attribution set is an attribution set (Definition 1) that draws from an attribution domain D that is restricted to training sources and relies upon a contributive evaluator. A contributive evaluator is defined as:

Definition 5. ***Contributive Evaluator**. Let $s \in D$ be a source in the attribution domain and $z = (x, y, i, j)$ be an attributable unit. A contributive evaluator for model M is an evaluator such that:*

$$v_{cont}^{M}(z, s) \in [0, 1],$$

where $v_{cont}^{M}(z, s)$ quantifies how important source s is to M (trained on D) evaluated on the attributable unit z. The counterfactual we compare against is z evaluated on a M trained without s (i.e., trained on $D \setminus s$).

- ***Counterfactual contribution to loss (CCL)**: v_{CCL}^{M} quantifies the extent to which the loss on y for input x would be different under the counterfactual model $M_{D \setminus s}$, compared to under M_D.*
- ***Counterfactual contribution to the output (CCO)**: Let $y' = M_{D \setminus s}(x)$ be the counterfactual output of a model trained without s. Then,*

$$v_{CCO}^{M}(z, s) = \begin{cases} 1 & \text{If } v_{corr}(z, y') = 0, \\ 0 & \text{otherwise.} \end{cases}$$

Note that v_{corr} is used to indicate whether z is corroborated by the counterfactual model output, y', rather than by a source. Moreover, v_{cont}^{M} is a class of different possible evaluators where "contribute" takes on different meanings with different v_{corr}. Any corroborative evaluator, including those mentioned in Definition 4, can be used to construct a contributive evaluator. We highlight two examples of counterfactual output comparison evaluators:

  - ***Counterfactual Exact Match**: v_{CEM}^{M} relies on the corroborative exact match evaluator v_{EM} to indicate whether $y[i : j]$ remains the same or changes, had source s not been present in the training data.*
  - ***Counterfactual Textual Entailment**: v_{CTE}^{M} relies on the corroborative textual entailment evaluator v_{TE} to indicate whether claims in $y[i : j]$ in the context of x and y remain the same or change, had source s not been present in the training data.*

A shortcoming of the CCL evaluator is that loss does not convey the semantic content of the output. To address this limitation, we introduce the CCO evaluator.[5] Keeping with the running example of querying a model with "What is the diameter of the moon?" and it generating the response, "3,475 kilometers", we can imagine using the counterfactual textual entailment CCO evaluator. In this case, a source s would be deemed contributive if its removal from the training set would result in a counterfactual model that outputs "At least 3,000 kilometers" in response to the same input, but not if it outputs "3,475,000 meters". This differs from the CCL evaluator, which identifies a training source as contributive if its removal leads to a counterfactual model that has significantly different loss on the output, regardless of how the semantic meaning of the counterfactual output differs, if at all. We advise that this novel concept of CCO evaluators be a focus of future work on contributive attributions for LLMs.

We note that the CCL evaluator follows standard machine learning methodology more closely than the CCO evaluator, because it operates on the loss, rather than on the discrete output space of language. Furthermore, the CCL evaluator is compatible with non-deterministic decoding strategies, unlike the CCO evaluator, which, in its current form, is limited to deterministic decoding strategies. This is because the CCL evaluator keeps y fixed under the counterfactual model, whereas the CCO evaluator generates a new y' from the counterfactual model, making it difficult to disentangle any stochasticity in the decoding process from the contribution of s. For instance, if a low probability completion happens to be sampled for y, then y' is likely to change, even if s was not removed. Accordingly, prior TDA work implements the CCL evaluator (see Section VI-B).

B. Properties and Metrics of Attribution Sets

Depending on the application of the LLM, different properties of attribution sets may be desirable. Crucially, these desiderata may be different from those of general machine learning explanation methods.[6] While properties are high-level qualities that are desirable in an LLM attribution, metrics are specific methods to measure these properties. A single property can be measured by many different metrics. While we provide a few metrics for each property in Table I, future work may use different metrics for these properties.

*a) **Correctness**:* The most ubiquitous measure of attribution sets in current work is whether an attribution set is correct. To interrogate properties of correctness, some notion of ground truth, often in the form of an oracle evaluator v, is required to properly score each attribution.

- **Attribution validity:** For each attribution in an attribution set, the notion of validity captures how correct the attribution is relative to a ground truth evaluator. Corroborative attributions generated by various systems have been evaluated for validity using v_{TE} implemented via human reasoning (8; 9). Contributive attributions have been evaluated for validity using leave-one-out retraining (10) and the proximal Bregman response function (17).
- **Coverage:** An attribution set $\mathcal{A}$ with evaluator cutoff α has perfect coverage if $\forall\ z \in Z\ \exists\ (z, s) \in \mathcal{A}, v(z, s) \geq \alpha$. Previous work has referred to coverage as attribution recall (20). One way to measure coverage is to calculate the

[5]The gap between machine learning objectives and practical understanding has been highlighted in other areas. For instance, a critique of mechanistic explanations of model behavior, which solely rely on the inner workings of the model, is that they are not interpretable to humans (36). Similar to how mechanistic explanations can be enriched by practical explanations that are meaningful to applied practitioners, existing loss based attributions can be enriched by attributions for counterfactual changes in the output semantics.

[6]https://christophm.github.io/interpretable-ml-book/properties.html

proportion of attributable units in Z with a valid attribution under an oracle evaluator v included in $\mathcal{A}$ (20).

- **Attribution precision:** Another way to measure attribution set correctness is precision. An implemented attribution set $\hat{\mathcal{A}}$ with evaluator cutoff α is precise if $v(z, s) \geq \alpha \ \forall \ (z, s) \in \hat{\mathcal{A}}$. By definition, an attribution set $\mathcal{A}$ has perfect precision. However, this is an important property when evaluating implementations of attribution systems, where the components analogous to attribution evaluators are imperfect. One way to measure the precision of an attribution set is to calculate the proportion of valid attributions under an oracle evaluator v (20).

*b) **Attribution Recall**:* Let S' be the set of all documents that provide attribution for a given z (i.e., $S'(z) = \{s | s \in D, v(z, s) \geq \alpha\}$). The attribution set $\mathcal{A}$ has perfect recall for z if $\forall \ s \in S'(z), (z, s) \in \mathcal{A}$. One way to measure the recall of an attribution set for z is to calculate the proportion of sources from the attribution domain that fulfill $v(z, s) \geq \alpha$ that are actually included in the attribution set. This is a measurement of the sources that can attribute one specific z, which differs from coverage which focuses on whether all $z \in Z$ is attributed.

In the corroborative setting, there may be many sources that can provide an attribution for z. Attribution recall might be important when an attributable unit z requires multiple sources to validate. For example, facts about the efficacy of certain drugs might require all relevant studies to be included rather than just a single source. In the contributive attribution setting, many training documents may have been influential in generating an output. Having perfect attribution recall is relevant when using attribution to assign credit to training data authors and for model debugging, where all sources need to be identified. Measuring attribution recall has appeared in prior work (14) as a measurement of the fraction of artificially mislabeled examples that were successfully identified through gradient tracing for TDA.

*c) r-**Relevancy**:* As explained in definition 3, an attribution set is r-relevant if all the sources in the attribution set meet the threshold of r under some relevancy function, ϕ. r-Relevancy is an important property because some applications find certain sources in the attribution domain to be more useful than others. This is the case in the setting of corroborative attributions for fact-checking, where trustworthy sources are more relevant than questionable sources. This is also the case in the setting of corroborative attributions for generating citations for written reports, where primary sources tend to be more relevant than secondary or other derivative sources. Although motivated from an efficiency standpoint, (17) in effect implements r-relevant contributive attribution sets with TF-IDF filtering as a relevancy function; only sources that are high in TF-IDF similarity to the input are considered for the attribution set. A metric to measure the r-relevancy of an attribution set is the proportion of attributed sources that meet the relevancy threshold r.

Properties	Metrics
Correctness	Validity (8; 9) Coverage (20; 25) Attribution Precision (20)
Attribution Recall	Mislabeled example identification (14)
Relevancy	Proportion of attribution set that is r-relevant
Consistency/Replicability	Attribution set distance
Efficiency	Training time (18; 26) Inference time (26; 37) Training memory requirements (26) Inference memory requirements (26)

TABLE I
PROPERTIES OF ATTRIBUTION SETS AND SYSTEMS. DIFFERENT METRICS HAVE BEEN PROPOSED BY PRIOR LITERATURE IN MEASURING EACH OF THESE PROPERTIES.

