Deductive Closure Training of Language Models for Coherence, Accuracy, and Updatability

Anonymous ACL submission

Abstract

While language models (LMs) can sometimes generate factually correct text and estimate truth values of individual claims, these generally do not reflect a globally coherent, manipulable model of the world. As a consequence, current LMs also generate incorrect or nonsensical content, and are difficult to edit and bring up to date. We present a method called Deductive Closure Training (DCT) that uses LMs themselves to identify implications of (and contradictions within) the text that they generate, yielding an efficient self-supervised procedure for improving LM factuality. Given a collection of seed documents, DCT prompts LMs to generate additional text implied by these documents, reason globally about the correctness of this generated text, and finally fine-tune on text inferred to be correct. Given seed documents from a trusted source, DCT provides a tool for supervised model updating; if seed documents are sampled from the LM itself, DCT enables fully unsupervised fine-tuning for improved coherence and accuracy. Across the CREAK, MQuAKE, and "Reversal Curse" datasets, supervised DCT improves LM fact verification and text generation accuracy by 3–26%; on CREAK, fully unsupervised DCT improves verification accuracy by 12%. These results show that LMs' reasoning capabilities during inference can be leveraged during training to improve their reliability.

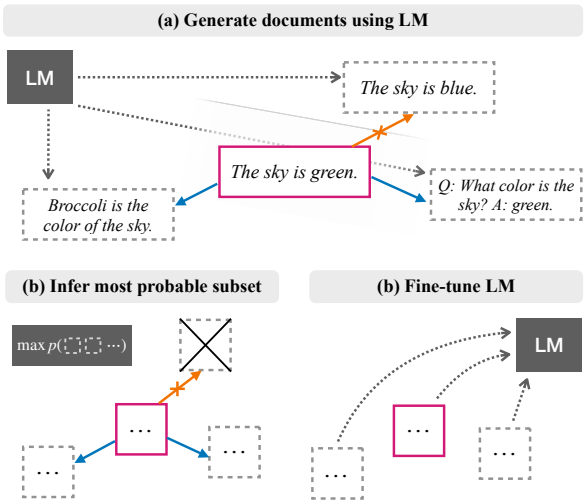

Figure 1: Overview of Deductive Closure Training (DCT). **(a)** To improve the coherence of language model predictions, we begin with a collection of seed documents, then use an LM to generate a set of documents implied by or contradicting these documents. **(b)** Next, we identify the generated documents most likely to be correct by finding the subset that is *most probable* and *logically consistent* (in this case excluding *the sky is blue*, because it contradicts the seed statement). **(c)** Finally, we fine-tune the LM on the selected subset of documents. While this example shows DCT applied to a **supervised model updating** application (where the seed statement is a new counterfactual assertion provided by a user as in most editing benchmarks), DCT can also be used for **unsupervised model improvement** by sampling seed statements from the LM itself.

1 Introduction

There is increasing interest in using language models (LMs) as sources of information and tools for fact verification (Porter, 2023). But today's LMs cannot robustly perform either task: they are prone to generating factually incorrect information, contradict themselves, and are difficult to update with new information (Honovich et al., 2021a; Liska et al., 2022; Sun et al., 2023; Gilson et al., 2023).

Even if LMs are imperfect judges of factuality, however, they are quite reliable models of factual relations *between* pieces of text: they can identify logical and probabilistic relationships between statements (Williams et al., 2017), and generate text based on new information provided as input (Yehudai et al., 2024). For example, an LM that cannot answer *How old was Charlie Chaplin when he died?* may nonetheless answer correctly when prompted with *Charlie Chaplin lived between 1889 and 1977*, and recognize that this statement contradicts the claim *Charlie Chaplin lived in the 21st*

century. How can we leverage LMs' ability to reason about relations between claims to improve (and control) the text that LMs themselves generate?

Conceptually, standard supervised objectives cause LMs to assign high probability to statements in their training data, but not necessarily these statements' logical consequences. Additional reasoning is required to determine the **deductive closure** of a training set (Armstrong, 1973)—the complete collection of inferences that can be made given the information initially available. An alternative procedure is needed to ensure that LMs assign high probability to a complete and consistent set of facts when they are trained and fine-tuned.

In this paper, we propose a new LM fine-tuning procedure we call **Deductive Closure Training (DCT)**, which leverages inference-time reasoning as a source of training-time supervision. At high level, given seed text (which may be provided externally or LM-generated), DCT uses an LM to identify additional text *implied by* or *contradicting* this text, reasons globally about which portions of seed and generated text are most likely to be correct given this context, and finally fine-tunes on inferred-correct text. This approach builds on a large body of recent work (Mitchell et al., 2022b; Kassner et al., 2023; Hase et al., 2023) on inference-time procedures for improving models' factual correctness, showing that these techniques may be used at training time as well.

DCT may be applied in several different ways depending on the source of seed documents. If these are drawn from a trusted factual source, DCT may be used to perform **supervised adaptation** for factuality. If documents contain new information to be inserted into an LM, DCT provides tool for **model updating** (or "editing"; De Cao et al., 2021). Finally, if seed documents are generated by the model itself, DCT enables **fully unsupervised fine-tuning** of models for improved accuracy.

We demonstrate the effectiveness of DCT across three domains: fact verification (CREAK benchmark; Onoe et al., 2021), question answering with new information (on the MQuAKE benchmark; Zhong et al., 2023), and a synthetic test of edit propagation (on the "Reversal Curse" benchmark; Berglund et al., 2023). On these tasks, unsupervised and supervised applications of DCT improve accuracy by up to 12% and 26%, respectively. These results show that, with little or no data, LM-generated supervision can be leveraged to improve LMs' coherence, accuracy and updatability.

2 Related Work

DCT builds on several recent techniques for improving model accuracy via inference-time computation or training-time self-supervision.

Bootstrapping accuracy during inference A growing body of research adopts techniques that bootstrap language model performance at inference time. Tafjord et al. (2022); Bostrom et al. (2022); Weir and Van Durme (2022) and Jung et al. (2022) build self-guided semantic chains of reasoning to support inference. Suzgun et al. (2022) propose a set of procedures that bin model-generated candidate answers by semantic equivalence and later uses aggregated probabilities to select the highest ranked predictions, analogous to self-consistency (Wang et al., 2023) for textual outputs. Finally, recent work has shown promise in improving coherence by conditioning language models on relevant reference texts through retrieval augmentation (Mitchell et al., 2022a; Akyürek et al., 2023). Our approach builds on this line of work by using inference-time techniques to generate supervision.

Training for accuracy LMs greatly benefit from training or post-training techniques for improving accuracy, including instruction-tuning (Sanh et al., 2022), learning from feedback (Ouyang et al., 2022) and loss truncation (Kang and Hashimoto, 2020). Closest to our approach is the work of Hase et al. (2023) which leverages graph-structured representations of model "beliefs" to train a hypernetwork for model editing. DCT aligns with this thread in improving model training; it differs by requiring minimal or no external supervision.

