Semantically Structured Causal Systems Knowledge for Causal Question-Answering

Anonymous ACL submission

Abstract

Language models often require external knowledge for causal reasoning in QA settings and employ public knowledge sources such as ConceptNet. Causality is inherently contextual, requiring models to reason about causal relations within specific situations. However, existing knowledge sources present causal facts as isolated universal triples (e.g., ⟨lit match; cause-effect; fire⟩) with limited contextual details. As a result, these repositories often fail to capture the causal context necessary for reasoning applications. To address this gap, we introduce CASK-Schema and CASK-Db. Inspired by mechanism theory, CASK-Schema formalizes causal systems and augments causal facts with relevant temporal, influential, and quantitative relations. We then construct CASK-Db, a public causal knowledge base of ~5.4K synthetically enriched causal systems. Our extensive empirical evaluation demonstrates that CASK-Db improves causal QA performance across six tasks in two knowledge augmentation settings: knowledge injection (average improvement of 14% / 9pp) and retrieval-augmented zero-shot QA (average improvement of 13% / 6pp).

1 Introduction

As AI research advances, language models and LLMs increasingly serve as conjecture machines, capable of generating hypotheses, producing explanations, and reasoning about the world (Valentino et al., 2021; Valentino and Freitas, 2022). They are applied to a wide range of causal reasoning tasks, including question answering (Hassanzadeh et al., 2020), scientific discovery(Wysocki et al., 2024), and medical diagnosis (Zhou et al., 2024). A crucial component of such systems is external causal knowledge extracted from public knowledge bases such as CauseNet (Heindorf et al., 2020) and ConceptNet(Speer et al., 2017). Prior work has shown that augmenting language models with external causal knowledge can improve accuracy on causal

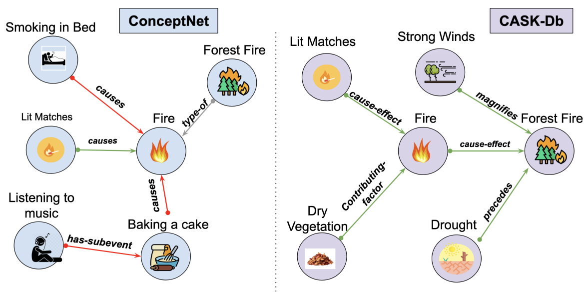

Figure 1: Extracted causal explanations from ConceptNet and CASK-Db (ours) for forest fires.

QA tasks (Sharp et al., 2016; Dalal et al., 2021; Hosseini et al., 2022). However, existing knowledge repositories are fundamentally limited in their ability to capture causal knowledge for effective causal reasoning.

Knowledge bases consist of declarative knowledge, i.e., explicit facts about the world that are assumed to be objectively and universally true (Zhong et al., 2023). Facts are encoded and stored as independent knowledge triples (e.g., ⟨lit match; cause-effect; fire⟩). Knowledge graphs are constructed by extracting triples sharing head and tail entities and linking multiple facts to represent more complex concepts such as event chains. Public knowledge bases assume monotonicity (facts are temporally invariant unless explicitly updated), universality (context-independent and globally true), and an open-world model (while incomplete, new facts are inferable) (Levesque and Lakemeyer, 2001). However, these assumptions are misaligned for causal reasoning as causality is non-monotonic and highly context-dependent (Bochman, 2007). For instance, new causes of fire have been discovered and lit matches are not the cause of all fires. Under open-world assumptions, all stored facts are considered true, and extracting causal graphs from knowledge bases risks spurious context, as not all adjacent facts are causally relevant. Consider the extracted causal explanations for forest fires from ConceptNet in Figure 1.

ConceptNet erroneously includes *baking a cake* and *smoking in bed* as causes of forest fires while also indirectly implying *listening to music*. Further, the graph lacks causal context, such as contributing factors (e.g., dry vegetation and strong winds) and temporal details (e.g., preceding droughts). Generally, existing knowledge bases contain sparse or even no causal context. For instance, CauseNet contains over 11 million cause-effect triples but no other contextual relations. Finally, there is no unified semantic definition of causal context, as knowledge bases use arbitrary relations, limiting interoperability. **To address these limitations, we introduce Causal Systems Knowledge (CASK) Schema and Database (Db).**

We take inspiration from mechanism theory (Johnson and Ahn, 2017), which posits that causality must be understood systematically. Integral to this perspective are *causal systems*, which specify systemic interactions between events, entities, and concepts that produce predictable causal outcomes. With CASK-Schema, we formalize causal systems into a semantic schema to enrich causal facts with influential, temporal, and other relevant causal contexts. We then construct CASK-Db, the first causal systems knowledge base consisting of ∼5.4K synthetically enriched causal systems. Finally, we validate CASK-Db through extensive empirical evaluations in two knowledge-augmentation settings. **In the knowledge injection experiments, CASK-Db with our *SyntheticQA* method improves causal QA performance on average by 14% (9pp). In the retrieval-augmented generation (RAG) setting, CASK-Db increases zero-shot causal QA accuracy on average by 13% (6pp).** All resources are publicly available on HuggingFace Datasets [1] and GitHub [2] to support future research.

2 Related Work

Causal Knowledge Public repositories of causal knowledge are generally populated by automatically mining causal relations from public knowledge sources such as Wikipedia or published news articles (Khoo et al., 1998; Hassanzadeh et al., 2020) using linguistic cues and lexical triggers (Girju et al., 2007; Neeleman and van de Koot, 2012), extracted, and converted into knowledge triples. Public causal knowledge sources include CauseNet, ConceptNet, ATOMIC (Sap et al., 2019), and Wikidata (Vrandečić and Krötzsch, 2014). *PublicKB* is constructed from these knowledge sources as a baseline for our experiments.

Synthetic Data LLMs parameterize factual and relational knowledge, which can be extracted to support downstream applications (Petroni et al., 2019). LLM-generated synthetic data have substantially improved QA accuracy and elicited emergent capabilities in smaller LLMs. For instance, Taori et al. (2023) generated 52K instruction-following examples to enable Llama 7B (Touvron et al., 2023) to match the performance of the 175B-parameter GPT-3 model. Li et al. (2023) created synthetic textbooks to train high-performance "small" LLMs. Mukherjee et al. (2023) introduced *progressive learning*, iteratively generating more complex training examples for LLM training. Our pipeline for constructing CASK-Db was inspired by these approaches and uses generative AI to produce semantically structured causal systems.

Knowledge-Augmented Causal QA Prior studies found that augmenting language models with external knowledge can improve causal QA performance. The most common approach involved injecting external knowledge during continual pretraining by modifying the *MLM* objective (Devlin et al., 2019; Sun et al., 2020) to strategically mask causal (Hosseini et al., 2022) or commonsense triples (Sap et al., 2019). (Sharp et al., 2016; Dalal et al., 2021) explored enriching language models with derived causal knowledge graph embeddings. However, prior work primarily evaluated causal QA on a single dataset (COPA (Gordon et al., 2012)) and did not examine the influence of causal knowledge on distinct causal reasoning tasks. Our empirical evaluation provides a comprehensive analysis by assessing multiple causal QA datasets to systematically identify the strengths and limitations of external causal knowledge across various causal reasoning tasks.