C. Properties and Metrics of Attribution Systems

Properties of attribution sets are inherent to a single attribution set. However, some properties are instead functions of the implemented system that generates the attribution sets in the first place. We discuss two such properties.

*a) **Consistency**:* An attribution system is considered consistent if, for similar inputs and outputs, the generated attribution sets are similar. For a fixed attributible set Z, attribution domain D, evaluator v, and evaluator cutoff α, an attribution system is ϵ-stable over sources of randomness in the system if for $\mathcal{A}$ and $\mathcal{A}'$ sampled from different executions, $\mathbb{E}[d(\mathcal{A}, \mathcal{A}')] \leq \epsilon$, where d is some distance metric defined over input-output pairs and over attribution sets respectively (e.g., d could be the Jaccard distance over sources' indicator functions).

This property is particularly important when decisions based on LLM outputs need to be documented as justification. For corroborative attributions, a legal service scenario may require documentation of sources for advice provided to customers. For contributive attributions, an authorship compensation scenario would require attribution consistency to fairly determine payments to creators. In both cases, there is value in replicating the same attribution set at a later time with the same inputs.

Prior work highlighting the shortcomings of contributive methods (e.g., influence functions) demonstrates increased variance in influence estimates for deeper models; this would preclude consistency unless influence is estimated using an average across multiple runs (38). Similarly, averaging gradients across checkpoints during training might lead to inconsistent estimates of influence estimation because the ordering of examples has a significant impact on observed influence (39). However, consistency has not been directly measured in prior work for contributive or corroborative attributions.

*b) **Efficiency**:* Efficiency describes the time and space complexity required by an implementation of an attribution system in generating an attribution set for a given attribution domain, input, and output. Prior works on large language models examine both training and inference efficiency in terms of energy cost and CO_2 emitted (40; 41). However, attribution systems vary widely in function and implementation.

In a survey of attribution methods, Hammoudeh et. al. (26) summarize inference time, space, and storage requirements for

influence analysis methods as a function of training dataset size, model parameter count, and training iteration count.

VI. Current Methods

A. Corroborative Attribution Methods

Prior work primarily focuses on identifying corroborative attributions with the textual entailment evaluator v_{TE}. Two common approaches to implementing v_{TE} are human reasoning (9; 20) and automated systems capable of natural language inference (NLI) (24; 25). Often, NLI systems are used in corroborative attribution systems to identify attributions, whereas human reasoning is used to evaluate attributions and also to generate training data for NLI systems. Both implementations exclude the usage of background information external to the source in judging the entailment relation (24). However, different sets of background knowledge may be leveraged by humans and NLI systems when interpreting the meaning of s and z (9); identifying discrepancies in NLI systems based on background knowledge and human judgment is important for addressing patterns of bias in evaluator performance.

Outside of implementing the evaluator, there are many different design choices to be made when building corroborative attribution systems and it is often unclear which method is the best. This is exacerbated by a lack of standardization in the evaluation metrics and datasets. To demonstrate this, we provide an overview of these implementations in Table II and how they align with the interaction model defined in Section IV.

In Table III, we outline the evaluation metrics used in prior work. Most proposed implementations evaluate attribution outputs with a metric that evaluates the quality of the LLM output, independent of the accompanying attribution, in addition to attribution correctness (Table III). To measure the quality of the LLM output, methods often measure the fluency or plausibility of the output to the user. Generally, this involves asking a user if the output is interpretable or helpful, or measuring performance on a QA or classification task (e.g., Exact Match for QA). Metrics for measuring correctness of an attribution set assess if the attributed output is fully supported by its corresponding corroborative documents (e.g., attribution precision and coverage). Li et. al. (52) provide a survey of techniques in this area; including metrics such as verifiability and factuality.

B. Contributive Attribution Methods For Language Models

Given a model, input, and output, contributive attributions provide a score for each source in the attribution domain that represents the relative amount that the source contributed to the output. The area of TDA for language tasks has been highlighted by Madsen et. al. (27) as a specific interpretability technique. Hammoudeh et. al. (26) give a broader view of different techniques for TDA that are theoretically applicable to language models. However, relatively few works thus far have specifically studied TDA in language models. We broadly categorize the many methods proposed for TDA into two families: data-centric and model-centric techniques. At a high-level, data-centric techniques average the effects of data changes across different models while model-centric techniques interrogate a single model. Since we are concerned with providing attributions for a specific model, we focus on describing verifiers for model-centric techniques.

*a) **Data-Centric TDA**:* To understand the impact of data points used to train models, one view is to take averages across different models that are trained without that data point. The common goal of retraining a model with the data point left out (i.e., leave-one-out (LOO) retraining) has been implemented differently by various techniques.

Let $f \in \mathcal{F}$ where $\mathcal{F}$ is a family of functions parameterized by θ trained on dataset D. Data-centric approaches characterize the influence (e.g., $\mathcal{I}(z_i, z_{te}, D)$) of a data point $z_i = (x_i, y_i)$ on a test point $z_{te} = (x_{te}, y_{te})$ over dataset D as an average effect over many possible models. For instance, LOO influence is the following:

$$\mathcal{I}_{LOO}(z_i, z_{te}, D) = \mathbb{E}_{f\in\mathcal{F}}\left[L(f(x_{te}, \theta_{D\setminus z_i}), y_{te}) - L(f(x_{te}, \theta_D), y_{te})\right]. \quad (1)$$

For LOO retraining, the effect of leaving one example out is averaged over different training runs removing the effect of the randomness of training. Approximations to LOO such as Datamodels (12) compute an average across leaving different subsets of points out and use the difference between logits as the functions L. Data Shapley Values (68) approximates this expectation using different possible subsets of the entire dataset. For Data Shapley, we can think of $\mathcal{F}$ as the family of functions induced by different subsets $D' \in D \setminus z_i$:

$$\mathcal{I}_{DS}(z_i, z_{te}, D) = \frac{1}{n}\sum_{D'\in D\setminus z_i} \frac{1}{\binom{n-1}{|D'|}} L(f(x_{te}, \theta_{D'}), y_{te}) - L(f(x_{te}, \theta_{D'\cup z_i}), y_{te}).$$

s These methods explicitly compute, approximate, or learn to predict counterfactual changes to the loss with one example removed.

*b) **Model-Centric TDA**:* For methods that aim at understanding and attributing a specific model, only parameters for a single model or a single training trajectory are considered. The *counterfactual contribution to loss* evaluator (v^M_{CCL}) is an abstraction of the notion of attribution in this section. Methods in this area take the following general form:

$$\mathcal{I}_{MC}(z_i, z_{te}, D) = \mathbb{E}_{f\in\mathcal{F}}[L(f(x_{te}, \theta_{D\setminus z_i}), y_{te})] - L(f(x_{te}, \theta_D), y_{te}). \quad (2)$$

While Equation (1) takes an expectation of both terms over $\mathcal{F}$ parameterized by θ trained on dataset D, Equation (2) only takes this expectation over the counterfactual term that excludes z_i from training. Therefore, $\mathcal{I}_{MC}(z_i, z_{te}, D)$ is relative to a specific model's loss, rather than to an expected model's loss.

Influence functions (10; 69) fall within this category because they approximate the expectation in the first term of $\mathcal{I}_{MC}(z_i, z_{te}, D)$ by modeling the response induced by upweighting z_i on model θ_D. Influence function methods estimate the counterfactual effect of individual training examples on model predictions for an individual model (10; 16; 18). Further