Self-training Past work has also studied leveraging LMs themselves for performance improvements (Pan et al., 2023). Several studies use external tools (Schick et al., 2023), binary feedback (Pang et al., 2023; Liu et al., 2023) and natural language feedback (Bai et al., 2022) to improve capability or reduce harms. Others propose actuality and consistency metrics, which might be used for filtering bad answers in retrospect (Honovich et al., 2021b; Wang et al., 2020; Honovich et al., 2022). Related to such approaches are methods that perform multiple inference attempts and aggregate them to get a more consistent answer (Wang et al., 2022; Yoran et al., 2023). Padmanabhan et al. (2023) fine-tune LMs on self-generated text without explicit implication generation or logical inference. Of immediate relevance to the current work,

Li et al. (2023) and a concurrent study by Tian et al. (2023) use LM-generated factuality labels to rank or filter LM-generated data for fine-tuning; by contrast, DCT uses LMs to explicitly extrapolate from LM-generated or externally provided information, providing a single framework for both supervised model updating and unsupervised improvement.

3 Method

3.1 Preliminaries

Given a **language model** p_{LM} that places a probability distribution over strings, our goal is to optimize p_{LM} so that it is **coherent** (if p_{LM} assigns high probability to statements P and Q, those statements must be logically compatible) and **complete** (if p_{LM} assigns high probability to P, and P implies Q, then p_{LM} must also assign high probability to Q). Together, these two properties imply that the LM is **closed** under logical deduction. Deductive closure is necessary condition for p_{LM} to be truthful, and approximate deductive closure is generally agreed to be an important feature of human-like belief (Armstrong, 1973).

Deductive closure training begins with a set of **seed documents** s_i, which may comprise facts from a trusted source, new information provided by a user, or even text generated by p_{LM} itself.[1] At a high level, DCT works by using p_{LM} to generate additional text implied by each seed document (i.e., true with high probability conditioned on s) or contradicting it. In Fig. 2, for example, the seed text *Country music originated in the United Kingdom)* is used to generate **statements** (*The UK is famous for country music*), **question-answer pairs** (*Q: Where did country music originate? A: England*) and even **multi-hop consequences** (*The steam train was invented in the UK; therefore, country music and the steam train were invented in the same country*). Once they have been generated, DCT again uses p_{LM} to reason about these documents as a set, identifying the subset of generated documents most likely to be true. Finally, DCT fine-tunes p_{LM} on documents in this inferred-true set. In the following sections, we describe each of these steps in more detail.

[1]While experiments in this paper focus on seed documents consisting of questions and declarative statements, this approach could be straightforwardly applied to larger pieces of text.

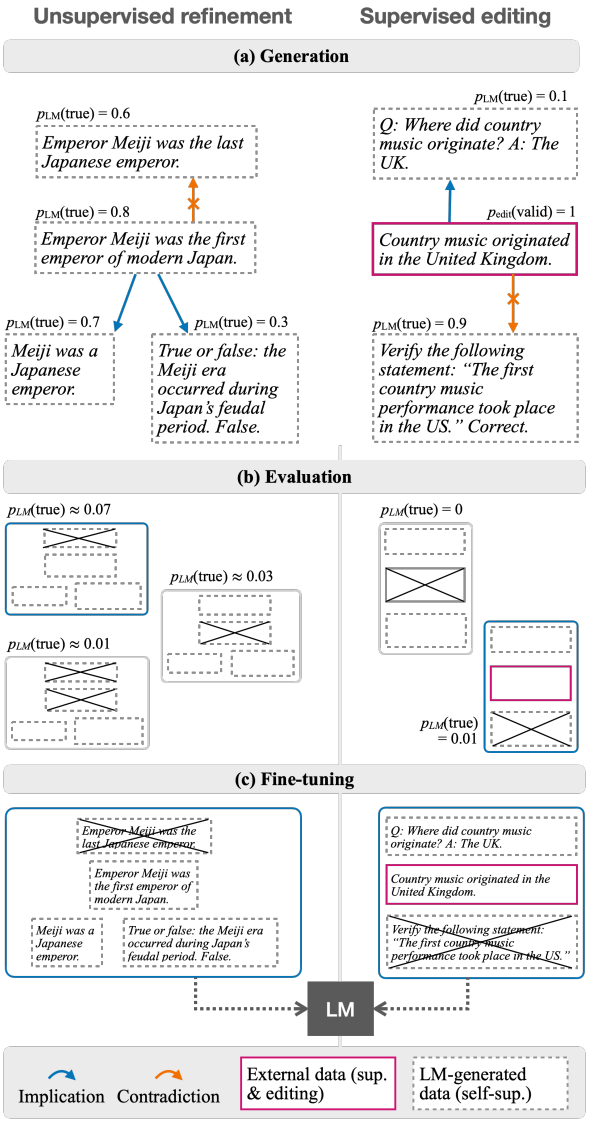

Figure 2: Detailed depiction of Deductive Closure Training. **(a)** Given an initial seed document (which may be generated from the LM, left; or supplied by a trusted source, right), DCT generates a set of related text implied by or contradicting the seed document. At the same time, it assigns a score to each generated document (including possibly the seed) denoting the probability that it is true. **(b)** Next, DCT identifies the subset of documents whose joint truthfulness score is highest, subject to the constraint that these documents are *logically coherent* (containing all implications and no contradictions). **(c)** Finally, the LM is fine-tuned on this set.

3.2 Document Generation

The first step of DCT is to generate a set of related documents for each seed document (Fig. 2a) using p_{LM}. Formally, we first construct a set of textual prompts that instruct the LM to generate other documents *entailed by* and *contradicted by* the input, along with 1–5 examples. We denote these prompts pr_{imp} and pr_{con} respectively (see Appendix D for full prompt text). Then, we construct a collection of **related documents** R_i for each seed document

$s_i, i \in \{1..n\}$ as:

$$
\begin{aligned}
R_i &= \mathcal{I}_i \cup \mathcal{C}_i \cup \{s_i\}, \\
\mathcal{I}_i &= \{r_{ij} \sim p_{\text{LM}}(\cdot \mid \text{pr}_{\text{imp}}, s_i)\}, \\
\mathcal{C}_i &= \{r_{ij} \sim p_{\text{LM}}(\cdot \mid \text{pr}_{\text{con}}, s_i)\}, \qquad (1)
\end{aligned}
$$

where $\mathcal{I}$ and $\mathcal{C}$ denote generated implications and contradictions respectively. (Other procedures for generating related documents are also possible, e.g. by simply prompting p_{LM} to generate *similar* text, as described in Section 5.1.) Note that the seed document s_i is included in R_i—this is crucial for detecting (and correcting) errors in the seed itself during unsupervised training.