3 Semantically Structured Causal Systems

CASK-Schema is strongly inspired by cognitive theories. Induction theory (Griffiths, 2017) posits that humans acquire causal knowledge through lived experiences and education, cognitively organizing it into ontological schemas rather than as enumerated facts. Schematic representations are memory-efficient, composable, and hierarchical,

[1] anonymous_url
[2] https://anonymous.4open.science/r/cask-paper-D67D

Relation	Type	Description	Domain/Range
cause-effect	causal	Establishes a direct causal link between concepts.	$D \subseteq \{\mathcal{A}, \mathcal{X}, \mathcal{E}, \mathcal{V}, \mathcal{S}\}$ $R \subseteq \{\mathcal{A}, \mathcal{X}, \mathcal{E}, \mathcal{V}, \mathcal{S}\}$
has-contributing-factor	influential	Auxiliary factors that influence but do not directly cause an outcome.	$D \subseteq \{\mathcal{V}, \mathcal{S}\}$ $R \subseteq \{\mathcal{A}, \mathcal{X}, \mathcal{V}, \mathcal{S}\}$
reacts-to	influential	Captures influential factors.	$D \subseteq \{\mathcal{A}, \mathcal{E}, \mathcal{S}\}$ $R \subseteq \{\mathcal{X}, \mathcal{A}, \mathcal{E}, \mathcal{V}, \mathcal{S}\}$
has-intent	motivation	Indicates the purpose or intention behind an action.	$D \subseteq \{\mathcal{X}\}$ $R \subseteq \{\mathcal{A}\}$
magnifies	quantification	Increases the severity or likelihood of an event or action.	$D \subseteq \{\mathcal{A}, \mathcal{X}, \mathcal{V}\}$ $R \subseteq \{\mathcal{A}, \mathcal{X}, \mathcal{V}\}$
mitigates	quantification	Decreases the intensity or likelihood of an event or action.	$D \subseteq \{\mathcal{A}, \mathcal{X}, \mathcal{V}\}$ $R \subseteq \{\mathcal{A}, \mathcal{X}, \mathcal{V}\}$
precedes	temporal	Establishes temporal precedence.	$D \subseteq \{\mathcal{X}, \mathcal{V}\}$ $R \subseteq \{\mathcal{X}, \mathcal{V}\}$
has-subevent	temporal	Captures successive events in a process.	$D \subseteq \{\mathcal{X}, \mathcal{V}\}$ $R \subseteq \{\mathcal{V}\}$

Table 1: CASK-Schema defines a set of relations and causal concepts to formally represent the influential, temporal, and contextual aspects of a *causal system*. The causal concepts specified in the domain and range include abstracts ($\mathcal{A}$), actions ($\mathcal{X}$), entities ($\mathcal{E}$), events ($\mathcal{V}$), and systems ($\mathcal{S}$).

enabling inference in novel situations. Cognitive schemas represent causal knowledge as mechanism systems that capture covariation patterns, temporal cues, and causal context. Mechanism systems are structured interactions between physical and abstract events, entities, and processes that produce predictable causal outcomes, allowing causality to be plausibly inferred and generalized to novel scenarios.

CASK-Schema formalizes these mechanisms by enriching causal triples with influential, temporal, and contextual relations to produce semantically structured causal systems. Where possible, we derive our relations from existing knowledge sources to ensure greater interoperability with established knowledge bases (see Appendix A.5).

3.1 CASK-Schema

We formally define a *causal system* CS as a semi-closed set of knowledge triples: $\text{CS} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n\}$. Each triple, $T = (h, r, t)$, consists of a head h, relation r, and tail t. The head and tail elements belong to the set of causal concepts $\mathcal{CC}$, where $\mathcal{CC} \in \{\text{Actions}, \text{Abstracts}, \text{Entities}, \text{Events}, \text{Systems}\}$. Relations r semantically link causal concepts to encode structured causal knowledge. A complete definition of CASK-Schema is provided in Table 1.

Causal concepts define the components of a causal system, ranging from discrete elements (e.g., entities) to broader constructs (e.g., abstracts). **Actions** are intentional activities performed by agents that create changes and influence outcomes. **Abstracts** are non-physical elements that shape actions, events, and entities. **Entities** are agents, objects, or things that participate in events, initiate actions, or are affected by them. **Events** are discrete occurrences that establish causal context at specific times and locations. Finally, **systems** are structured interactions among entities, events, and actions that produce well-defined outcomes.

Relations semantically connect causal concepts, capturing influential, temporal, quantitative, and motivational aspects of causal interactions. The **cause-effect** relation establishes direct causal links between concepts. Influential factors are represented by the **has-contributing-factor** and **reacts-to** relations, where *reacts-to* describes responses or reactions, and *has-contributing-factor* identifies auxiliary factors that influence outcomes without directly causing them. Temporality is modeled through the **precedes** and **has-subevent** relations, derived from (Mostafazadeh et al., 2016). We introduce **magnifies** and **mitigates** to describe factors that amplify or diminish the intensity or likelihood of actions, events, and abstracts. Finally, the **has-intent** relation specifies the purpose behind an action.

3.2 CASK-DB Construction

Figure 2 illustrates the CASK-Db construction pipeline, which uses generative AI to enrich causal triples and applies validation steps to ensure the veracity and quality of synthetically enriched causal systems. The pipeline consists of three stages: (1) **seeding**, (2) **generation**, and (3) **refinement**.

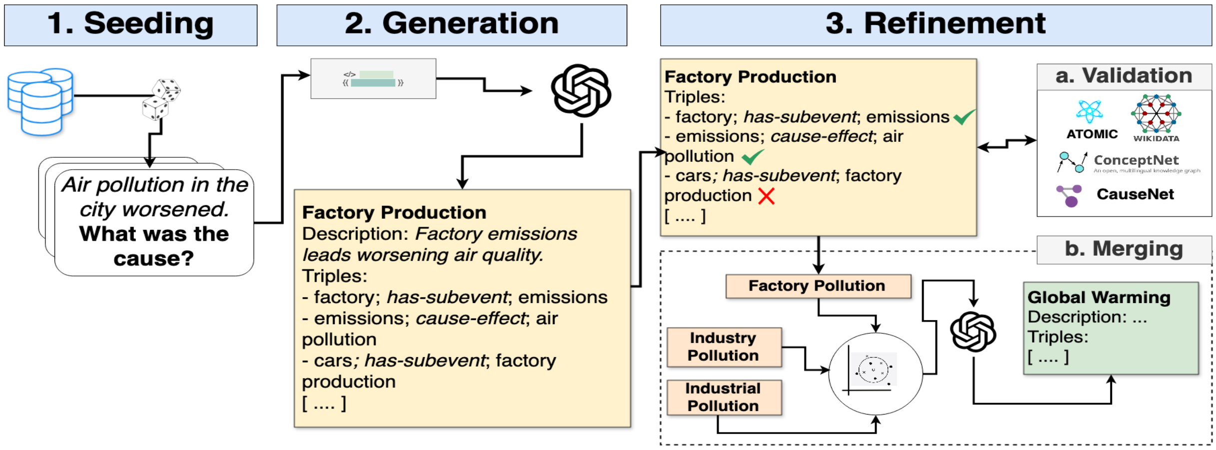

Figure 2: Pipeline for constructing CASK-Db.

1. Seeding: This stage identifies a broad and diverse range of causal contexts for generation. We randomly sample 6,000 causal questions from the training splits of CALM-Bench (Dalal et al., 2023) to prevent test set leakage. For reproducibility, we use a fixed random seed (42) for sampling.

2. Generation: A generative LLM is prompted to produce causal systems aligned with our schema. The model considers the seed question, identifies the underlying causal system and requisite knowledge, and then generates the causal system in alignment with CASK-Schema. Each output contains a title, a one-sentence description, and a set of knowledge triples describing the system. To facilitate post-processing, we instruct the LLM to use predefined headers. We use GPT-3.5 Turbo (Brown et al., 2020)[3] for generation in our implementation. See Appendix A.6.1 for generation prompt.