Method	Attributable Unit	Attribution Domain	Model	Evaluator
ALCE (6)	Output y parsed into sentences $\{z_1..z_n\}$	Wikipedia (2018-12-20), Sphere (42) (Filtered Common Crawl)	1. **Retrieval**: Retrieve top 100 passages (using GTR (43) and DPR (44) for Wikipedia and BM25 (45) for Sphere). 2. **Synthesis**: Synthesize retrieved passages to identify the k most relevant. 3. **Generation**: Include these k passages in-context alongside the input and additional prompting that instructs the model to cite the passages used.	**Textual Entailment**: NLI model that outputs 1 if the source entails the outputs.
GopherCITE (31)	Output y	Internet (queried by Google Search)	Collect human preferences of evidence paragraphs that support provided answers. Perform both supervised learning on highly rated samples and reinforcement learning from human preferences on Gopher (46), to learn a model that finds relevant web pages on the internet and quotes relevant passages to support its response.	**Textual Entailment**: LLM is fine-tuned to perform NLI.
LaMDA (47)	Output y	Internet (queried by information retrieval system that returns brief text snippets)	Model is fine-tuned to learn to call an external information retrieval system and use the results in-context to generate an attributed output.	**Textual Entailment**: LLM bases its output off of retrieved sources.
WebGPT (48)	Output y parsed into sentences $\{z_1..z_n\}$	Internet (queried by Microsoft Bing Web Search API)	Given a text-based web-browsing environment, GPT-3 is fine-tuned with RLHF to use the browser to identify sources it then uses in-context to answer the query.	**Textual Entailment**: LLM bases its output off of retrieved sources.
Lazaridou et al. (2022) (49)	Output y	Internet (queried by Google Search)	1. **Retrieval**: Extract text from top 20 URLs returned by Google to. 2. **Generation**: Use few-shot prompting to steer model to provide an answer conditioning on evidence. 3. **Attribution**: Rank all the paragraphs from top 20 URLs by cosine similarity between the paragraph and query.	**Cosine similarity** between question and evidence paragraphs.
RARR (25)	Output y	Internet (queried by Google Search)	1. **Generation**: For an input, which takes the form of a question, use PaLM (50) to generate the output. 2. **Retrieval**: Use Google Search to retrieve five web pages and then identify four-sentence evidence snippets from these pages that are relevant to the input, according to GTR (43). 3. **Attribution**: Use chain-of-thought few-shot prompting (51) on PaLM (50) to identify cases where the evidence snippet and the model output provide the same answer to the input.	**Valid Paraphrase:** LLM identifies when the source and model output provide the same answer to the input.

TABLE II
OVERVIEW OF EXISTING CORROBORATIVE ATTRIBUTION SYSTEMS FOR LANGUAGE MODELS

work acknowledges that when applied to nonconvex learning objectives, influence functions more closely estimate the Proximal Bregman Response Function, rather than the counterfactual influence (17; 69). All of these methods are implementations, even if computationally impractical for today's LLMs, of the counterfactual contribution to loss v^M_{CCL} evaluator.

For Gradient Tracing methods, such as TracIn (14), the quantity measured is different from all the definitions above and we believe it lacks the explicit counterfactual motivation needed for contributive attributions. Specifically, the ideal objective function of TracIn seeks to measure the contribution of an example to the loss over the training process by summing the change in loss across training time steps that include z_i in the batch:

$$\mathcal{I}_{TI}(z_i, z_{te}, D) = \sum_{t:z_i \in B_t} L(f(x_{te}, \theta_{t-1}), y_{te}) - L(f(x_{te}, \theta_t), y_{te}).$$

TracIn does not explicitly define a relationship between its notion of influence of a training point z_i and the final model's behavior on the test point z_{te}. Therefore, this method does not fall within our framework of counterfactual evaluators.

VII. USE CASES REQUIRING ATTRIBUTIONS

While perhaps the most obvious use case of attributions is to provide citations for a model's answer to a question, the interaction model we have presented obviates a number of use cases, each with its own list of desirable properties. Across the board, the properties of correctness and high efficiency are important. Depending on the use case, either contributive attributions, corroborative attributions, or a composition of the

Method	Datasets	Non-attribution Evaluation	Attribution Evaluation: Correctness
Attributable to Identified Sources (AIS) (9)	QReCC (53) and WoW (QA) (54), CNN/DM (summarization) (55), ToTTo (table-to-text task) (56)	**Human Reasoning**: Is all of the information relayed by the system response **interpretable** to you?	**Human Reasoning**: Is all of the information provided by the system response (a) **fully supported** by the source document?
Evaluating Verifiability in Generative Search Engines (20)	AllSouls, davinci-debate, ELI5 (57), WikiHowKeywords, NaturalQuestions (58) (all filtered)	**Human Reasoning**: **Fluency, perceived utility** (whether the response is a helpful and informative answer to the query)	**Human Reasoning**: Coverage, citation precision
Automatic Evaluation of Attribution by Large Language Models (4)	HotpotQA (59), EntityQuestions (60), PopQA (61), TREC (62), TriviaQA (63), WebQuestions (64)	**None**	**Automatic Evaluation**: Fine-grained citation precision: Is the attribution attributable, extrapolatory, or contradictory?
ALCE (6)	ASQA (65), QAMPARI (66), ELI5 (57)	**Automatic Evaluation**: Fluency (MAUVE), Correctness (compared to a ground truth answer) measured with exact match and entailment (NLI)	**Automatic Evaluation**: Coverage, citation precision
GopherCITE (31)	NaturalQuestionsFiltered, ELI5Filtered	**Human Reasoning**: Is the answer a **plausible reply** to the question?	**Human Reasoning**: Coverage
WebGPT (48)	ELI5 (57), TruthfulQA (67)	**Human Reasoning**: Overall usefulness, coherence	**Human Reasoning**: Factual correctness

TABLE III
OVERVIEW OF THE EVALUATION OF CORROBORATIVE ATTRIBUTIONS

Method Type	Oracle Evaluator	Implemented Evaluator	LM Implementations
Data-Centric Methods			
Leave-one-out	Change in the expected counterfactual output	Expected counterfactual contribution to the loss	DataModels (12)
Shapley Values	Change in the expected counterfactual output	Expected counterfactual contribution to the loss	Data Shapley (70)
Model-Centric Methods			
Influence Functions	Change in the counterfactual output (v^M_{CCO})	Counterfactual contribution to the loss (v^M_{CCL})	* TRAK (18) * EK-FAC (17)
Gradient Tracing	Change in training trajectory	Contribution to the loss	TracIn (14) * Simfluence (11) * TracIN-WE (37)

TABLE IV
OVERVIEW OF CONTRIBUTIVE ATTRIBUTION METHODS (METHODS WITH LLM IMPLEMENTATIONS ARE DENOTED WITH *)

Task	Properties				
	Correct.	Recall	Effici.	Consist.	Relev.
Corroborative Attribution					
Question Answering	✓		✓		✓
Fact Checking	✓		✓		
Contributive Attribution					
Author Compensation	✓	✓	✓	✓	
GDPR Compliance	✓	✓	✓	✓	
Model Bias Detection	✓	✓	✓		
Contributive+Corroborative Attribution					
Model Debugging	✓	✓	✓		
Auditing Model Memorization	✓	✓	✓	✓	
Human AI Collaboration	✓	✓	✓	✓	✓

TABLE V
OVERVIEW OF ATTRIBUTION USE CASES AND THEIR DESIRED PROPERTIES.

two are required. In this section, we enumerate use cases and our recommendation on how to apply attributions.

A. Use Cases of Corroborative Attributions

While there are a variety of use cases where corroborative attributions are important, we highlight several tasks that showcase how different attribution properties and metrics are meaningful.

a) ***Question Answering****:* QA is a common task for LLMs. Unfortunately, LLM answers are not always trustworthy, especially in critical domains such as law and healthcare (71). (8) and QA engines such as Bing Chat and Perplexity AI have explored using corroborative attributions to provide citations for answers (72). In this use case, humans can verify the output by examining the sources that are provided as attributions. This step of output verification by the human user is critical because the attribution domain may not be fully composed of trusted sources (e.g., QA engines retrieve from the internet).

High attribution recall is not a strict requirement for QA since only a few corroborating sources may be sufficient to support an attributable unit. Implementations of attribution for QA may customize source relevance to prioritize primary sources, rather

than secondary sources, or more reputable sources, rather than those from authors of dubious credentials.

*b) **Fact Checking**:* Fact checking has emerged as a promising tool in the fight against misinformation (73). Despite its importance, fact checking has long been an entirely manual process (74). Many researchers have attempted to automate fact checking (33). We posit that our attributions framework can help create and evaluate methods for fact checking.

Given an attribution domain of sources that are up-to-date, trustworthy, and non-contradictory, it follows that an attributable unit can be taken as true if it has at least one corroborative attribution. Therefore, high attribution recall is not an important property for this use case. As in the QA use case, customized source relevance can be useful for prioritizing primary sources. However, because the attribution domain is assumed to contain only trustworthy sources, customized source relevance is redundant to the end of selecting trusted sources.

Interestingly, perfect coverage is *not* necessarily desired in this use case; low coverage indicates that either the output is nonfactual or that the attribution domain does not include sufficient sources to corroborate the statement. If the model output is factual, however, the coverage should be perfect. Coverage is perhaps a numerical counterpart to non-binary labels for factuality, such as "mostly true" or "half true", from previous work (33).

This setting of fact-checking motivates another class of corroborative evaluators that indicates a lack of logical entailment. For example, an evaluator that indicates when a source contradicts an attribution unit would make it possible to flag a model output for containing misinformation. Prior work has implemented such evaluators before; RARR (25) first identifies sources that are relevant to an LLM output and then post-edits unattributed parts of the output by using an agreement model that indicates when part of the output disagrees with a source.