This generation step may be followed by a **double-checking** step over R_i, in which we use the p_{LM} to verify whether s_i entails / contradicts r_{ij}, and discard all r_{ij} for which p_{LM} does not output *yes* with high probability (the prompt template is available in Appendix D). This step mirrors a variety of other recent methods in which models re-evaluate their initial answers (Suzgun et al., 2022).

3.3 Consistency Evaluation

The previous step produces a collection of documents in the "deductive neighborhood" of each seed document. These documents may be mutually contradictory, and we wish to identify the *subset* most likely to be collectively true. To identify this subset, we leverage p_{LM}'s ability to classify logical relations between documents, as well as the *prior* probability p_{LM} assigns to each document. For example, if it is true that *Emperor Meiji was the first emperor the Modern Japan*, it cannot be the case that *Emperor Meiji was the last Japanese emperor*; if the former statement is very likely to be true, then the latter is likely to be false.

Formally, we first associate with the seed document s_i and every generated document r_{ij} a **truth value** $t_{ij} \in \{0, 1\}$. Given an assignment of documents to truth values denoted by $T_i = \{t_{ij}\}$, we compute the LM's probability of T_i:

$$
p(T_i \mid R_i) = \prod_j p_{LM}(t_{ij} \mid r_{ij}). \qquad (2)
$$

We use prompting to estimate each $p_{\text{LM}}(t_{ij} \mid r_{ij})$: we first condition p_{LM} on a small set of document–label pairs where labels are one of {*True, False*}. Next, we use the normalized logits corresponding to the tokens *true* and *false* in the string $p_{\text{LM}}(r_{ij}$ *is true*$)$ and $p_{\text{LM}}(r_{ij}$ *is false*$)$, respectively. Refer to Appendix D for the prompt template. Next, we define a value assignment $T_i = \{t_{ij}\}$ to be **consistent** if all implications and contradictions are respected.

$$
c(T_i) = \prod_{j:r_{ij} \in \mathcal{I}_i} 1[t_i \to t_{ij}] \prod_{j:r_{ij} \in \mathcal{C}_i} 1[t_i \to \neg t_{ij}]
$$

where t_i denotes the truth value of the seed document, $1[a \to b]$ is 1 iff b is true or a is false, and $1[a \not\to b]$ is 1 iff b is false or a is false. We also provide an example for consistency computation across different truth value assignments in Table 6 in Appendix A. Finally, we select the most probable consistent assignment:

$$
T_i^* = \arg\max_T \; c(T \mid R_i) \cdot p(T \mid R_i)\,. \qquad (3)
$$

The procedure is depicted in Fig. 2b, with the highest-scoring truth value assignment shown in the blue-highlighted box.[2]

3.4 Language Model Fine-Tuning

Finally, we fine-tune p_{LM} only on the inferred-true documents, optimizing:

$$
\arg\max_\theta \sum_{i,\,j} t_{ij} \log p_{\text{LM}}(r_{ij})\,. \qquad (4)
$$

where θ parameterizes p_{LM}. In practice, we do not train p_{LM} to convergence, but instead for a fixed number of iterations.

3.5 Sources of Seed Data

Depending on how seed documents S are obtained, DCT-based fine-tuning may be used to improve models in several ways:

- **Unsupervised fine-tuning for coherence**: in this case, we sample the initial seed set *from* p_{LM} *itself*, e.g. simply by prompting it to generate a set of documents on a topic of interest.
- **(Semi-)supervised alignment with a trusted source**: in this case, the seed set comes from an external source of supervised data. If this data is known to be reliable, we fix each seed datum's truth value $t_i = 1$ during the evaluation step. This may be combined with the unsupervised procedure.

[2] For fact-verification tasks, it is possible to derive positive supervision from statements **marked as false**: if the consistency evaluation step infers that *Meiji was the last Japanese emperor* is incorrect, then we may generate a *correct* example of the form *Verify the following statement: Meiji was the last Japanese emperor. False*. We use this strategy for our experiments on fact verification.

- **Model updating, editing and continual learning**: in this case, as with supervised updating, we treat descriptions of desired edits as seed documents, fix these truth values for these seeds to 1, and fine-tune both on these documents and all their implications only.

Note that in the latter two cases (where we fix the truth value of seed documents to 1), the evaluation step is greatly simplified, and simply discards all generated documents that are not logically consistent with the seed. In the case of unsupervised learning, this evaluation step can (and empirically does) cause LMs to re-label sampled seed documents as well as conditionally generated ones.

Generalizations of DCT We remark that the procedure described above is the basic implementation of a family of DCT-like approaches, within which many more sophisticated procedures are possible—for example: **probabilistic DCT** (computing marginal statement probabilities rather than hard truth assignments), **contrastive DCT** (replacing Eq. (4) with an objective that encourages true statements to be assigned higher probability than false ones), and **multi-hop DCT** (generating not just direct implications of documents, but a wider graph of related ones).

4 Formal Analysis of DCT

At first glance, it may seem surprising that this procedure (especially in its unsupervised form) can improve LM accuracy using only LM-generated text. In this section, we describe a set of assumptions under which DCT is *guaranteed* to improve accuracy on certain inputs. We focus this analysis on generation and evaluation of (question, answer) pairs, but it could be extended to the other tasks considered in this paper as well.

Informally, suppose:

1. *Questions generated by the LM with high probability are likely to be correct.* (Intuitively, high-probability questions will be ones that occurred frequently in the training set, and are therefore more likely to be answered correctly; McCoy et al., 2023, though c.f. Lin et al., 2021.)

2. *Given a question, prompting an LM with a related, correct question–answer pair increases the probability of a correct answer.* (Intuitively, such prompts may steer models generally in the direction of truthfulness, as in Lin et al., 2021, and can provide concrete evidence useful for answering the new question.)

We wish to show that if these two conditions hold, DCT improves model performance.

For simplicity, we consider a minimal version of unsupervised DCT in which a single implication is generated from each seed statement, the check in Eq. (3) is not performed, and the LM is trained to convergence on data generated from an arbitrarily large number of seeds. Let q be some specific question of interest, let $p_{\text{LM}}(a^* \mid q)$ denote the probability that p_{LM} assigns the correct answer to q (before applying DCT), and let $p_{\text{DCT}}(a^* \mid q)$ be the probability that the LM assigns after DCT. Let (q_0, a_0) denote a (question, answer) pair generated as a *seed* document, and a_0^* specifically the *correct* answer to q_0. Finally, for convenience, define $p(q_0 \mid q) = \frac{p_{\text{LM}}(q|q_0)\, p_{\text{LM}}(q_0)}{\sum_{q_0'} p_{\text{LM}}(q|q_0')\, p_{\text{LM}}(q_0')}$ (this is the probability that the seed question was q_0 given that the sampled question was q), and $p(a_0 \mid q, q_0)$ via Bayes' rule analogously.