3. Refinement: Lastly, we validate the generated knowledge and merge similar causal systems. For validation, we construct a vector database of ground truth knowledge. A triple is considered valid if at least two distinct matches in the semantic store have a cosine similarity of 0.75 or higher. Further details on the validation process are provided in Appendix A.6.2.

Overlapping and redundant causal systems are merged. To identify merge candidates, we use K-nearest neighbors to cluster the causal systems. Within each cluster, systems with a similarity of 0.80 or higher are selected for unification. A generative LLM is then prompted to merge them into a single causal system. Implementation details are provided in Appendix A.6.3.

[3]https://platform.openai.com/docs/models/gpt-3-5-turbo

3.3 CASK-Db Details

Our final synthetic causal knowledge base consists of 5,450 causal systems, 32,638 unique knowledge triples, and 40,360 concepts. On average, each causal system contains 7 knowledge triples and 10 unique causal concepts. Of the causal systems in CASK-Db, 45% are science-related, 38% commonsense knowledge, and 17% pertain to social interactions. CASK-Db is publicly available under the Apache 2.0 license.

Quality analysis 150 causal systems, containing a total of 846 triples, were sampled for manual evaluation. We found that 4% of sampled systems and less than 1% of all triples contained errors, suggesting that most causal systems are high quality, logically correct, and factually accurate. Among the identified errors, 60% were named entity errors, where generated triples failed to generalize and included direct references to people or named locations. Logical errors, where head and tail entities were swapped, accounted for 18% of errors. Ambiguous entities and unresolved anaphora (e.g., "it," "they," "those") comprised 20% of errors. Finally, only 6% of errors involved incorrect facts. Addressing these errors remains an area for future work.

4 Empirical Evaluation

Our experiments aim to (1) validate the efficacy of CASK-Db as an external causal knowledge source for language models and LLMs and (2) assess the benefits and limitations of causal knowledge across distinct causal reasoning tasks. We evaluate CASK-Db in two knowledge augmentation settings: knowledge injection and RAG for zero-shot QA.

Type	Description	Example
Cause Comparison	Given competing causal contexts C_1 and C_2, the goal is to identify which context is most likely to produce effect E such that $C \Rightarrow E$.	There are two planets Glarnak and Bornak. Glarnak is experiencing global warming while Bornak is not. **Which planet is more likely to have more pollution in the atmosphere?**
Cause Prediction	Given an event description D, the question requires identifying the most likely cause C such that $C \Rightarrow D$.	Pollution in the city worsened? **What was the cause?**
Effect Comparison	Given competing event descriptions D_1 and D_2, the goal is to identify which event would most likely result from a provided cause C such that $C \Rightarrow D$.	There are two planets Glarnak and Bornak which have breathable atmospheres for humans. Glarnak's atmosphere has a higher concentration of CO2 in contrast to Bornak. **Which planet is more likely to have implemented environmental regulation policies?**
Effect Prediction	Given an event description D, the question requires identifying the most likely effect that results from D such that $D \Rightarrow E$.	The city is determined to control air pollution. **What is the effect?**
Effect Quantification	Given an event chain consisting of temporally ordered subevents $S_1, S_2, \ldots, S_n$ and a causal intervention I, the goal is to quantify the effect of the causal intervention on the event $Q(E\|I, S_{1\ldots n})$.	1. A seed is in soil. 2. The seed germinates. 3. The plant grows roots. 4. The plant grows out of the ground. 5: The plant flowers. 6: The flower produces fruit. 7: The fruit releases seeds. 8: The plant dies. **Suppose less pollution in the environment happens, how will it affect the overall population of plants?**

Table 2: Typology of common of causal causal reasoning tasks found in CALM-Bench.

4.1 Data

4.1.1 Causal Knowledge

CASK-Db (Section 3.3) is the primary causal knowledge resource evaluated in all experiments. For a fair comparison with public causal knowledge sources, we construct **PublicKB** as a baseline. PublicKB consists of 357,706 triples extracted from ATOMIC, CauseNet, ConceptNet, and Wikidata. In addition to cause-effect triples, we extract all analogous relations (e.g., *has-subevent*, *has-prerequisite*, etc.) that map to contextual relations in CASK-Schema (see Table 8).

PublicKB contains nearly 11× more triples than CASK-Db (∼357K vs.∼32K), yet we hypothesize that causal systems knowledge better aligns with the causal reasoning needs of language models and should improve downstream QA accuracy over PublicKB. Further details are provided in Appendix A.7.

4.1.2 Causal QA Tasks

We employ CALM-Bench (Dalal et al., 2023) as the source of causal QA tasks. CALM-Bench comprises six diverse QA benchmark datasets that require causal knowledge and reasoning. These tasks include abductive reasoning, commonsense causal reasoning, procedural reasoning, and reasoning over paragraph effects. Further details on the benchmark tasks are provided in Appendix A.3.

4.1.3 Causal Reasoning Typology

Due to the diversity of question formats and tasks encountered, we define a typology to categorize the common types of causal reasoning tasks in CALM-Bench. Questions are classified along two dimensions: *directionality* and *inferential requirements*. Directionality specifies whether the question seeks likely causes or effects. Inferential requirements define the type of causal reasoning needed to answer the question (e.g., comparing contexts or quantifying effects). Details of the typology and examples are provided in Table 2.

4.2 Experiment Details

4.2.1 Experiment Environment

All experiments were conducted on a single AWS EC2 g5.8xlarge instance[4], equipped with an NVIDIA A20 24GB GPU, 32 vCPUs, and 400GB of storage.

4.2.2 Knowledge Injection Experiments

Our experiments assess whether injected causal systems knowledge enhances causal QA performance and how pretraining strategy impacts downstream reasoning. Knowledge injection methods imbue language models with external knowledge to improve performance in knowledge-intensive tasks

[4] https://aws.amazon.com/ec2/instance-types/g5/

(Hu et al., 2023). The most common approaches mask knowledge triples, requiring the model to recover the masked elements during training (Sun et al., 2020; Lu et al., 2022). However, prior work has primarily evaluated these methods on factual QA rather than causal reasoning. We hypothesize that masking-based strategies are misaligned with causal QA and propose *SyntheticQA* as a more effective alternative.

Knowledge-Guided Pretraining Strategies. We explore two masking-based methods (**random masking** and **concept masking**) and introduce **SyntheticQA**. In *random masking*, knowledge triples are linearized using sentence templates and randomly masked during pretraining (Hosseini et al., 2022). *Concept masking* selectively masks specific elements (e.g., head entity) within a linearized sentence (Bosselut et al., 2019). *SyntheticQA* replaces masking with multiple-choice questions generated from causal system descriptions. During pretraining, the model is given a causal description as context and must answer an associated question. Implementation details are provided in Appendix A.8.2.

Experimental Setup. We use FLAN-T5 (Chung et al., 2022), a 250M parameter encoder-decoder model pretrained on 1.8K diverse tasks in the FLAN collection (Longpre et al., 2023), achieving SOTA performance across QA tasks. As a baseline, we evaluate the model before knowledge injection. Experiments involve finetuning for 5 epochs on pretraining examples from CASK-Db or PublicKB. After knowledge-guided pretraining, we checkpoint the model, further finetune it on the benchmark task, and report QA accuracy on the test set. The model is then reverted to the pretraining checkpoint to ensure only transferred knowledge from CASK-Db is measured. Training specifics are in Appendix A.8.3.