B. Use Cases of Contributive Attributions

Prior work has explored using contributive attributions to understand the training data of models. We discuss some of these tasks and their desired properties here.

*a) **Author Compensation**:* With LLMs being trained on large datasets that include sources under various licenses, people have begun to observe language models returning output that heavily resembles licensed works owned by specific authors. As a result, thousands of authors have demanded compensation for their work being used to train language models (75). This demand necessitates the ability to attribute language model output to specific author sources and to quantify the degree to which the author's work contributed to the output.

In this use case, authors could be compensated based on their work appearing in the contributive attributions of an LLM output. High attribution recall and consistency are critical since leaving out a major contributor could have legal consequences.

*b) **GDPR Compliance**:* GDPR compliance requires language model maintainers to update their models by removing the influence of training data upon request. Prior work has explored efficient data deletion for ML models (76) to avoid training from scratch with a few data points removed. In such a scenario, it is critical to ensure that the original data points are no longer contributing to the model output.

An empty contributive attribution set for a set of language model outputs can imply the deleted data is no longer influential. The attribution set must have high attribution recall or else an empty set may be a false positive for compliance. For the same reason, stability is also critical.

C. Use Cases of Corroborative and Contributive Attributions

We describe several use cases that require both corroborative and contributive attributions for LLM predictions.

*a) **Model Debugging**:* Identifying the training data points that contribute to a test case that is incorrect, or otherwise undesirable (e.g., toxic), is helpful for cleaning the training data and remedying the failure case in the model development cycle.[7] While this has been a longstanding motivation of TDA papers (10; 13; 14; 15), we argue that when working with language models, not only do we need contributive attributions, but we also need corroborative notions of attribution. This is because TDA methods are not guaranteed to flag training sources that are semantically relevant to the input and output (17); removing semantically *unrelated* contributive sources is not guaranteed to change the semantic meaning of the model output. Therefore, the semantic relation between contributive sources and the input and output is important for model debugging. Corroborative attributions are integral in identifying such semantic relation. *Data poisoning detection* (10) is adjacent to model debugging and thus requires the same types of attribution.

*b) **Document Generation**:* The task of document generation, or drafting, refers to the language model writing a passage of text given a prompt. A growing number of ventures are now proposing using LLMs for writing documents such as legal briefs and contracts (Section VIII-A). In this task, both types of attributions are helpful for the generated output y. Contributive attributions would provide context for what sources the generated documents are similar to and corroborative attributions would provide validation for the claims made in the generated document.

*c) **Auditing Model Memorization**:* To determine that an output is a case of model memorization of a training point, the output must exactly match a training point that was also highly influential in its generation. Therefore, this use case requires exact match corroborative attributions, as well as contributive attributions. Prior work has measured the extent to which models have memorized their training sources via self-influence, defined as the influence of a training point on its own loss (14; 77). However, this approach does not extend to the evaluation of inputs from outside the training set. Furthermore, we believe that heuristic approaches that solely use corroborative exact match to diagnose cases of

[7]Retraining an LLM from scratch is too resource intensive to be practical. However, the fine-tuning process is less resource intensive and more reasonable to repeat; attributions for fine-tuned model outputs to fine-tuning data may be an actionable setting for debugging models with attributions.

model memorization exclude contributive attributions due to the inefficiency of current TDA methods.

*d) **Human-AI Collaboration**:* Another rapidly emerging use case is using LLMs for human-AI collaboration. For example, Sun et. al. (78) study AI-supported software engineering through several language model collaborative tasks. In their study, participants wanted to know how the code was generated (i.e., contributive attribution) as well as code correctness (e.g., corroborative attributions). Liao et. al. (79) summarize a broader family of AI-assisted tasks such as including decision support and communication support; study participants wanted to know what training data produced the model suggestion as well as the correctness of the suggestion. Furthermore, in application domains such as assistive call center tools or travel itinerary tools, companies are using LLMs for various collaborative planning and decision tasks.[8] In Human-AI collaboration tasks, all of the properties we describe may be important. Particularly, when a task process is documented, consistency in the attribution provided for making such a decision is important. In this example, both types of attributions are desired for the same output y of a language model.

VIII. Case Studies: A Closer Look at Two Application Domains

A. Case Study 1: LLMs for Legal Drafting

AI and LLMs in particular have been increasingly applied to the legal domain as training data for different legal tasks are becoming more readily available (80; 81; 82; 83; 84; 85). While LLMs show promising results for legal document analysis, contract review, and legal research, they also raise concerns regarding privacy, bias, and explainability (86; 87). To address such shortcomings, the development of attribution methods to promote transparency and interpretability are needed (86). Moreover, Bommasani et. al. (2) discuss the opportunities and risks of using foundation models for applications rooted in US law in particular. They review different fields of law and specifically contemplate the ability of foundation models to write legal briefs. While tools for writing legal briefs using language models are still under development, different products based on LLMs such as legal question answering, immigration case drafting, and document summarization have started to appear in various startups.[9] In this case study, we describe the document generation setting when an LLM is used by a lawyer or firm to draft a legal document. The input would be a prompt asking for a specific type of legal document (e.g., a contract or brief) for a specific purpose and the output would be the resulting document.

In this setting, a lawyer may want **contributive** attributions to understand which training documents the generated document is borrowing words or concepts from. For example, if the document requested is a bespoke rental contract, users may want to ensure that the generated contract is not borrowing from rental contracts from other states or countries. Continuing with the rental contract example, **corroborative** attributions are also important to ensure the contract adheres to local laws. The sources for such corroborative attributions need not be in the training data and may come from a repository of documents that are more frequently updated than the language model itself. In this setting, the LLM is assistive to lawyers handling the case. Correct attributions that provide the right sources to corroborate the drafted document are important. High-precision attributions in particular would improve the efficiency of lawyers using these tools.

B. Case Study 2: LLMs for Healthcare

The application of language models to the field of medicine has been heavily studied (88; 89; 90; 91; 92; 93). Recently, LLMs have been increasingly adopted for real-world clinical tasks that largely fall into the two categories of summarization of clinical notes (94; 95; 96) and medical QA (93; 97; 98).

The task of summarizing clinical notes has received attention in both academia and industry.[10] These summaries have been evaluated for consistency with the underlying clinical notes using automated metrics, such as ROUGE and BERTScore (99), and human evaluators (94; 95; 96). These examples highlight how **corroborative** attributions can validate generated summaries using the underlying clinical notes. Furthermore, interviews with medical practitioners reveal the importance of **contributive** attributions for trusting model outputs (100). Together, corroborative and contributive attributions can provide insights into the validity of a summary of clinical notes.

For medical QA systems[11], it is important for clinicians to have citations of evidence to support model answers (101). **Corroborative** attributions can be used to provide these citations, as is done by MediSearch and OpenEvidence. While these two companies broadly restrict their attribution domains to research publications from reputable venues, MedAlign (102) highlights the option of using a corpus of EHRs. The implementation of corroborative attributions with trusted attribution domains is adjacent to the use case of fact checking, the stakes of which are particularly high in the clinical setting due to the potential consequences on human health.

Attributions are also valuable in debugging medical QA LLMs, such as MedPaLM 2 (98), by flagging training sources that are relevant to incorrect outputs. As discussed previously in VII-C, this can be accomplished with a composition of **contributive** and **corroborative** attributions. Model developers and medical experts should leverage domain knowledge when manually inspecting training sources flagged for debugging.

IX. Future Work

We highlight several promising directions for future work.

[8] Start-ups in this area at the time of this publication include Observe AI, GenixGPT

[9] Y-Combinator companies in this area at the time of this publication include Casehopper, Lexiter.ai, DocSum.ai, and Atla AI

[10] Start-ups in summarization at the time of this publication include Notable Health and Abridge AI.

[11] Companies in this area at the time of this publication include MediSearch, Open Evidence, Hippocratic AI, and Glass Health.

A. Counterfactual contribution to output evaluators

In Definition 5, we outline the possibility of contributive evaluators that are sensitive to semantic changes in the counterfactual output, rather than to changes in the counterfactual loss. The notion of citation to parametric content discussed by Huang et al. (19) also addresses this potential connection between contributive attribution and the semantic content of the output. To the best of our knowledge, such output-based contributive attributions for LLMs have not yet been explored. A critical challenge of implementing CCO evaluators lies in detangling the stochasticity of the decoding strategy from the contribution of sources. Future work in addressing this limitation would allow for semantically meaningful contributive attributions.