Proposition 1. *Suppose for some q that:*

1. $p(a_0^* \mid q, q_0) \geq p^*$. *(Conditioned on generating q during the document generation step of* DCT*, the probability that the generated answer to any seed question q_0 contains a correct answer is (uniformly) at least p^*.)*

2. $\mathbb{E}_{q_0|q}\, p_{\text{LM}}(a^* \mid q, q_0, a_0^*) \geq p_{\text{LM}}(a^* \mid q) \,/\, p^*$. *(In expectation, conditioning on a correct (q_0, a_0) pair increases the probability of generating a correct answer by at least $1/p^*$.)*

Then,

$$p_{\text{DCT}}(a^* \mid q) > p_{\text{LM}}(a^* \mid q)\,. \tag{5}$$

In other words, for any question q satisfying the two conditions above, unsupervised DCT *increases the probability that p_{LM} answers q correctly.*

Proof is given in Appendix E.

5 Experiments

We evaluate Deductive Closure Training on a set of benchmark tasks measuring fact verification, question answering with new information, and a diagnostic model editing dataset. We use Llama-2-7B in all experiments. Additional qualitative results are provided in Appendix C.

	Method	# Supervised	# Generated	Accuracy
Unsup.	**Prompting**	4	-	71.7 ±0.0
	DCT (Seed only)	-	93	**80.0** ±4.6
	DCT (Imp. + Cont.)	-	586	**83.5** ±3.0
	DCT (Imp. + Cont.) − Consistency Eval	-	586	77.5 ±2.8
	DCT (Imp. + Cont.) + Double-Check	-	313	**83.7** ±2.2
Sup.	**Fine-Tuning**	20	-	77.2 ±5.4
	DCT (Imp. + Cont.)	20	40	80.6 ±3.1
	DCT (Imp. + Cont.) + Double-Check	20	14	81.7 ±1.9
	Semi-Supervised*	20	586	**84.9** ±0.9
Transd.	**Graph-Inference**	-	14,342	77.7 ±0.4
	DCT (Rel.)	-	6,026	84.5 ±0.5
	DCT (Rel.) + (Imp. + Cont.)	-	28,711	80.3 ±0.4
	DCT (Rel.) + (Imp. + Cont.) + Double-Check	-	14,342	**85.5** ±0.1

Table 1: **Results on the CREAK validation set.** Accuracies are averaged over three seeds. Results that are not significantly worse than the best result in each block are made bold. *Indicates that training data includes generated statements from the Unsupervised DCT (Imp. + Cont.) experiment along with the supervised statements.

5.1 Fact Verification

Task and training details We first evaluate whether DCT improves the models' ability to classify factual claims. Our experiments use CREAK (Onoe et al., 2021), a dataset of claims about entities. We investigate four different learning settings: unsupervised, supervised, semi-supervised, and transductive, each using a different procedure for sampling seed documents. We report results on the CREAK development set. During DCT fine-tuning, we use a linear learning rate schedule until the *training* loss converges—this corresponds around 30 epochs for the majority of experiments unless otherwise indicated (see Appendix A for further details on experimental settings).

Evaluation and baselines Models are scored based on the fraction of claims they correctly label as true or false. For each condition, we compare to a state-of-the-art baseline. For unsupervised DCT, the baseline is an ordinary few-shot prompt. For supervised DCT, the baseline fine-tunes the LM on the provided true statements. For transductive DCT, we also compare to an inference-time baseline *Graph-Inference* similar to those described by Mitchell et al., 2022b and Kassner et al., 2023, which generates implications and contradictions for each test example, performs reasoning as in Eq. (3), then directly outputs the inferred truth value for the example (with no fine-tuning). Unlike past work, we use the base LM to generate these graphs rather than a specialized pre-trained implication generation model. All results are presented in Table 1.

Results: Unsupervised DCT To generate seed documents, we query p_{LM} 10 times, each time prompting the model to generate 10 diverse claims and sampling with a temperature of 0.9. We filter out the duplicate claims before continuing to sample implications and contradictions. The full method substantially outperforms a few-shot prompting baseline, and may outperform ablated versions of DCT that fine-tune only on seed statements assigned a high prior probability (labeled "seed only" in Table 2) or that do not perform the logical inference step described in Section 3.3 (labeled "− Consistency Eval").

For these unsupervised experiments, we perform an additional evaluation specifically aimed at measuring logical *coherence* as well as factual accuracy. Here we use the contrast set in CREAK, which comprises 250 pairs of lexically similar examples with opposite truth values (e.g. *Zendaya was raised in the US* and *Zendaya was raised in Scotland*). In addition to accuracy, we compute the fraction of pairs that are labeled *Both True* (indicating incoherence) and *Both Correct*.

Here, DCT not only improves correctness but also reduces the number of incoherent predictions, decreasing the probability that p_{LM} judges two contradictory statements to both be correct.

Results: Supervised & Semi-supervised DCT In the supervised case (Table 1), we utilize a small set of externally provided claims and associated ground-truth labels to initialize DCT seed nodes. We sample 20 claims from the CREAK training set and filter those labeled as `true` to use as our seed documents D. For semi-supervised learning, we pool together data generated following the unsupervised and supervised settings for fine-tuning.

All variants of DCT improve over an ordinary

Method	Both True ↓	Both Correct ↑	Acc. ↑
Prompting	34.4	36.8	63.2
DCT (Seed only)	20.4	47.6	72.2
DCT (Cont.)	**12.0**	47.6	72.0
DCT (Imp. + Cont.)	19.2	**49.6**	**73.0**

Table 2: **Logical coherence (Both True) and factuality (Both Correct) for unsupervised DCT on the CREAK contrast set.** DCT not only increases accuracy, but decreases the number of logically incoherent predictions (in which p_{LM} assigns labels two contradictory statements as both true).

fine-tuning baseline; interestingly, examples generated supervisedly and self-supervisedly are complementary, such that semi-supervised learning improves over both results.

Results: Transductive DCT The previous evaluations assumed a strict train / test split. Here we study the behavior of DCT in a "transductive" setting (Gammerman et al., 1998) in which we have access to *unlabeled* claims from the evaluation set while updating the model. For each of the 1,371 claims in the validation set, we generate seed text by prompting the LM to generate a set of *related* claims, which are then used to generate additional implications and contradictions. In addition to the inference-time baseline described above, these experiments compare to an ablated version of DCT that trains only on the generated related claims.

As in other experiments, DCT outperforms the inference-time reasoning baseline as well as the related-text-only ablation.

5.2 Model Updating and Question Answering

Task and training details Language models often hallucinate wrong information and rapidly become out-of-date after initial training. As a consequence, there has been increased interest in specialized continual learning (or "model editing") procedures for updating LMs with new information without full re-training. A key desideratum is LMs should not simply assign high probability to the new fact, but all of its *consequences*: if we wish to update an LM encode the fact that the current U.K. prime minister is not Boris Johnson but Rishi Sunak, the LM should also produce text consistent with the fact that the current P.M.'s wife is not Carrie Johnson but Akshata Murthy. Past work has found that fine-tuning on edits, as well as many specialized editing procedures, fail to propagate such information.