4.2.3 RAG Zero-Shot QA Experiments

Our experiments examine whether CASK-Db is broadly valuable for LLMs as an external resource for providing in-context causal knowledge in zero-shot causal QA. RAG (Lewis et al., 2020) has become the de facto method for augmenting LLMs with external knowledge, helping reduce hallucinations (Shuster et al., 2021) and improve domain-specific reasoning (Gao et al., 2024).

Experimental Setup. We implement a standard RAG system with a vector knowledge store, retrieval model, and a generative LLM for QA inference (Gao et al., 2024). First, we measure the baseline zero-shot capabilities of the evaluated LLMs by providing only the question. In RAG experiments, each causal system is treated as an independent knowledge record, linearized into paragraph descriptions, encoded as vectors, and stored in ChromaDB[5] using `multi-qa-mpnet-base-dot-v1`[6] for encoding and retrieval. During inference, the most relevant causal system is retrieved based on cosine similarity and provided as in-context evidence. For multiple-choice questions, LLMs return the corresponding letter; for open-ended questions, only exact matches are considered correct. QA accuracy is reported for all experiments. Further technical details are provided in Appendix A.10.

Evaluated LLMs. We evaluate CASK-Db using four diverse LLMs: Phi-2 3B (Li et al., 2023), Mistral 7B (Jiang et al., 2023), Llama 2 13B (Touvron et al., 2023), and GPT-3.5-Turbo. GPT-3.5-Turbo is accessed via the OpenAI API, while the other models are loaded with QLoRA (Dettmers et al., 2023) quantization for efficient inference. Quantization configurations and prompt templates are provided in Table 13 and Table 14.

5 Empirical Findings

5.1 Main Results

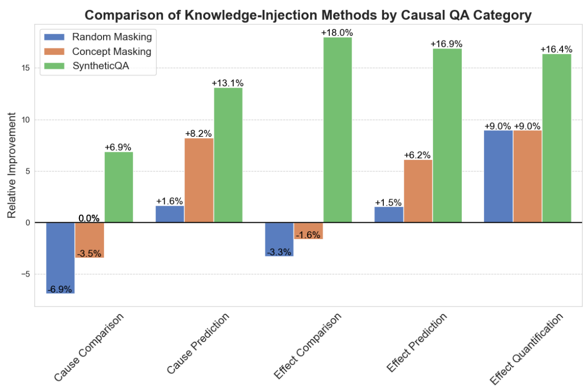

Figure 3: Comparison of pretraining strategies. Results are reported as relative changes from the finetuned baseline.

Knowledge injection results are reported in Table 3, and RAG results in Table 4. CASK-Db substantively improves causal QA accuracy in both augmentation settings, with an average relative improvement of 14% (9pp) using *SyntheticQA* for

[5] https://docs.trychroma.com/

[6] https://huggingface.co/sentence-transformers/multi-qa-mpnet-base-dot-v1

	Cause Comp.	Cause Pred.	Effect Comp.	Effect Pred.	Effect Quant.
Baseline	*0.58*	*0.61*	*0.61*	*0.65*	*0.67*
Knowledge: Public KB					
Random Masking	0.57	0.62	0.56	0.6	0.68
Concept Masking	0.57	0.6	0.58	0.67	0.71
SyntheticQA	0.5	0.66	0.58	0.7	0.7
Knowledge: CALM-KB (ours)					
Random Masking	0.54	0.62	0.59	0.66	0.73
Concept Masking	0.56	0.66	0.6	0.69	0.73
SyntheticQA	**0.62**	**0.69**	**0.72**	**0.76**	**0.78**

Table 3: Evaluation of CASK-Db and PublicKB in the knowledge injection setting across various pretraining strategies. Improvements over the finetuned baseline are shaded green, and regressions are shaded red.

	Cause Comp.		Cause Pred.		Effect Comp.		Effect Pred.		Effect Quant.	
	Base	+KB	Base	+KB	Base	+KB	Base	+KB	Base	+KB
GPT3.5	0.3	0.45	0.66	**0.72**	0.54	**0.62**	0.74	0.79	0.58	**0.66**
Llama 2 13B	0.57	**0.59**	0.47	0.61	0.58	0.59	0.50	0.65	0.51	0.53
Mistral 7B	0.48	0.52	0.68	**0.72**	0.60	**0.62**	0.76	**0.80**	0.46	0.48
Phi-2 3B	0.50	0.54	0.47	0.56	0.48	0.54	0.50	0.60	0.48	0.49

Table 4: Evaluation of CASK-Db for zero-shot QA with RAG. The "Base" column represents baseline zero-shot accuracy, while +KB reflects accuracy using the RAG pipeline with CASK-Db. Relative improvements over the baseline are shaded in green, while regressions are shaded in red.

knowledge injection and 13% (6pp) across all LLMs for zero-shot QA.

5.2 Knowledge Injection Findings

Which pretraining strategy best improves causal reasoning? A direct comparison of pretraining strategies for CASK-Db are provided in Figure 3. *SyntheticQA* is the only strategy that yields consistent improvements across all reasoning categories, increasing accuracy by an average of 14%, making it the preferred method for causal knowledge injection. In contrast, masking-based strategies improve causal reasoning by only 2% on average across tasks. Additionally, masking-based strategies tend to reduce accuracy in cause and effect comparison tasks, with an average performance regression of -4%, while offering modest improvements of 7% for cause prediction, effect prediction, and effect quantification. The results also indicate that *SyntheticQA* is particularly beneficial for effect-related reasoning, improving effect comparison by 18% and averaging a 17% gain for effect-related tasks compared to 10% for cause-related tasks.

To what extent does transferred causal knowledge affect reasoning? Transferred causal knowledge is most effective for cause prediction, effect comparison, effect prediction, and effect quantification, yielding an average accuracy gain of 16%, compared to just 6% for cause comparison. SyntheticQA's format may be limited for cause comparison as the questions are generated from single causal systems, whereas cause comparison requires multiple contexts.

We also observe a consistent directionality bias: effect-related tasks achieve higher accuracy than cause-related tasks both before and after knowledge injection (75% vs. 65% on average). This may stem from causal sufficiency challenges, where the space of possible causes is larger than the constrained space of effects. While causal knowledge injection improves reasoning, its effectiveness is limited by the model's ability to generalize across causal contexts.

How does CASK-Db compare to public sources of knowledge? A direct comparison between CASK-Db and PublicKB is shown in Figure 4. Despite being 30× smaller than PublicKB, CASK-Db is consistent and better improves downstream causal QA performance. Further PublicKB negatively impacts effect quantification decreasing accuracy by 10%.

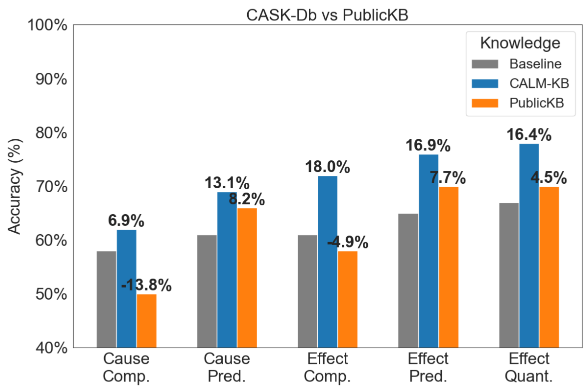

Figure 4: Comparison of CALM-KB to PublicKB

Does QA format impact causal knowledge transfer? Figure 5 compares the impact of QA format in SyntheticQA on accuracy. The results indicate that multiple-choice is the superior format, yielding an average relative gain of 14% compared to just 2% for open-ended QA. Moreover, the open-ended format reduces performance on the effect quantification task by 10%, limiting its effectiveness in cause prediction, effect comparison, and effect prediction tasks.