B. Contributive attributions with large-scale training data

The large scale of data used to train LLMs raises concerns not only about the high resource burdens of TDA methods, but also whether the influence of a single training source is meaningfully noticeable on the loss, not to mention the output. Past work has quantitatively observed that training sources with high influences are more rare than not, but they do exist and in fact largely make up the total influence on an LLM output (17). Nonetheless, future work may consider extending contributive attributions for language models to notions of influence on a group of training sources, rather than individual training sources (103). Also, the ubiquity of finetuning encourages further work on TDA methods suited for finetuned models (16). In this case, the attribution domain could be restricted to the finetuning dataset, which is orders of magnitude smaller than the pre-training dataset. This direction is an interesting pursuit in and of itself, especially for model developers interested in debugging fine-tuned models.

C. Hybrid attribution systems

While we present a framework that unifies existing work in both corroborative and contributive attribution literature, developing techniques capable of both types of attributions is left to future work. The area of *fact-tracing* makes a step in this direction by providing contributive attributions in a setting where corroboration matters (22). However, the identification and corroboration of facts within the language model output requires further work. Hybrid attribution systems would improve the customizability of attributions, potentially making them useful across a broader range of applications.

D. Standardized Evaluation

From our survey of attribution methods, particularly for corroborative attribution, we observe that evaluation is not standardized between methods. Each attribution method is evaluated on different datasets and often with different metrics. For example, GopherCITE's (31) outputs are evaluated on a subset of NaturalQuestions and ELI5 with binary metrics if the answer is plausible and supported by the attribution. On the other hand, WebGPT's (48) outputs are evaluated on a different subset of ELI5 and open-ended dialogue interactions by comparisons to human-generated attributions. More broadly, the utility of an attribution can be expanded beyond correctness to the other properties we introduce.

E. Use-Case-Driven Development and Evaluation

In our work, we explore tasks and case studies where attributions are important for industry applications of LLMs. We recommend that attribution system developers choose an application and then identify the relevant properties for evaluation. This approach of use-case-driven development is preferable to strong-arming a developed method to serve a use case. Furthermore, use-case-driven development may surface additional settings where corroborative and contributive attributions are needed simultaneously.

X. Conclusion

This paper presents a unifying framework for corroborative and contributive attributions in LLMs. We formulate an interaction model to define the core components of attributions and to define their properties. This framework serves as a lens for analyzing existing attribution methods and use cases for attributions. Our analysis elucidates prescriptive suggestions for future research, namely CCO evaluators, the challenges of contributive methods at the scale of LLMs, the value of hybrid attribution systems, the need for standardized evaluation of attribution systems, and use-case-driven development. We hope our unifying perspective on the field of attributions leads to improved solutions for misinformation, accountability, and transparency in real-world applications of language models.

XI. Acknowledgements

This paper was developed in a fairness, accountability, transparency, and explainability working group run by Carlos Guestrin. We would like to thank Anka Reul, Ava Jeffs, Krista Opsahl-Ong, Myra Cheng, Sierra Wang, and Xuechen Li as well as all members of the working group for their thoughts and feedback in early discussions. We also thank Tatsunori Hashimoto and John Hewitt for their feedback on our manuscript. TW and NM are supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-2146755. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors(s) and do not necessarily reflect the views of the National Science Foundation. NM was also supported by the Stanford Electrical Engineering Department Fellowship. JHS is supported by the Simons collaboration on the theory of algorithmic fairness and the Simons Foundation Investigators award 689988. CG is a Chan Zuckerberg Biohub – San Francisco Investigator. The figures in this work have been designed using images from Flaticon.

References

[1] R. Azamfirei, S. R. Kudchadkar, and J. Fackler, "Large language models and the perils of their hallucinations," *Critical Care*, vol. 27, no. 1, pp. 1–2, 2023.

[2] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill *et al.*, "On the opportunities and risks

of foundation models," *arXiv preprint arXiv:2108.07258*, 2021.

[3] B. Fetahu, K. Markert, W. Nejdl, and A. Anand, "Finding news citations for wikipedia," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, ser. CIKM '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 337–346. [Online]. Available: https://doi.org/10.1145/2983323.2983808

[4] X. Yue, B. Wang, K. Zhang, Z. Chen, Y. Su, and H. Sun, "Automatic evaluation of attribution by large language models," *arXiv preprint arXiv:2305.06311*, 2023.

[5] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. Chang, "Retrieval augmented language model pre-training," in *International conference on machine learning*. PMLR, 2020, pp. 3929–3938.

[6] T. Gao, H. Yen, J. Yu, and D. Chen, "Enabling large language models to generate text with citations," 2023.

[7] D. Metzler, Y. Tay, D. Bahri, and M. Najork, "Rethinking search: Making domain experts out of dilettantes," *SIGIR Forum*, vol. 55, no. 1, jul 2021. [Online]. Available: https://doi.org/10.1145/3476415.3476428

[8] B. Bohnet, V. Q. Tran, P. Verga, R. Aharoni, D. Andor, L. B. Soares, J. Eisenstein, K. Ganchev, J. Herzig, K. Hui *et al.*, "Attributed question answering: Evaluation and modeling for attributed large language models," *arXiv preprint arXiv:2212.08037*, 2022.

[9] H. Rashkin, V. Nikolaev, M. Lamm, L. Aroyo, M. Collins, D. Das, S. Petrov, G. S. Tomar, I. Turc, and D. Reitter, "Measuring attribution in natural language generation models," *arXiv preprint arXiv:2112.12870*, 2021.

[10] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *International conference on machine learning*. PMLR, 2017, pp. 1885–1894.

[11] K. Guu, A. Webson, E. Pavlick, L. Dixon, I. Tenney, and T. Bolukbasi, "Simfluence: Modeling the influence of individual training examples by simulating training runs," 2023.

[12] A. Ilyas, S. M. Park, L. Engstrom, G. Leclerc, and A. Madry, "Datamodels: Predicting predictions from training data," *arXiv preprint arXiv:2202.00622*, 2022.

[13] C.-K. Yeh, J. S. Kim, I. E. H. Yen, and P. Ravikumar, "Representer point selection for explaining deep neural networks," 2018.

[14] G. Pruthi, F. Liu, S. Kale, and M. Sundararajan, "Estimating training data influence by tracing gradient descent," *Advances in Neural Information Processing Systems*, vol. 33, pp. 19 920–19 930, 2020.

[15] A. Schioppa, P. Zablotskaia, D. Vilar, and A. Sokolov, "Scaling up influence functions," 2021.

[16] Y. Kwon, E. Wu, K. Wu, and J. Zou, "Datainf: Efficiently estimating data influence in lora-tuned llms and diffusion models," 2023.

[17] R. Grosse, J. Bae, C. Anil, N. Elhage, A. Tamkin, A. Tajdini, B. Steiner, D. Li, E. Durmus, E. Perez *et al.*, "Studying large language model generalization with influence functions," *arXiv preprint arXiv:2308.03296*, 2023.

[18] S. M. Park, K. Georgiev, A. Ilyas, G. Leclerc, and A. Madry, "Trak: Attributing model behavior at scale," 2023.

[19] J. Huang and K. C.-C. Chang, "Citation: A key to building responsible and accountable large language models," 2023.

[20] N. F. Liu, T. Zhang, and P. Liang, "Evaluating verifiability in generative search engines," 2023.

[21] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," 2017.

[22] E. Akyürek, T. Bolukbasi, F. Liu, B. Xiong, I. Tenney, J. Andreas, and K. Guu, "Towards tracing knowledge in language models back to the training data," in *Findings of the Association for Computational Linguistics: EMNLP 2022*, 2022, pp. 2429–2446.

[23] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.

[24] O. Honovich, R. Aharoni, J. Herzig, H. Taitelbaum, D. Kukliansy, V. Cohen, T. Scialom, I. Szpektor, A. Hassidim, and Y. Matias, "True: Re-evaluating factual consistency evaluation," *arXiv preprint arXiv:2204.04991*, 2022.

[25] L. Gao, Z. Dai, P. Pasupat, A. Chen, A. T. Chaganty, Y. Fan, V. Y. Zhao, N. Lao, H. Lee, D.-C. Juan, and K. Guu, "Rarr: Researching and revising what language models say, using language models," 2023.

[26] Z. Hammoudeh and D. Lowd, "Training data influence analysis and estimation: A survey," *arXiv preprint arXiv:2212.04612*, 2022.

[27] A. Madsen, S. Reddy, and S. Chandar, "Post-hoc interpretability for neural nlp: A survey," *ACM Computing Surveys*, vol. 55, no. 8, pp. 1–42, 2022.