Our experiments on this task use the counterfactual subset from MQuAKE (Zhong et al., 2023) dataset, which evaluates models on their ability to answer questions about new information not provided in their training sets. To apply DCT, we take as seed documents the text of the new information to be inserted into the model. During the generation phase, models are prompted to combine this information with other background knowledge related to the same topic (see Appendix D for prompting details), producing what we term *Correlative Implications*. Finally, because MQuAKE is a question answering dataset, we convert each generated statement into a question–answer pair using the LM, then fine-tune it on these pairs.

Evaluation and baselines We compare DCT to ordinary fine-tuning on new information and three state-of-the-art baseline approaches for model updating: a context distillation baseline by Padmanabhan et al. (2023), which fine-tunes LMs to behave out-of-context the same way they would with prompts containing the new information (see Appendix A for implementation details), a weight editing baseline by (Meng et al., 2023), and the retrieval baseline MeLLo (Zhong et al., 2023), which stores new text in an external memory. We evaluate the behavior of DCT and these baselines in settings where varying numbers of new pieces of information (between 10 and 1000) are provided, and report the model's accuracy at question answering.

Results As shown in Table 3, DCT significantly outperforms fine-tuning, fine-tuning on continuations, weight editing, and MeLLo (the previous state-of-the-art on MQuAKE). Using correlative implications systematically improves over simple implications. Combining the two sets improves on average over using either in all settings. Our qualitative analysis in Appendix C reveals that correlative implications contain about 50% more new information than standard implications.

5.3 Sanity Checks for LM Consistency

Task and training details In addition to naturalistic question asking tasks like MQuAKE, there has been recent interest in developing precise tests of LMs' ability to capture simple logical implications of new facts (e.g. assigning high probability to sentences of the form *B is A* after training on *A is B*). We investigate whether DCT can address these issues using the "Reversal Curse" benchmark (Berglund et al., 2023). We report results

Method	Number of Edits				
	10	20	50	100	1000
Retrieval-based					
MeLLo (Zhong et al., 2023)	15.3 ±4.4	18.3 ±3.5	12.0 ±1.0	12.1 ±0.7	11.4
Parameter-update-based					
Fine-tuning on Edits	0.7 ±1.5	9.0 ±5.4	5.5 ±1.8	6.7 ±1.2	4.1
FT on Continuations (Padmanabhan et al., 2023)	4.4 ±2.9	4.4 ±2.0	3.1 ±1.2	3.6 ±1.2	3.3
MEMIT (Meng et al., 2023)	11.1 ±2.9	11.7 ±5.1	6.0 ±0.7	1.1 ±0.6	0.6
DCT (Imp.)	20.0 ±6.7	20.5 ±6.1	14.0 ±5.2	10.7 ±1.9	8.1
DCT (Corr. Imp.)	**41.7** ±5.0	**29.4** ±6.8	**25.6** ±3.8	**14.2** ±1.2	12.3
DCT (Corr. Imp. + Imp.)	**30.0** ±6.7	**35.6** ±4.8	**15.6** ±7.3	**18.9** ±2.1	**15.4**

Table 3: **MQuAKE counterfactual subset results.** We provide average test set accuracy (standard errors are given in parentheses) across three seeds except for 1,000 where we evaluate only once. Results that are not significantly different from the best score are made **bold** (paired t-test $p \ll 0.05$). For each edit, there are 3 multi-hop test questions. Before fine-tuning we convert each edit into a question using prompting. In DCT (Corr. Imp.), we prompt the model to first produce related facts to the initial claim before generating implications.

on two evaluations: first, a set of celebrity parent–child pairs with training examples like *Jennifer Lawrence's mother is Karen Lawrence* and test examples *Who is the child of Karen Lawrence?*; second, a set of entity–description pairs with training examples like *Olaf Scholz was the ninth Chancellor of Germany* and cloze-style test examples *The ninth Chancellor of Germany is __*.

Evaluation and baselines For these experiments, we compare to the fine-tuning baseline used in the original work of Berglund et al. (2023) as well as the fine-tuning on continuations approach by Padmanabhan et al. (2023). We use training examples as seed statements, and generate implications using *the same prompt as* CREAK *experiments in 5.1*. While we expect that a DCT-type approach specifically tailored for this benchmark could trivially re-generate all the test examples, our experiments in this section aim to evaluate whether a general-purpose prompt can improve performance on a specific class of generalizations. Following Berglund et al. (2023), we report exact-match accuracy after removing punctuation and lower-casing. In this dataset, LMs are evaluated on a mix of questions and cloze completion tasks featuring both training statements and their reversed forms.

Results Results are shown in Table 4. DCT improves accuracy on reversed statements without significantly hurting performance on original questions. Notably, however, DCT with this general-purpose prompt does not completely solve this dataset, and we leave for future work the question of whether more extensive sampling or other procedures could further improve these results.

	Direction		
	Same	**Reverse**	**Average**
Child-to-Parent			
Fine-tuning	**95.3**	2.2	48.7
FT (Padmanabhan et al., 2023)	57.3	7.1	32.2
DCT (Imp.)	87.9	**48.3**	**68.1**
Person-to-Description			
Fine-tuning	**83.7**	3.7	43.7
FT (Padmanabhan et al., 2023)	54.3	**27.0**	40.7
DCT (Imp.)	81.3	10.7	**46.0**
Description-to-Person			
Fine-tuning	**99.7**	3.0	51.3
FT (Padmanabhan et al., 2023)	99.3	1.0	50.2
DCT (Imp.)	**99.7**	**15.7**	**57.7**

Table 4: **Reversal Curse benchmark results.** While this challenge remains far from solved, applying DCT (with the same prompt used for CREAK experiments) substantially improves accuracy.

6 Conclusion

We have described Deductive Closure Training (DCT), a supervision procedure that optimizes models toward deductive closure—encouraging them to assign high probability to a logically coherent set of factual assertions.By doing so, DCT also improves the truthfulness and updatability of models, substantially increasing accuracy on a variety of fact verification and editing datasets in both supervised and unsupervised conditions. More generally, these results show that some factual errors in LMs stem not from limitations of their training data, but limitations of training algorithms. By using LMs themselves to reason about relationships between (and implications of) their predictions, they can be made more accurate with little or no additional supervision.

Limitations

While Deductive Closure Training (DCT) could in principle be applied to arbitrary graphs of relations between statements, here we have applied it only to a single layer of implications of seed data. All datasets used for evaluation involve English text, and it is possible that DCT behaves differently in different languages. Even within English, it is possible that exhibits systematic biases or differences in accuracy for certain types of factual content.