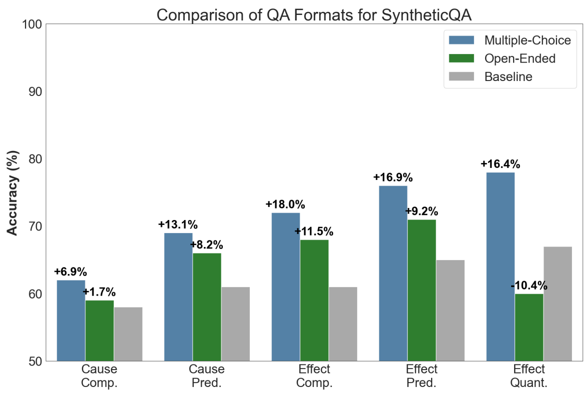

Figure 5: Comparison of multiple-choice vs open-ended as the QA format for SyntheticQA.

5.3 Zero-Shot Causal QA with RAG Findings

How Do LLMs Differ in Their Use of External Knowledge? In Figure 6, A. highlights that LLMs utilize causal knowledge differently, as relative improvements vary across reasoning categories. In B, we find that CASK-Db yields the highest gains in cause comparison, cause prediction, and effect prediction, with an average improvement of 16%. However, effect comparison and quantification see smaller gains, averaging 7%. Interestingly, while pretraining experiments showed greater improvements for effect prediction, knowledge-augmented LLMs exhibit the opposite trend.

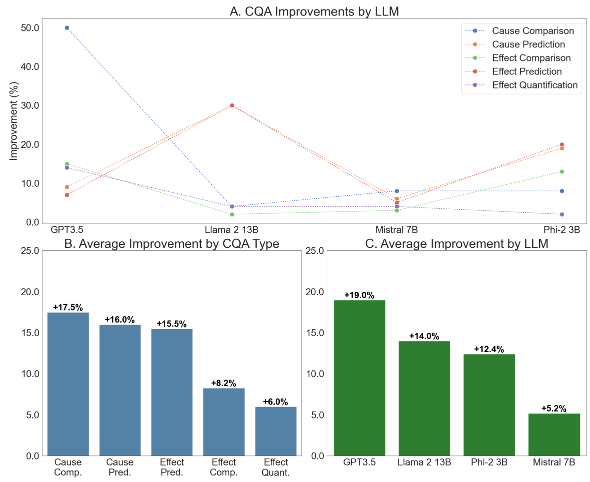

Figure 6: Patterns of LLM behavior when utilizing CASK-Db.

Does LLM scale affect causal knowledge utilization? Subfigure C shows that CASK-Db improves QA performance across all evaluated LLMs. While larger models generally utilize causal knowledge more effectively, this trend is inconsistent, as seen with Mistral 7B. GPT-3.5 benefits the most overall but performs the worst on cause comparison, indicating that LLM size does not directly correlate with causal knowledge utilization. Additionally, in certain tasks like cause prediction and effect comparison, smaller models (e.g., Mistral 7B) perform comparably to augmented GPT-3.5. These findings suggest that LLMs process causal knowledge differently depending on task structure and reasoning requirements.

6 Conclusion

We propose CASK-Schema, a semantic schema for formally representing *causal systems*, and introduce CASK-Db, a knowledge base of synthetically constructed causal systems. Our analysis demonstrates that CASK-Db enhances causal reasoning in language models across both knowledge injection and retrieval-based augmentation settings. We show that causal systems knowledge facilitates more effective knowledge transfer and improves reasoning over causal relationships. Additionally, our findings highlight differences in how LLMs utilize causal knowledge, revealing key challenges in aligning external knowledge with causal QA tasks. Our work establishes a foundation for future research on causal knowledge representation, causal question answering, and the systematic evaluation of causal reasoning in language models.

7 Limitations

We recognize several opportunities to improve our work and acknowledge the limitations of our methods and empirical evaluation. While we conduct extensive knowledge augmentation experiments to validate CASK-Db, further evaluation remains necessary. In the knowledge injection setting, all experiments are limited to FLAN-T5; future work could explore pretraining strategies across diverse architectures (e.g., BERT, DeBERTa) and model scales.

Our question templates primarily focus on cause and effect prediction, limiting the diversity of reasoning tasks. Future work could incorporate a broader range of question types and explore generative AI for synthetic question generation beyond template-based methods.

Our knowledge validation relies on indirect verification, evaluating triples independently rather than within full causal systems. An ideal approach would involve expert verification or an oracle system to assess factual accuracy. Future work could leverage high-performance LLMs like GPT-o for verification or introduce an entailment-based validation step using a fine-tuned natural language inference model to ensure contextual consistency.

Additionally, we do not align our generated causal systems with existing semantic knowledge graphs such as WikiData. Future work could enhance CASK-Db through entity linking, integrating causal concepts with structured public knowledge sources.

References

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen-tau Yih, and Yejin Choi. 2020. Abductive commonsense reasoning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Alexander Bochman. 2007. A causal theory of abduction. *Journal of Logic and Computation*, 17(5):851–869.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *Preprint*, arXiv:2210.11416.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang. 2021. Pre-training with whole word masking for chinese bert. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3504–3514.

Dhairya Dalal, Mihael Arcan, and Paul Buitelaar. 2021. Enhancing multiple-choice question answering with causal knowledge. In *Proceedings of Deep Learning Inside Out (DeeLIO): The Second Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*. Association for Computational Linguistics.

Dhairya Dalal, Paul Buitelaar, and Mihael Arcan. 2023. CALM-bench: A multi-task benchmark for evaluating causality-aware language models. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 296–311, Dubrovnik, Croatia. Association for Computational Linguistics.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Preprint*, arXiv:2305.14314.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Li Du, Xiao Ding, Kai Xiong, Ting Liu, and Bing Qin. 2022. e-CARE: a new dataset for exploring explainable causal reasoning. In *Proceedings of the 60th*

Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 432–446, Dublin, Ireland. Association for Computational Linguistics.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *Preprint*, arXiv:2312.10997.

Roxana Girju, Preslav Nakov, Vivi Nastase, Stan Szpakowicz, Peter Turney, and Deniz Yuret. 2007. SemEval-2007 task 04: Classification of semantic relations between nominals. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 13–18, Prague, Czech Republic. Association for Computational Linguistics.

Andrew Gordon, Zornitsa Kozareva, and Melissa Roemmele. 2012. Semeval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 394–398, Montréal, Canada. Association for Computational Linguistics.

Thomas L Griffiths. 2017. Formalizing prior knowledge in causal induction. *The Oxford Handbook of Causal Reasoning*, pages 115–126.

Oktie Hassanzadeh, Debarun Bhattacharjya, Mark Feblowitz, Kavitha Srinivas, Michael Perrone, Shirin Sohrabi, and Michael Katz. 2020. Causal knowledge extraction through large-scale text mining. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(09):13610–13611.

Stefan Heindorf, Yan Scholten, Henning Wachsmuth, Axel-Cyrille Ngonga Ngomo, and Martin Potthast. 2020. Causenet: Towards a causality graph extracted from the web. In *Cikm*. Acm.