[28] J. Ramos *et al.*, "Using tf-idf to determine word relevance in document queries," in *Proceedings of the first instructional conference on machine learning*, vol. 242, no. 1. Citeseer, 2003, pp. 29–48.

[29] A. Roberts, C. Raffel, and N. Shazeer, "How much knowledge can you pack into the parameters of a language model?" in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020.

[30] K. Nakamura, S. Levy, Y.-L. Tuan, W. Chen, and W. Y. Wang, "Hybridialogue: An information-seeking dialogue dataset grounded on tabular and textual data," 2022.

[31] J. Menick, M. Trebacz, V. Mikulik, J. Aslanides, F. Song, M. Chadwick, M. Glaese, S. Young, L. Campbell-Gillingham, G. Irving, and N. McAleese, "Teaching language models to support answers with verified quotes,"

2022.

[32] E. Cosijn and P. Ingwersen, "Dimensions of relevance," *Information Processing & Management*, vol. 36, no. 4, pp. 533–550, 2000.

[33] Z. Guo, M. Schlichtkrull, and A. Vlachos, "A survey on automated fact-checking," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 178–206, 2022.

[34] X. Yue, X. Pan, W. Yao, D. Yu, D. Yu, and J. Chen, "C-more: Pretraining to answer open-domain questions by consulting millions of references," *arXiv preprint arXiv:2203.08928*, 2022.

[35] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning—i," *Information sciences*, vol. 8, no. 3, pp. 199–249, 1975.

[36] A. Erasmus, T. D. Brunet, and E. Fisher, "What is interpretability?" *Philosophy & Technology*, vol. 34, no. 4, pp. 833–862, 2021.

[37] C.-K. Yeh, A. Taly, M. Sundararajan, F. Liu, and P. Ravikumar, "First is better than last for language data influence," *Advances in Neural Information Processing Systems*, vol. 35, pp. 32 285–32 298, 2022.

[38] S. Basu, P. Pope, and S. Feizi, "Influence functions in deep learning are fragile," *arXiv preprint arXiv:2006.14651*, 2020.

[39] A. Søgaard *et al.*, "Revisiting methods for finding influential examples," *arXiv preprint arXiv:2111.04683*, 2021.

[40] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, "On the dangers of stochastic parrots: Can language models be too big?" in *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, 2021, pp. 610–623.

[41] P. Liang, R. Bommasani, T. Lee, D. Tsipras, D. Soylu, M. Yasunaga, Y. Zhang, D. Narayanan, Y. Wu, A. Kumar *et al.*, "Holistic evaluation of language models," *arXiv preprint arXiv:2211.09110*, 2022.

[42] A. Piktus, F. Petroni, V. Karpukhin, D. Okhonko, S. Broscheit, G. Izacard, P. Lewis, B. Oğuz, E. Grave, W. tau Yih, and S. Riedel, "The web is your oyster - knowledge-intensive nlp against a very large web corpus," 2022.

[43] J. Ni, C. Qu, J. Lu, Z. Dai, G. Hernandez Abrego, J. Ma, V. Zhao, Y. Luan, K. Hall, M.-W. Chang, and Y. Yang, "Large dual encoders are generalizable retrievers," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022.

[44] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, "Dense passage retrieval for open-domain question answering," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020.

[45] S. Robertson, H. Zaragoza *et al.*, "The probabilistic relevance framework: Bm25 and beyond," *Foundations and Trends in Information Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.

[46] J. W. Rae, S. Borgeaud, T. Cai, K. Millican, J. Hoffmann, F. Song, J. Aslanides, S. Henderson, R. Ring, S. Young, E. Rutherford, T. Hennigan, J. Menick, A. Cassirer, R. Powell, G. van den Driessche, L. A. Hendricks, M. Rauh, P.-S. Huang, A. Glaese, J. Welbl, S. Dathathri, S. Huang, J. Uesato, J. Mellor, I. Higgins, A. Creswell, N. McAleese, A. Wu, E. Elsen, S. Jayakumar, E. Buchatskaya, D. Budden, E. Sutherland, K. Simonyan, M. Paganini, L. Sifre, L. Martens, X. L. Li, A. Kuncoro, A. Nematzadeh, E. Gribovskaya, D. Donato, A. Lazaridou, A. Mensch, J.-B. Lespiau, M. Tsimpoukelli, N. Grigorev, D. Fritz, T. Sottiaux, M. Pajarskas, T. Pohlen, Z. Gong, D. Toyama, C. de Masson d'Autume, Y. Li, T. Terzi, V. Mikulik, I. Babuschkin, A. Clark, D. de Las Casas, A. Guy, C. Jones, J. Bradbury, M. Johnson, B. Hechtman, L. Weidinger, I. Gabriel, W. Isaac, E. Lockhart, S. Osindero, L. Rimell, C. Dyer, O. Vinyals, K. Ayoub, J. Stanway, L. Bennett, D. Hassabis, K. Kavukcuoglu, and G. Irving, "Scaling language models: Methods, analysis & insights from training gopher," 2022.

[47] R. Thoppilan, D. D. Freitas, J. Hall, N. Shazeer, A. Kulshreshtha, H.-T. Cheng, A. Jin, T. Bos, L. Baker, Y. Du, Y. Li, H. Lee, H. S. Zheng, A. Ghafouri, M. Menegali, Y. Huang, M. Krikun, D. Lepikhin, J. Qin, D. Chen, Y. Xu, Z. Chen, A. Roberts, M. Bosma, V. Zhao, Y. Zhou, C.-C. Chang, I. Krivokon, W. Rusch, M. Pickett, P. Srinivasan, L. Man, K. Meier-Hellstern, M. R. Morris, T. Doshi, R. D. Santos, T. Duke, J. Soraker, B. Zevenbergen, V. Prabhakaran, M. Diaz, B. Hutchinson, K. Olson, A. Molina, E. Hoffman-John, J. Lee, L. Aroyo, R. Rajakumar, A. Butryna, M. Lamm, V. Kuzmina, J. Fenton, A. Cohen, R. Bernstein, R. Kurzweil, B. Aguera-Arcas, C. Cui, M. Croak, E. Chi, and Q. Le, "Lamda: Language models for dialog applications," 2022.

[48] R. Nakano, J. Hilton, S. Balaji, J. Wu, L. Ouyang, C. Kim, C. Hesse, S. Jain, V. Kosaraju, W. Saunders *et al.*, "Webgpt: Browser-assisted question-answering with human feedback," *arXiv preprint arXiv:2112.09332*, 2021.

[49] A. Lazaridou, E. Gribovskaya, W. Stokowiec, and N. Grigorev, "Internet-augmented language models through few-shot prompting for open-domain question answering," *arXiv preprint arXiv:2203.05115*, 2022.

[50] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann *et al.*, "Palm: Scaling language modeling with pathways," *arXiv preprint arXiv:2204.02311*, 2022.

[51] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," 2023.

[52] D. Li, Z. Sun, X. Hu, Z. Liu, Z. Chen, B. Hu, A. Wu, and M. Zhang, "A survey of large language models attribution," *arXiv preprint arXiv:2311.03731*, 2023.

[53] R. Anantha, S. Vakulenko, Z. Tu, S. Longpre, S. Pulman, and S. Chappidi, "Open-domain question answering goes conversational via question rewriting," 2021.

[54] E. Dinan, S. Roller, K. Shuster, A. Fan, M. Auli, and J. Weston, "Wizard of wikipedia: Knowledge-powered conversational agents," 2019.

[55] R. Nallapati, B. Zhou, C. N. dos santos, C. Gulcehre, and B. Xiang, "Abstractive text summarization using sequence-to-sequence rnns and beyond," 2016.