Ethical Considerations

While our experiments have focused on using DCT as a tool for bringing LMs into alignment with reliable sources, these techniques could also be used to optimize LMs toward generation of (logically consistent) false facts, increasing their effectiveness as tools for generation of misinformation.

References

Afra Akyürek, Eric Pan, Garry Kuwanto, and Derry Wijaya. 2023. DUnE: Dataset for unified editing. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1847–1861, Singapore. Association for Computational Linguistics.

David Malet Armstrong. 1973. *Belief, Truth and Knowledge*. Cambridge University Press.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. 2023. The reversal curse: Llms trained on" a is b" fail to learn" b is a". *arXiv preprint arXiv:2309.12288*.

Kaj Bostrom, Zayne Sprague, Swarat Chaudhuri, and Greg Durrett. 2022. Natural language deduction through search over statement compositions. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4871–4883, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

A. Gammerman, V. Vovk, and V. Vapnik. 1998. Learning by transduction. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, UAI'98, page 148–155, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Aidan Gilson, Conrad W Safranek, Thomas Huang, Vimig Socrates, Ling Chi, Richard Andrew Taylor, David Chartash, et al. 2023. How does chatgpt perform on the united states medical licensing examination? the implications of large language models for medical education and knowledge assessment. *JMIR Medical Education*, 9(1):e45312.

Peter Hase, Mona Diab, Asli Celikyilmaz, Xian Li, Zornitsa Kozareva, Veselin Stoyanov, Mohit Bansal, and Srinivasan Iyer. 2023. Methods for measuring, updating, and visualizing factual beliefs in language models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2714–2731, Dubrovnik, Croatia. Association for Computational Linguistics.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. TRUE: Re-evaluating factual consistency evaluation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3905–3920, Seattle, United States. Association for Computational Linguistics.

Or Honovich, Leshem Choshen, Roee Aharoni, Ella Neeman, Idan Szpektor, and Omri Abend. 2021a. q^2: Evaluating factual consistency in knowledge-grounded dialogues via question generation and question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7856–7870, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Or Honovich, Leshem Choshen, Roee Aharoni, Ella Neeman, Idan Szpektor, and Omri Abend. 2021b. q^2: Evaluating factual consistency in knowledge-grounded dialogues via question generation and question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7856–7870, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Jaehun Jung, Lianhui Qin, Sean Welleck, Faeze Brahman, Chandra Bhagavatula, Ronan Le Bras, and Yejin Choi. 2022. Maieutic prompting: Logically consistent reasoning with recursive explanations. In *Proceedings of the 2022 Conference on Empirical*

Methods in Natural Language Processing, pages 1266–1279, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Daniel Kang and Tatsunori B. Hashimoto. 2020. Improved natural language generation via loss truncation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 718–731, Online. Association for Computational Linguistics.

Nora Kassner, Oyvind Tafjord, Ashish Sabharwal, Kyle Richardson, Hinrich Schuetze, and Peter Clark. 2023. Language models with rationality. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14190–14201, Singapore. Association for Computational Linguistics.

Xiang Lisa Li, Vaishnavi Shrivastava, Siyan Li, Tatsunori Hashimoto, and Percy Liang. 2023. Benchmarking and improving generator-validator consistency of language models. *arXiv preprint arXiv:2310.01846*.

Stephanie C. Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. In *Annual Meeting of the Association for Computational Linguistics*.

Adam Liska, Tomas Kocisky, Elena Gribovskaya, Tayfun Terzi, Eren Sezener, Devang Agrawal, D'Autume Cyprien De Masson, Tim Scholtes, Manzil Zaheer, Susannah Young, et al. 2022. Streamingqa: A benchmark for adaptation to new knowledge over time in question answering models. In *International Conference on Machine Learning*, pages 13604–13622. PMLR.

Hao Liu, Carmelo Sferrazza, and Pieter Abbeel. 2023. Languages are rewards: Hindsight finetuning using human feedback. *arXiv preprint arXiv:2302.02676*.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

R Thomas McCoy, Shunyu Yao, Dan Friedman, Matthew Hardy, and Thomas L Griffiths. 2023. Embers of autoregression: Understanding large language models through the problem they are trained to solve. *arXiv preprint arXiv:2309.13638*.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2023. Mass editing memory in a transformer. *The Eleventh International Conference on Learning Representations (ICLR)*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. 2022a. Memory-based model editing at scale. In *International Conference on Machine Learning*, pages 15817–15831. PMLR.

Eric Mitchell, Joseph Noh, Siyan Li, Will Armstrong, Ananth Agarwal, Patrick Liu, Chelsea Finn, and Christopher Manning. 2022b. Enhancing self-consistency and performance of pre-trained language models through natural language inference. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1754–1768, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yasumasa Onoe, Michael JQ Zhang, Eunsol Choi, and Greg Durrett. 2021. CREAK: A dataset for commonsense reasoning over entity knowledge. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Shankar Padmanabhan, Yasumasa Onoe, Michael J.Q. Zhang, Greg Durrett, and Eunsol Choi. 2023. Propagating knowledge updates in lms through distillation. *Advances in Neural Information Processing Systems*, 36.

Liangming Pan, Michael Stephen Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. 2023. Automatically correcting large language models: Surveying the landscape of diverse self-correction strategies. *ArXiv*, abs/2308.03188.

Jing-Cheng Pang, Pengyuan Wang, Kaiyuan Li, Xiong-Hui Chen, Jiacheng Xu, Zongzhang Zhang, and Yang Yu. 2023. Language model self-improvement by reinforcement learning contemplation. *ArXiv*, abs/2305.14483.

Jon Porter. 2023. Chatgpt active user count revealed at openai developer conference. The Verge. Accessed: January 1, 2024.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer:

Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*.

Jiuding Sun, Chantal Shaib, and Byron C Wallace. 2023. Evaluating the zero-shot robustness of instruction-tuned language models. *arXiv preprint arXiv:2306.11270*.

Mirac Suzgun, Luke Melas-Kyriazi, and Dan Jurafsky. 2022. Prompt-and-rerank: A method for zero-shot and few-shot arbitrary textual style transfer with small language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2195–2222, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Oyvind Tafjord, Bhavana Dalvi Mishra, and Peter Clark. 2022. Entailer: Answering questions with faithful and truthful chains of reasoning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2078–2093, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D Manning, and Chelsea Finn. 2023. Fine-tuning language models for factuality. *arXiv preprint arXiv:2311.08401*.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, Online. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Nathaniel Weir and Benjamin Van Durme. 2022. Dynamic generation of interpretable inference rules in a neuro-symbolic expert system. *arXiv preprint arXiv:2209.07662*.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Asaf Yehudai, Boaz Carmeli, Yosi Mass, Ofir Arviv, Nathaniel Mills, Assaf Toledo, Eyal Shnarch, and Leshem Choshen. 2024. Genie: Achieving human parity in content-grounded datasets generation. *International Conference of Learning Representations*, abs/2401.14367.