Pedram Hosseini, David A. Broniatowski, and Mona Diab. 2022. Knowledge-augmented language models for cause-effect relation classification. In *Proceedings of the First Workshop on Commonsense Representation and Reasoning (CSRR 2022)*, pages 43–48. Association for Computational Linguistics.

Linmei Hu, Zeyi Liu, Ziwang Zhao, Lei Hou, Liqiang Nie, and Juanzi Li. 2023. A survey of knowledge enhanced pre-trained language models. *IEEE Trans. on Knowl. and Data Eng.*, 36(4):1413–1430.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. pages 2391–2401.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Samuel G. B. Johnson and Woo-kyoung Ahn. 2017. 127causal mechanisms. In *The Oxford Handbook of Causal Reasoning*. Oxford University Press.

Pride Kavumba, Naoya Inoue, Benjamin Heinzerling, Keshav Singh, Paul Reisert, and Kentaro Inui. 2019. When choosing plausible alternatives, clever hans can be clever. In *Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing*, pages 33–42, Hong Kong, China. Association for Computational Linguistics.

Christopher S. G. Khoo, Jaklin Kornfilt, Robert N. Oddy, and Sung-Hyon Myaeng. 1998. Automatic extraction of cause-effect information from newspaper text without knowledge-based inferencing. *Literary and Linguistic Computing*, 13:177–186.

Hector J. Levesque and Gerhard Lakemeyer. 2001. Knowledge bases as representations of epistemic states. In *The Logic of Knowledge Bases*. The MIT Press.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023. Textbooks are all you need ii: phi-1.5 technical report. *Preprint*, arXiv:2309.05463.

Kevin Lin, Oyvind Tafjord, Peter Clark, and Matt Gardner. 2019. Reasoning over paragraph effects in situations. pages 58–62.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: designing data and methods for effective instruction tuning. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *Preprint*, arXiv:1711.05101.

Yinquan Lu, Haonan Lu, Guirong Fu, and Qun Liu. 2022. Kelm: Knowledge enhanced pre-trained language representations with message passing on hierarchical relational graphs. *Preprint*, arXiv:2109.04223.

Nasrin Mostafazadeh, Alyson Grealish, Nathanael Chambers, James Allen, and Lucy Vanderwende. 2016. CaTeRS: Causal and temporal relation scheme for semantic annotation of event structures. In *Proceedings of the Fourth Workshop on Events*, pages 51–61, San Diego, California. Association for Computational Linguistics.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *Preprint*, arXiv:2306.02707.

Ad Neeleman and Hans van de Koot. 2012. The linguistic expression of causation.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2019. Atomic: an atlas of machine commonsense for if-then reasoning.

Rebecca Sharp, Mihai Surdeanu, Peter Jansen, Peter Clark, and Michael Hammond. 2016. Creating causal embeddings for question answering with minimal supervision. In *ACL 2016 Proceedings of Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, Aaai'17, page 4444–4451. AAAI Press.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. Ernie 2.0: A continual pre-training framework for language understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8968–8975.

Niket Tandon, Bhavana Dalvi, Keisuke Sakaguchi, Peter Clark, and Antoine Bosselut. 2019. WIQA: A dataset for "what if..." reasoning over procedural text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6076–6085. Association for Computational Linguistics.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2021. Augmented SBERT: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 296–310, Online. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Marco Valentino and André Freitas. 2022. Scientific explanation and natural language: A unified epistemological-linguistic perspective for explainable ai. *arXiv preprint arXiv:2205.01809*.

Marco Valentino, Ian Pratt-Hartmann, and André Freitas. 2021. Do natural language explanations represent valid logical arguments? verifying entailment in explainable nli gold standards. In *Proceedings of the 14th International Conference on Computational Semantics (IWCS)*, pages 76–86, Groningen, The Netherlands (online). Association for Computational Linguistics.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledge base. *Communications of the ACM*, 57:78–85.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara

Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Oskar Wysocki, Magdalena Wysocka, Danilo Carvalho, Alex Bogatu, and Andre Freitas. 2024. An LLM-based knowledge synthesis and scientific reasoning framework for biomedical discovery. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 355–364, Bangkok, Thailand. Association for Computational Linguistics.

Lingfeng Zhong, Jia Wu, Qian Li, Hao Peng, and Xindong Wu. 2023. A comprehensive survey on automatic knowledge graph construction. *ACM Comput. Surv.*, 56(4).

Shuang Zhou, Zidu Xu, Mian Zhang, Chunpu Xu, Yawen Guo, Zaifu Zhan, Sirui Ding, Jiashuo Wang, Kaishuai Xu, Yi Fang, Liqiao Xia, Jeremy Yeung, Daochen Zha, Genevieve B. Melton, Mingquan Lin, and Rui Zhang. 2024. Large language models for disease diagnosis: A scoping review. *Preprint*, arXiv:2409.00097.

A Appendix

A.1 Reproducibility

CASK-Db and all relevant code are made publicly available. CASK-Db can be accessed on Hugging Face Datasets at `anonymous_url`, and the code is available on GitHub. For all experiments, we set a global seed of 42 to ensure reproducibility.

A.2 Dataset Usage and Licenses

We use all datasets in accordance with their respective licenses. Furthermore, we provide CASK-Db under the Apache 2.0 license, which permits broad academic and commercial use to encourage further exploration of causal knowledge representation.

A.3 CALM-Bench Task Descriptions

Task-specific dataset details and an overview of CALM-Bench can be found in Table 7. We summarize the tasks below.

Abductive Natural Language Inference (aNLI) (Bhagavatula et al., 2020) is an abductive reasoning task over narratives of social situations. Given a sequential pair of social observations, the model must predict which of the two provided hypotheses best explains the observations.

Choice of Plausible Alternatives (COPA) (Gordon et al., 2012) is a commonsense causal reasoning task. Given a premise, the goal is to select the most likely cause or effect from a pair of options. (Kavumba et al., 2019) introduced 500 additional training examples in Balanced-COPA to mitigate corpus-level artifacts that language models could exploit during fine-tuning.

COSMOS QA (Huang et al., 2019) is a multiple-choice QA task requiring social commonsense knowledge. Given a narrative about people in everyday situations, the goal is to identify the most plausible cause or effect within the story.

E-Care (Du et al., 2022) consists of two causal reasoning tasks. The first, similar to COPA, requires identifying the most likely cause or effect of a given premise. The second involves generating a causal explanation for the correct answer. We consider only the first task as part of CALM-Bench.

Reasoning over Paragraph Effects (ROPES) (Lin et al., 2019) is a reading comprehension task. Given a knowledge passage, the model must reason over the causal and qualitative relations in the text and apply them to answer questions about a hypothetical scenario. 70% of background passages contain causal relations, and 26% include both causal and qualitative relations.

What If Question-Answering (WIQA) (Tandon et al., 2019) is a multiple-choice QA task requiring reasoning over procedural descriptions of natural processes. WIQA involves predicting the downstream magnitude (*more*, *less*, or *no effect*) of a perturbation to an individual step in a procedural chain.

A.4 Causal Reasoning Typology

In Figure 7, we present the overall distribution of causal QA questions by causal category from out typology. Cause prediction is the most represented at 30%, followed by effect prediction at 26%, while cause comparison is the least encountered. Figure 8 shows the distribution of CQA categories within each CALM-Bench task. We find a general mixture of all categories across tasks, with an overrepresentation of cause prediction and effect prediction examples. However, WIQA is an outlier, consisting exclusively of effect quantification examples.

A.5 CASK-Schema Relation Mapping

In Table 8 we provide a mapping of CASK-Schema to other public knowledge sources.