[56] A. Parikh, X. Wang, S. Gehrmann, M. Faruqui, B. Dhingra, D. Yang, and D. Das, "ToTTo: A controlled table-to-text generation dataset," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds. Online: Association for Computational Linguistics, Nov. 2020, pp. 1173–1186. [Online]. Available: https://aclanthology.org/2020.emnlp-edefYou're in trouble here. Try typing <return> to proceed.˙If that doesn't work, type X <return> to quit.$main$.89

[57] A. Fan, Y. Jernite, E. Perez, D. Grangier, J. Weston, and M. Auli, "ELI5: Long form question answering," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, A. Korhonen, D. Traum, and L. Màrquez, Eds. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 3558–3567. [Online]. Available: https://aclanthology.org/P19-edefYou're in trouble here. Try typing <return> to proceed.˙If that doesn't work, type X <return> to quit.1346

[58] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, K. Toutanova, L. Jones, M. Kelcey, M.-W. Chang, A. M. Dai, J. Uszkoreit, Q. Le, and S. Petrov, "Natural questions: A benchmark for question answering research," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 452–466, 2019. [Online]. Available: https://aclanthology.org/Q19-edefYou're in trouble here. Try typing <return> to proceed.˙If that doesn't work, type X <return> to quit.1026

[59] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, and C. D. Manning, "HotpotQA: A dataset for diverse, explainable multi-hop question answering," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, Eds. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 2369–2380. [Online]. Available: https://aclanthology.org/D18-edefYou're in trouble here. Try typing <return> to proceed.˙If that doesn't work, type X <return> to quit.1259

[60] C. Sciavolino, Z. Zhong, J. Lee, and D. Chen, "Simple entity-centric questions challenge dense retrievers," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, Eds. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 6138–6148. [Online]. Available: https://aclanthology.org/2021.emnlp-edefYou're in trouble here. Try typing <return> to proceed.˙If that doesn't work, type X <return> to quit.$main$.496

[61] A. Mallen, A. Asai, V. Zhong, R. Das, D. Khashabi, and H. Hajishirzi, "When not to trust language models: Investigating effectiveness of parametric and non-parametric memories," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 9802–9822. [Online]. Available: https://aclanthology.org/2023.acl-edefYou're in trouble here. Try typing <return> to proceed.˙If that doesn't work, type X <return> to quit.$long$.546

[62] P. Baudis and J. Sedivý, "Modeling of the question answering task in the yodaqa system," in *Conference and Labs of the Evaluation Forum*, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:17723267

[63] M. Joshi, E. Choi, D. Weld, and L. Zettlemoyer, "TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, R. Barzilay and M.-Y. Kan, Eds. Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 1601–1611. [Online]. Available: https://aclanthology.org/P17-edefYou're in trouble here. Try typing <return> to proceed.˙If that doesn't work, type X <return> to quit.1147

[64] J. Berant, A. Chou, R. Frostig, and P. Liang, "Semantic parsing on Freebase from question-answer pairs," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, D. Yarowsky, T. Baldwin, A. Korhonen, K. Livescu, and S. Bethard, Eds. Seattle, Washington, USA: Association for Computational Linguistics, Oct. 2013, pp. 1533–1544. [Online]. Available: https://aclanthology.org/D13-edefYou're in trouble here. Try typing <return> to proceed.˙If that doesn't work, type X <return> to quit.1160

[65] I. Stelmakh, Y. Luan, B. Dhingra, and M.-W. Chang, "ASQA: Factoid questions meet long-form answers," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 8273–8288. [Online]. Available: https://aclanthology.org/2022.emnlp-edefYou're in trouble here. Try typing <return>

to proceed. If that doesn't work, type X <return> to quit.$main.566$

[66] S. J. Amouyal, T. Wolfson, O. Rubin, O. Yoran, J. Herzig, and J. Berant, "Qampari: An open-domain question answering benchmark for questions with many answers from multiple paragraphs," 2023.

[67] S. Lin, J. Hilton, and O. Evans, "Truthfulqa: Measuring how models mimic human falsehoods," 2022.

[68] A. Ghorbani and J. Zou, "Data shapley: Equitable valuation of data for machine learning," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2242–2251.

[69] J. Bae, N. Ng, A. Lo, M. Ghassemi, and R. B. Grosse, "If influence functions are the answer, then what is the question?" *Advances in Neural Information Processing Systems*, vol. 35, pp. 17 953–17 967, 2022.

[70] A. Ghorbani and J. Zou, "Data shapley: Equitable valuation of data for machine learning," 2019.

[71] A. Choudhury and H. Shamszare, "Investigating the impact of user trust on the adoption and use of chatgpt: Survey analysis," *Journal of Medical Internet Research*, vol. 25, p. e47184, 2023.

[72] U. A. Khan, "The unstoppable march of artificial intelligence: The dawn of large language models," *eSignals PRO*, 2023.

[73] N. M. Krause, I. Freiling, B. Beets, and D. Brossard, "Fact-checking as risk communication: the multi-layered risk of misinformation in times of covid-19," *Journal of Risk Research*, vol. 23, no. 7-8, pp. 1052–1059, 2020.

[74] M. A. Amazeen, "Revisiting the epistemology of fact-checking," *Critical Review*, vol. 27, no. 1, pp. 1–22, 2015.

[75] P. Samuelson, "Generative ai meets copyright," *Science*, vol. 381, no. 6654, pp. 158–161, 2023.

[76] J. Brophy, "Exit through the training data: A look into instance-attribution explanations and efficient data deletion in machine learning," *Technical report*, 2020.

[77] V. Feldman and C. Zhang, "What neural networks memorize and why: Discovering the long tail via influence estimation," 2020.

[78] J. Sun, Q. V. Liao, M. Muller, M. Agarwal, S. Houde, K. Talamadupula, and J. D. Weisz, "Investigating explainability of generative ai for code through scenario-based design," in *27th International Conference on Intelligent User Interfaces*, 2022, pp. 212–228.

[79] Q. V. Liao, D. Gruen, and S. Miller, "Questioning the ai: informing design practices for explainable ai user experiences," in *Proceedings of the 2020 CHI conference on human factors in computing systems*, 2020, pp. 1–15.

[80] P. Henderson, M. S. Krass, L. Zheng, N. Guha, C. D. Manning, D. Jurafsky, and D. E. Ho, "Pile of law: Learning responsible data filtering from the law and a 256gb open-source legal dataset," 2022.

[81] N. Guha, J. Nyarko, D. E. Ho, C. Ré, A. Chilton, A. Narayana, A. Chohlas-Wood, A. Peters, B. Waldon, D. N. Rockmore, D. Zambrano, D. Talisman, E. Hoque, F. Surani, F. Fagan, G. Sarfaty, G. M. Dickinson, H. Porat, J. Hegland, J. Wu, J. Nudell, J. Niklaus, J. Nay, J. H. Choi, K. Tobia, M. Hagan, M. Ma, M. Livermore, N. Rasumov-Rahe, N. Holzenberger, N. Kolt, P. Henderson, S. Rehaag, S. Goel, S. Gao, S. Williams, S. Gandhi, T. Zur, V. Iyer, and Z. Li, "Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models," 2023.

[82] J. Niklaus, V. Matoshi, M. Stürmer, I. Chalkidis, and D. E. Ho, "Multilegalpile: A 689gb multilingual legal corpus," 2023.

[83] J. Cui, Z. Li, Y. Yan, B. Chen, and L. Yuan, "Chatlaw: Open-source legal large language model with integrated external knowledge bases," 2023.

[84] S. Shaghaghian, Luna, Feng, B. Jafarpour, and N. Pogrebnyakov, "Customizing contextualized language models forlegal document reviews," 2021.

[85] J. J. Nay, D. Karamardian, S. B. Lawsky, W. Tao, M. Bhat, R. Jain, A. T. Lee, J. H. Choi, and J. Kasai, "Large language models as tax attorneys: A case study in legal capabilities emergence," 2023.

[86] Z. Sun, "A short survey of viewing large language models in legal aspect," 2023.

[87] A. Deroy, K. Ghosh, and S. Ghosh, "How ready are pre-trained abstractive models and llms for legal case judgement summarization?" 2023.

[88] K. Singhal, S. Azizi, T. Tu, S. S. Mahdavi, J. Wei, H. W. Chung, N. Scales, A. Tanwani, H. Cole-Lewis, S. Pfohl, P. Payne, M. Seneviratne, P. Gamble, C. Kelly, N. Scharli, A. Chowdhery, P. Mansfield, B. A. y Arcas, D. Webster, G. S. Corrado, Y. Matias, K. Chou, J. Gottweis, N. Tomasev, Y. Liu, A. Rajkomar, J. Barral, C. Semturs, A. Karthikesalingam, and V. Natarajan, "Large language models encode clinical knowledge," 2022.

[89] P. Lewis, M. Ott, J. Du, and V. Stoyanov, "Pretrained language models for biomedical and clinical tasks: Understanding and extending the state-of-the-art," in *Proceedings of the 3rd Clinical Natural Language Processing Workshop*. Online: Association for Computational Linguistics, Nov. 2020.

[90] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "Biobert: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, p. 1234–1240, Sep. 2019.

[91] R. Luo, L. Sun, Y. Xia, T. Qin, S. Zhang, H. Poon, and T.-Y. Liu, "Biogpt: generative pre-trained transformer for biomedical text generation and mining," *Briefings in Bioinformatics*, vol. 23, no. 6, Sep. 2022.