Ori Yoran, Tomer Wolfson, Ben Bogin, Uri Katz, Daniel Deutch, and Jonathan Berant. 2023. Answering questions by meta-reasoning over multiple chains of thought. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5942–5966, Singapore. Association for Computational Linguistics.

Zexuan Zhong, Zhengxuan Wu, Christopher Manning, Christopher Potts, and Danqi Chen. 2023. MQuAKE: Assessing knowledge editing in language models via multi-hop questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15686–15702, Singapore. Association for Computational Linguistics.

A Experimental Details

We use the `Llama-2-7B-hf` checkpoint provided by HuggingFace Transformers library for all of our experiments. Code to reproduce the experiments will be made publicly available. While developing the codebase, the authors used GitHub Copilot via Visual Studio Code.

Generation We sample at temperature 0.6 and top-p 0.9 for all samples except for the set of seed documents for the unsupervised experiment in Table 1 where we used temperature 0.9 to obtain a diverse set of initial documents.

Training For fine-tuning we use the LoRA implemention via the PEFT library (Hu et al., 2022; Mangrulkar et al., 2022) and set rank to 8, alpha to 32 and dropout to 0.1. In the absence of a held-out development set, we set the learning rate to 0.0001 throughout, batch size to 4 and train for 30 epochs by default. We find that training loss typically converges after 30 epochs with the exception of the supervised experiments in Table 1 for which we train for 60 epochs. The transductive setting for CREAK results in substantially more training documents, hence we train only for 1 epoch. We use a linear learning rate scheduler with 100 warm up steps and AdamW optimizer. For fact verification training, we use weighted sampling as the class distribution is sometimes unbalanced.

Editing experiments We use the MQuAKE-CF subset from Zhong et al. (2023) and evaluate only on the multi-hop questions. Padmanabhan et al. (2023) proposes two techniques to introduce model updates based on fine-tuning: simple fine-tuning on continuations conditioned on the edit statement (which we call *FT on Continuations*) and context distillation on continuations. We find the former approach–fine-tuning the model on the continuations when the model is conditioned on the edit sequence–to perform better on MQuAKE than the latter. Hyperparameters used for MEMIT are available in Appendix A. For validation we use a set of held-out 50 edits.

B Details for DCT

In Table 6, we consider a small graph consisting of one seed node (r_i), one implication (r_{i1}) and one contradiction (r_{i2}). In the beginning, there are 8 candidate truth value assignment yet not all assignments are consistent within e.g. If r_i if true, then r_{i1} must be true and r_{i2} must be false. When computing the most probable assignment in Eq. (3), we only consider consistent assignments.

Parameter	Value
`layers`	[3, 4, 5, 6, 7]
`clamp_norm_factor`	4.0
`layer_selection`	all
`fact_token`	subject_last
`v_num_grad_steps`	25
`v_lr`	5e-1
`v_loss_layer`	31
`v_weight_decay`	0.5
`kl_factor`	0.0625
`mom2_adjustment`	true
`mom2_update_weight`	15000
`mom2_dataset`	wikipedia
`mom2_n_samples`	100000
`mom2_dtype`	float32

Table 5: MEMIT hyperparameters.

Truth Value Assignment (T_i)			
Seed	Implication	Contradiction	Consistency $c(T_i)$
T	T	T	0
T	T	F	1
T	F	T	0
T	F	F	0
F	T	T	1
F	T	F	1
F	F	T	1
F	F	F	1

Table 6: Consistency evaluations candidate truth value assignments for a small graph of three nodes: one seed, one implication and one contradiction documents.

C Qualitative Analysis

To better understand how DCT improves LM performance, we manually annotated about 350 generations from various experiments to assess whether (1) double-checking improves the precision of generated implications and contradictions; (2) whether DCT incorporates model internal knowledge when making new conclusions; and (3) whether generated text includes non-trivial new inferences.

Double-checking We evaluated whether the double-checking following DCT (Imp. + Cont.) improves precision. In the supervised setting for CREAK, we annotated 100 implications and contradictions generated using DCT (Imp. + Cont.). We found that 74 of these are valid. The double-checking procedure removes about 2/3 of generations, resulting in 33. Among these, 27 are valid, raising the ratio of correct statements predicted by

the model from 76% to 82%.

Incorporating previous information The MQuAKE subset used in our experiments comprises difficult multi-hop questions. Hence, generations that incorporate existing information about the entities mentioned in the edit are especially useful. We compare the set of implications generated using the DCT (Imp.) and DCT (Corr. Imp.). Respectively, only 30% and 36% of generations involve strict logical implications; however, 78% and 69% were judged to be plausible given the edit. Furthermore, 24% and 33% of the generations incorporate new information supplied by the LM. For example, given an edit *Chauncey Billups is associated with the sport of pesäpallo*, the LM uses background knowledge *Pesäpallo is popular in Finland* to generate *Chauncey Billups was born in Finland.*

Novelty of inferences Lastly, we find that most implications made by the model on the "Reversal Curse" dataset are paraphrases or are trivial (*Jennifer Lawrence's mother is Karen Lawrence* → *Jennifer Lawrence has a mother*) but some add world knowledge to the implication (*Sadie Frost's mother is Mary Davidson* → *Mary Davidson is the mother of a British actress*, where the LM itself has supplied the knowledge about Sadie Frost). While generating implications, DCT often (but not always) generates test-set-like reversed implications on its own: the model reverses 22% of the statements of the form *X's parent is Y*, 43% of statements of the form *the person with property X is Y*, but only 6% of statements of the form *Person X has property Y*. These findings suggest a strong bias toward generating text that starts with the person as opposed to the description. In general, most generated extensions are fluent, different from the source, and sometimes contain new information.

D Prompt Templates

We use a set of fixed prompts to generate our graphs, calculate model-estimated probability for the correctness of a given statement, generating a set of seed documents and automatically converting statements into questions which are available in Tables 7 to 10.