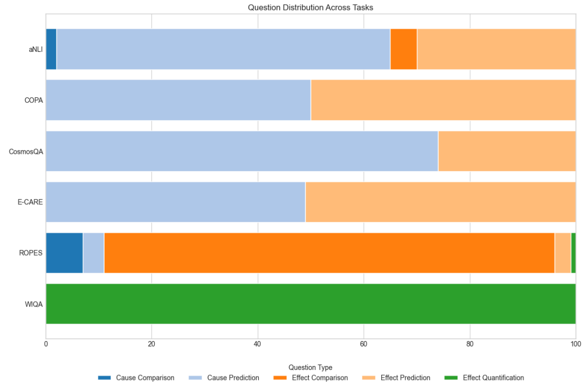

Figure 7: Overall distribution of questions by causal reasoning category.

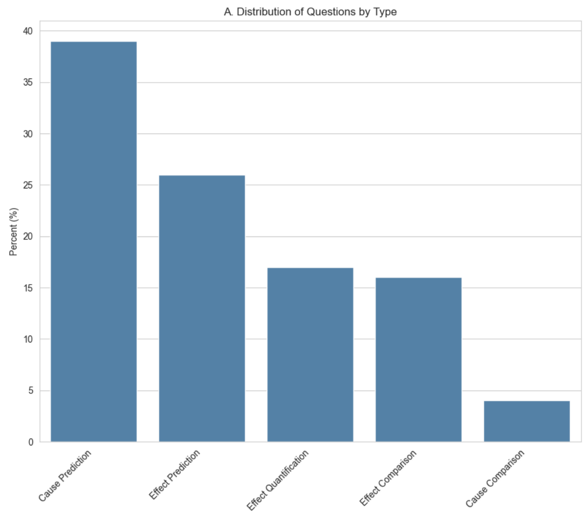

Figure 8: Distribution of causal reasoning type within each CALM-Bench task.

A.6 Causal System Generation Pipeline

A.6.1 Causal System Generation Prompt

In Table 10, we provide the prompt used to the generate causal systems.

A.6.2 Causal System Validation

The validation process consists of two steps: first, building an index of ground truth knowledge, and second, validating knowledge from the generated causal systems. Relevant knowledge triples and factual statements are extracted from public knowledge sources, including ATOMIC, CauseNet, GenericsKB, and Wikidata. A mapping of CASK-Schema to these sources is provided in Table 8. The extracted triples are then linearized using sentence templates provided in Table 9.

For our ground truth semantic knowledge store, we use ChromaDB. The `all-mpnet-base-v2` model from the SentenceTransformers library is used for indexing and retrieval. The vector database is initialized (Table 11) to support cosine similarity matching, and the linearized triples are added and indexed.

During validation, the knowledge store is queried for matching ground truth facts. A triple is considered valid if at least two different matches are found in the semantic store with a cosine similarity of 0.75 or greater.

A.6.3 Causal System Merging

The merging process is formally described in Algorithm 1. First, we generate clusters based on the TF-IDF representations of causal systems. We use the K-means clustering implementation[7] with default parameters, setting the number of clusters to half the number of generated causal systems.

For each cluster, we iterate through the causal systems and compute the pairwise similarity between the comparator system and all other systems within the cluster. Systems with a similarity score of 0.80 or greater are selected as merge candidates. All selected candidates are provided in-context to GPT-3.5 Turbo, which is instructed to unify them into a single causal system. The merged candidates are then removed from the cluster. The merge prompts is made available in Figure 9.

A.6.4 Causal System Linearization

Sample templates for triple linearization are provide in Table 9.

A.7 PublicKB

For a fair comparison with public sources of causal knowledge, we construct *PublicKB* as a baseline. PublicKB consists of 347,706 causal triples extracted from ATOMIC (Sap et al., 2019), CauseNet (Heindorf et al., 2020), and ConceptNet (Speer et al., 2017). CauseNet contributes the majority of triples, with 197,806 triples and 80,223 unique entities. However, the cause-effect relation is the most prevalent in CauseNet.

To ensure a fair comparison with CASK-Db, we include all causality-related relations from ATOMIC and ConceptNet listed in Table 8 (e.g., */r/HasSubevent* from ConceptNet or *Desires* from ATOMIC) that directly map to relations in CASK-Schema. However, cause-effect triples in PublicKB are not aligned with supporting context (e.g., *hasSubevent*, *xReason*, etc.), simulating the limitations

[7] `https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html`

Algorithm 1: Causal System Unification

Causal systems $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$,
sentence embedding model E,
similarity function $\text{sim}(\mathbf{v}_s, \mathbf{v}_{s'}) = \frac{\mathbf{v}_s \cdot \mathbf{v}_{s'}}{\|\mathbf{v}_s\| \|\mathbf{v}_{s'}\|}$
similarity threshold $\theta = 0.80$,
merge function $\text{merge}(S)$ Merged set of causal systems $\mathcal{S}'$
Step 1: Clustering
Apply K-nearest neighbors algorithm to cluster the causal systems in $\mathcal{S}$.
Let $\{C_1, C_2, \ldots, C_m\}$ be the resulting clusters.
Step 2: Encoding
for *cluster* C_i **do**
  for *causal system* $s \in C_i$ **do**
    Encode s using the sentence embedding model E.
    Let $\mathbf{v}_s$ be the embedding vector of system s.
  end
end
Step 3: Merge Candidate Identification
for *each cluster* C_i **do**
  for *causal system* $s \in C_i$ **do**
    Identify the set of causal systems $S' \subset C_i$ where $\text{sim}(\mathbf{v}_s, \mathbf{v}_{s'}) \geq \theta$.
    Designate S' as merge candidates.
  end
end
Step 4: Merging
for *set of merge candidates* S' **do**
  Apply the merge function $s' = \text{merge}(S')$ to create a single causal system s'.
  Remove the systems in s' from C_i and add s' to S_i.
end
Output: Merged set of causal systems $\mathcal{S}'$.

Causal System Merging Prompt
Merge the candidate causal systems into a single, comprehensive description. The description should contain a title, a short description, and a list of relevant knowledge triples.
Guidelines:
1. Identify the underlying causal system from the provided list of candidates.
2. Generate a concise title (2-3 words) for the causal system.
3. Provide a generic description describing the merged causal system. This description should highlight the primary causal relationship.
4. Construct knowledge triples to describe this system. Each triple should be formatted as: `- [Head Predicate]; [Relation]; [Tail Predicate]`.
5. Ensure that head and tail predicates are generalized and do not contain specific names or pronouns.
6. Merge similar knowledge triples from the candidates into a single triple.
7. Ensure that the entities used in the triples are consistent across the entire system.
8. Use only the following relations in your triples: *cause-effect*, *has-contributing-factor*, *has-requirement*, *has-subevent*, *precedes*, *reacts-to*, *has-intent*, *magnifies*, and *mitigates*.
9. Use as many of the specified relations as possible to cover various aspects of the causal interactions.
10. The output should contain the following headers: *Title*, *Description*, and *Triple*. Use a newline after each header.

The relations are defined as follows: [...]
Task Input:
Merge Candidates: [...]

Figure 9: Causal System Merging Prompt

of existing public causal knowledge stores.

PublicKB contains nearly 63 times more triples than CASK-Db (350K vs. 5.4K). However, we hypothesize that causal knowledge structured as *causal systems*, as in CASK-Db, provides better alignment with the causal reasoning needs of language models and should lead to improved downstream QA accuracy.