[92] Y. Gu, R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, and H. Poon, "Domain-specific language model pretraining for biomedical natural language processing," *ACM Transactions on Computing for Healthcare*, vol. 3, no. 1, pp. 1–23, oct 2021.

[93] V. Liévin, C. E. Hother, and O. Winther, "Can large

language models reason about medical questions?" 2023.
[94] A. B. Abacha, W.-w. Yim, Y. Fan, and T. Lin, "An empirical study of clinical note generation from doctor-patient encounters," in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, 2023, pp. 2283–2294.
[95] Y.-N. Chuang, R. Tang, X. Jiang, and X. Hu, "Spec: A soft prompt-based calibration on performance variability of large language model in clinical notes summarization," 2023.
[96] M. Liu, D. Zhang, W. Tan, and H. Zhang, "DeakinNLP at ProbSum 2023: Clinical progress note summarization with rules and language ModelsClinical progress note summarization with rules and languague models," in *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*. Toronto, Canada: Association for Computational Linguistics, Jul. 2023.
[97] Y. Cao, F. Liu, P. Simpson, L. Antieau, A. Bennett, J. J. Cimino, J. Ely, and H. Yu, "Askhermes: An online question answering system for complex clinical questions," *Journal of biomedical informatics*, vol. 44, no. 2, pp. 277–288, 2011.
[98] K. Singhal, T. Tu, J. Gottweis, R. Sayres, E. Wulczyn, L. Hou, K. Clark, S. Pfohl, H. Cole-Lewis, D. Neal *et al.*, "Towards expert-level medical question answering with large language models," *arXiv preprint arXiv:2305.09617*, 2023.
[99] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," 2020.
[100] E. Nguyen, E. Kortukov, J. Song, and S. J. Oh, "Exploring practitioner perspectives on training data attribution explanations," *arXiv preprint arXiv:2310.20477*, 2023.
[101] M. Sallam, "Chatgpt utility in healthcare education, research, and practice: systematic review on the promising perspectives and valid concerns," in *Healthcare*, vol. 11, no. 6. MDPI, 2023, p. 887.
[102] S. L. Fleming, A. Lozano, W. J. Haberkorn, J. A. Jindal, E. P. Reis, R. Thapa, L. Blankemeier, J. Z. Genkins, E. Steinberg, A. Nayak *et al.*, "Medalign: A clinician-generated dataset for instruction following with electronic medical records," *arXiv preprint arXiv:2308.14089*, 2023.
[103] P. W. W. Koh, K.-S. Ang, H. Teo, and P. S. Liang, "On the accuracy of influence functions for measuring group effects," *Advances in neural information processing systems*, vol. 32, 2019.
[104] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, Q. Guo, M. Wang, and H. Wang, "Retrieval-augmented generation for large language models: A survey," 2024.
[105] S. Zhang, J. Wang, H. Jiang, and R. Song, "Locally aggregated feature attribution on natural language model understanding," *arXiv preprint arXiv:2204.10893*, 2022.
[106] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang, "Lost in the middle: How language models use long contexts," 2023.

XII. Supplementary Material

Appendix A
In-context data as the attribution domain

Due to the ubiquity of retrieval-augmented generation (104) and prompt engineering, data provided in-context is a highly relevant attribution domain. Determining the source boundaries of data provided in-context may be more difficult than for attribution domains that are corpora of predefined sources. This is discussed in more detail below.
Both corroborative and contributive attributions are possible to obtain from an attribution domain of the in-context data. While corroborative attributions are exactly the same for this setting, contributive attributions have an adjustment, as further discussed later in this section. Lastly, it is not guaranteed, even if it is likely, that an attribution from such a domain is simultaneously corroborative and contributive.

Defining sources within the attribution domain: Some forms of in-context data, such as documents retrieved from an external corpora and few-shot examples, contain natural structure to determine the segments that correspond to individual sources. Other forms of in-context data, however, may not have such structure for delineating the boundaries between sources. In order to designate such forms of in-context data as an attribution domain under our framework, it is necessary for a task designer to mark the segments of the in-context data that correspond to individual sources, according to the specifics of the task at hand.
For example, consider seeking contributive attributions to parts of the in-context data to gain insight into model behavior. Prior work refers to this setting as feature attribution (105), where each word of the in-context data is treated as an individual feature, or source. Here, we examine alternatives to defining each word of the input as a source. Consider the following examples of inputs, each containing different forms of in-context data, and their corresponding sources:

1) **Input with natural structure:** "Sentence: The moonlight gently illuminated the peaceful meadow.
Sentiment: Positive

Sentence: The sun cast harsh rays over the sweltering sand.
Sentiment: Negative

Sentence: The moonlight shone bright over the sparkling water.

Sentiment:"

s_0: "Sentence: The moonlight gently illuminated the peaceful meadow.
Sentiment: Positive"

s_1: "Sentence: The sun cast harsh rays over the sweltering sand.
Sentiment: Negative"

2) **Input without natural structure:** "What are the lunar phases? What is a lunar eclipse? Explain like I'm five."

s_0: "What are the lunar phases? What is a lunar eclipse?"

s_1: "Explain like I'm five."

Contributive attributions to in-context data: The contributive evaluators defined in Definition 5 pertain to sources on which the model was trained, either during pre-training or fine-tuning; they all pose the counterfactual scenario of a model trained without the source in question. If, however, the attribution domain is data provided in-context, rather than during pre-training or fine-tuning, one could pose an alternative counterfactual: what would the model loss or output be if the source had not been included in-context? With this alternative counterfactual in hand, the contributive evaluators discussed in Definition 5 could be extended to provide contributive attributions to an in-context attribution domain.

The implementation of contributive attributions in this setting is a very timely direction for future work. One potential challenge is highlighted by recent work which shows that the location of relevant information in the context impacts the model's usage of that information (106). Such an effect could be a confounding factor when measuring the difference between the original and counterfactual model losses or outputs.

Our discussion so far presents one view of in-context data attribution through the lens of feature attribution. We hope future work will develop various paradigms and accompanying methods for generating and verifying in-context data attributions.

Appendix B
Clause-level Explicatures

The following definitions formally define clause-level explicatures, which can be used as attributable units for corroborative attributions.

Definition 6. ***Clause-level Standalone Proposition***. *A standalone proposition, as defined by (9), that cannot be broken down into two or more non-overlapping standalone propositions.*

Consider the following examples:

Example 1: In 2010, Obama was the president of the United States and a father of two.

Example 2: In 2010, Obama was the president of the United States.

Application Area	Count
Code Generation	12
AI Tools	9
Content Creation	8
Persona/Assistants/QA	6
Healthcare	2
Legal	3
Education	1
Total	41

TABLE VI
LLM APPLICATION DOMAINS TARGETED BY SUMMER 2023 Y-COMBINATOR GENERATIVE AI VENTURES. 41/46 GENERATIVE AI CATEGORY COMPANIES LEVERAGE LLM OUTPUTS IN THEIR PRODUCT.

Example 3: In 2010, Obama was a father of two.

Example 4: In 2010, Obama was a father.

Example 1 is a standalone proposition, but is not clause-level. Examples 2 and 3 are the clause-level standalone propositions that compose Example 1. Example 4 is also a clause-level standalone proposition. Note that example 3 contains the example 4 but they are overlapping because they share the same information; this does not prevent example 3 from being clause-level.

Definition 7. ***Clause-level Explicature***. *The clause-level standalone propositions contained within a sentence-level explicature, as defined by (9).*

A clause-level explicature is a clause-level standalone proposition that is fully interpretable given only the wall clock time at which the query is made. We refer readers to (9) for the formal definition of a sentence-level explicature, which Definition B extends.

Appendix C
Entrepreneurial motivation for LLM Attributions: Y-Combinator Case Study

Our motivation for introducing this unified framework of attributions is driven by the rapidly advancing development of large language models to increasingly high-stakes domains. To understand how LLMs will likely be used in the near future, we examine ventures that have been proposed and funded based on LLM technology. As a case study, we look through the Summer 2023 Y-Combinator class[12] and examine the ventures that use LLMs, and highlight where attributions, both corroborative and contributive, may be important. Of the 46 companies listed under and *Generative AI*, 41 companies described the usage of large language models in various application domains (Table VI). The use cases (Section VII) and case studies (Section VIII) we study in our work are motivated by the different ways these companies have chosen to apply LLMs. Moreover, both corroborative attributions and contributive attributions may be helpful as these ventures and others begin deploying LLMs.

[12]https://www.ycombinator.com/blog/meet-the-yc-summer-2023-batch