Table 7: **Implication & contradiction prompt templates.**

Procedure	Prompt

Implication	`List three implications of the given claims.`
	`Claim: Cleopatra was the last active ruler of the Ptolemaic Kingdom of Egypt between 51 to 30 BC.`
	`Logical implications:`
	`1. Cleopatra was one of the rulers of the Ptolemaic Kingdom of Egypt.`
	`2. Egypt had a female ruler during the Ptolemaic Kingdom age.`
	`3. Ptolemaic Kingdom of Egypt ended on 30 BC.`
	
	`Claim: {claim}`
	`Logical implications:`

Implication (MQuAKE)	`List five logical implications of the given claims.`
	
	`Claim: Stephen Hawking was born and raised in Russia.`
	`Logical implications:`
	`1. Stephen Hawking has knowledge of Russian language.`
	`2. The head of the country where Stephen Hawking was born is Vladimir Putin.`
	`3. The country where Stephen Hawking was born is Russia.`
	`4. Stephen Hawking is a Russian citizen and has a Russian passport.`
	`5. The city where Stephen Hawking was born is in Russia.`
	
	`Claim: {claim}`
	`Logical implications:`

Correlative Implication (MQuAKE)	`Given a main claim, list five related facts, and then logical implications of the claim and related fact.`
	
	`Main Claim: Stephen Hawking was born and raised in Russia.`
	`Related Facts:`
	`1. The language of Russia is Russian.`
	`2. The head of Russia is Vladimir Putin.`
	`3. Russia is on the continents of Asia and Europe.`
	`4. The capital of Russia is Moscow.`
	`5. The currency of Russia is Russian ruble.`
	
	`Implications:`
	`1. Stephen Hawking has knowledge of Russian language.`
	`2. The head of the country where Stephen Hawking was born is Vladimir Putin.`
	`3. The country where Stephen Hawking was born is on the continents of Europe and Asia.`
	`4. The capital of Stephen Hawking's home country is Moscow.`
	`5. Stephen Hawking has used Russian ruble growing up.`
	
	`Main Claim: {claim}`
	`Related Facts:`

Table 8: **Prompt templates for double-checking, generating similar claims and estimating model-assigned truth value.**

Procedure	Prompt

Implication (Double-Check)	`For the given pair of claims you need to decide if the first one implies the second. Give your final verdict at the end. Here are some examples.`
	
	`The tallest building in the world is taller than 800 metres.`
	`The tallest building in the world is taller than 700 metres.`
	`Discussion: If something is taller than 800 then it is necessarily taller than 700.`
	`Final Verdict: Implies.`
	
	`Orange is a fruit.`
	`Orange is an apple.`
	`Discussion: Not all fruit are apples so orange being a fruit does not imply that is also an apple.`
	`Final Verdict: Does not imply.`
	
	`{claim1}`
	`{claim2}`
	`Discussion:`

Contradiction (Double-Check)	`For the given pair of claims you need to decide if they are contradictory or not. Give final verdict at the end. Here are some examples.`
	
	`Claim 1: The tallest building in the world is taller than 800 metres.`
	`Claim 2: The tallest building in the world is shorter than 1000 metres.`
	`Reasoning: A building can be both taller than 800 and shorter than 1000.`
	`Final Verdict: Not contradictory.`
	
	`Claim 1: Orange is a fruit.`
	`Claim 2: Orange is a vegetable.`
	`Reasoning: Fruit and vegetable are disjoint categories.`
	`Final Verdict: Contradictory.`
	
	`Claim 1: {claim1}`
	`Claim 2: {claim2}`
	`Reasoning:`

Estimating Truth Value	`Label the following statements according to whether or not they are true:`
	`World War II began in 1965. Label: false`
	`Alan Alda is an actor. Label: true`
	`The moon is made of obsidian. Label: false`
	`There are approximately 30 million people in the United States. Label: false`
	`Dracula was written by Bram Stoker. Label: true`
	`{claim} Label:`

Table 9: **Prompt templates for generating contradictions, related statements (used in the transductive setting) and unsupervised seed document generation.**

Procedure	Prompt

Contradiction	`Given a claim, generate three other very similar-looking but CONTRADICTING claims.`
	
	`Claim: Cleopatra was the last active ruler of the Ptolemaic Kingdom of Egypt between 51 to 30 BC.`
	`Similar but contradicting claims:`
	`1. Cleopatra was the first active ruler of the Ptolemaic Kingdom of Egypt.`
	`2. Cleopatra was the last active ruler of the Ptolemaic Kingdom of Egypt between 51 to 30 AD.`
	`3. Cleopatra was the daughter of the last active ruler of the Ptolemaic Kingdom of Egypt.`
	
	`Claim: {claim}`
	`Similar but contradicting claims:`

Similar claims ($\mathrm{pr}_{\mathrm{rel}}$)	`Generate five related factual statements on the same topic as the given claim. Note that the given claim may or may not be correct. However, the generated statements should each be correct and different.`
	`Claim (may be true or false): Neil Armstrong and Buzz Aldrin became the first humans to land on the Mars.`
	`Related Correct Facts:`
	`1. Apollo 11 was the first manned mission to land on the moon.`
	`2. Neil Armstrong was the first person to step on the moon.`
	`3. No human has been to Mars yet.`
	`4. Neil Armstrong and Buzz Aldrin were the first humans to land on the moon.`
	`5. Neil Armstrong and Buzz Aldrin were the first humans to walk on the moon.`
	`Claim (may be true or false): {claim}`
	`Related Correct Facts:`

Unsupervised seed claims	`Generate ten examples of factual claims. List your claims in separate lines.` `1.`

Table 10: **Prompt template for converting model-generated statements into questions. We re-use the original statements as the corresponding answers.**

Procedure	Prompt

Conversion to questions	`Sentence: Kate Winslet is a citizen of the UK.`
	`Question: Which country is Kate Winslet a citizen of?`
	`Sentence: Ukraine is a country in Europe.`
	`Question: Which continent is Ukraine in?`
	`Sentence: The country where Priyanka Chopra is from is India. The capital of India is New Delhi.`
	`Question: What is the capital of the country where Priyanka Chopra is from?`
	`Sentence: sentence`
	`Question:`

E Proof of Proposition 1

At optimality, $p_{\mathrm{DCT}}(a^* \mid q)$ (the probability that the updated LM assigns to the correct answer) will be the probability of a^* given q marginally over all generated seed documents:

$$p_{\mathrm{DCT}}(a^* \mid q) = \sum_{q_0, a_0} p_{\mathrm{LM}}(a^* \mid q, q_0, a_0)\, p(a_0 \mid q_0, q)\, p(q_0 \mid q)\ .$$

We may decompose this according to whether the generated seed pair is itself correct:

$$= \sum_{q_0} p(q_0 \mid q) \Big[p_{\mathrm{LM}}(a^* \mid q, q_0, a_0^*)\, p(a_0^* \mid q, q_0) + \sum_{a_0' \neq a_0^*} p_{\mathrm{LM}}(a^* \mid q, q_0, a_0')\, p(a_0' \mid q_0, q)\, p(q_0 \mid q) \Big]$$

(where a_0^* denotes the correct answer to q_0)

$$\geq \sum_{q_0} p(q_0 \mid q)\, p_{\mathrm{LM}}(a^* \mid q, q_0, a_0^*)\, p(a_0^* \mid q, q_0)\ .$$

By assumption 1:

$$\geq \sum_{q_0} p(q_0 \mid q) p_{\mathrm{LM}}(a^* \mid q, q_0, a_0^*)\, p^*$$

$$= p^*\, \mathbb{E}_{q_0 \mid q}\, p_{\mathrm{LM}}(a^* \mid q, q_0, a_0^*)\ .$$

By assumption 2:

$$\geq p_{\mathrm{LM}}(a^* \mid q)\ . \quad \square$$