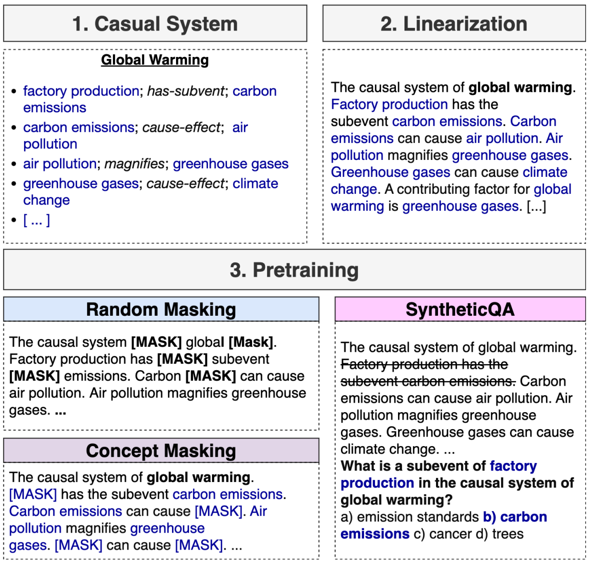

Figure 10: Knowledge-guided pretraining process and strategies.

A.8 Knowledge Injection Details

A.8.1 Pretraining Strategies

Masking-Based Strategies. We consider two masking-based methods (random masking and concept masking) and propose *SyntheticQA*. We adopt the pretraining process from Hosseini et al. (2022), where knowledge triples are first converted to natural language sentences (linearization) before pretraining. Our approach (Figure 10) processes each causal system independently by first linearizing its knowledge triples using predefined sentence templates (Table 12) and aggregating them into natural language causal descriptions. These descriptions are then used to construct pretraining examples.

Next, tokens in the causal descriptions are masked. For random masking, we follow the standard BERT masking ratio of 15% (Devlin et al., 2019) and mask tokens corresponding to whole words (Cui et al., 2021). Concept masking (Sun et al., 2020) extends this approach by masking only entity and relation tokens from the knowledge triples. We randomly select 15% of causal concepts within the system and mask all mentions in the description.

SyntheticQA. *SyntheticQA* generates multiple-choice QA examples based on causal system descriptions. The generation process is formalized in Algorithm 2. Before generating questions, we construct a set of templates for each relation in CASK-Schema (Table 12). Each relation has two template types: one where the answer is the head element and one where the answer is the tail element of the seed triple. For instance, for the *cause-effect* relation, templates include *"What is the cause of tail?"* (answer: head) and *"What is the effect of head?"* (answer: tail). Multiple templates introduce linguistic variation in the generated questions.

The *SyntheticQA* process begins with a seed causal system and a randomly selected seed triple. The system's triples are linearized into a paragraph-level description. The head or tail entity of the seed triple is chosen as the answer candidate, and a question template corresponding to the selected candidate is applied. The generated multiple-choice question consists of four answer options: one correct answer, one adversarial option sampled from within the causal system, and two additional distractors sampled from CASK-Db more broadly. The answer choices are shuffled and labeled (a through d) to prevent positional biases. To encourage reasoning, the sentence corresponding to the seed triple is removed from the causal system description.

After generation, the pretraining dataset contains 6,522 questions, which is split into training and validation sets using a 90-10 split.

A.8.2 SyntheticQA Implementation

A.8.3 Experiment Details

For our experiments, we use the encoder-decoder FLAN-T5 (Chung et al., 2022) base model, which has 220 million parameters. FLAN-T5 has been extensively trained on the FLAN collection (Longpre et al., 2023), comprising 1.8K tasks, and has demonstrated state-of-the-art performance across a wide range of QA tasks.

Each experiment initializes the FLAN-T5 model with its original weights[8], finetunes it for 5 epochs on a specific CALM-Bench task, and evaluates QA accuracy on the corresponding test set. Baseline measurements are obtained by evaluating the model on each benchmark task before knowledge injection.

Knowledge injection experiments require an initial 5-epoch finetuning phase using pretraining examples from CASK-Db, applying one of the described strategies. The model is then checkpointed and subsequently finetuned for 5 additional epochs with early stopping on the benchmark task before evaluating QA accuracy. After testing, the model

[8] https://huggingface.co/google/flan-t5-base

Algorithm 2: Synthetic QA Generation

```
Data: CASK-Db, Templates: dict 𝒯
Result: Multiple-choice CQA example
foreach causal description CD, causal system CS ∈ CASK-Db do
    foreach triple (h, r, t) ∈ CS do
        Randomly select e ∈ {h, t} as the answer candidate;
        Select template T = 𝒯[r][e] based on relation r and selected answer candidate e;
        Generate question Q using template T;
        Randomly select an adversarial concept c_a from other triples in the causal system;
        Randomly select two additional concepts c_1, c_2 from CASK-Db;
        Formulate the question Q with options {e, c_a, c_1, c_2};
        Remove the linearized sentence corresponding to the seed triple (h, r, t) from CD;
    end
end
```

is reverted to the pretraining checkpoint to isolate the impact of knowledge transfer from CASK-Db.

For reproducibility, we set a global seed of 42. We use the Hugging Face (Wolf et al., 2020) FLAN-T5 implementation and base weights[9]. All experiments are conducted on a single AWS g5.8xlarge EC2 instance with an A10G GPU (24GB memory), 32 vCPUs, and 400GB of storage. PyTorch Lightning[10] is used for training management, and the optimizer is AdamW (Loshchilov and Hutter, 2019), initialized with a constant learning rate of 5e-4.

A.9 Knowledge Injection Experiments

A.10 RAG Details

In a RAG (Lewis et al., 2020) system, knowledge is stored externally and retrieved at inference time. Contemporary RAG implementations follow a standard two-stage pipeline: *retrieval* and *generation*. Formally, the RAG system consists of documents D stored in a vector database B and a generative LLM G. Documents are encoded using a dense-passage retrieval model, which also encodes the query at inference time. Given a query q, the retrieval function R is defined as $R(q|D, B) \rightarrow D'$, where $D' \subseteq D$ represents the subset of semantically relevant documents retrieved by R. After retrieval, a prompt $p = (q, D')$ is constructed by including the query and relevant documents as in-context information. The generative model then produces an answer, expressed as $G(p) \rightarrow a$, where a is the generated response leveraging the retrieved knowledge D'.

All RAG experiments are conducted on an AWS g5.8xlarge EC2 instance with a single A20 GPU (24GB memory), 32 vCPUs, and 400GB of storage. For our RAG setup, we use ChromaDB as the vector database and *multi-qa-mpnet-base-dot-v1* as the retrieval model. This retrieval model is a SentenceTransformer (Thakur et al., 2021) finetuned on 215 million question-answer pairs for asymmetric semantic retrieval.

All non-GPT LLMs (Phi-2, Mistral, and Llama 2) are loaded using the QLoRA (Dettmers et al., 2023) quantization configuration. The configuration used in our experiments is provided below. LLM-specific prompts are detailed in Table 14.

[9] See footnote 3.
[10] https://lightning.ai/docs/pytorch/stable/

A.11 Additional Results

How does pretraining specifically impact the various CALM-Bench tasks?

In Table 6, we present the results of knowledge injection experiments for specific CALM-Bench tasks. We find that CASK-Db is generally more effective than PublicKB for improving downstream causal reasoning. Across both knowledge resources, SyntheticQA is the most effective pretraining method for injecting causal knowledge. On average, SyntheticQA with CASK-Db improves accuracy by 11% (7pp), compared to 3% (1pp) with PublicKB.

CASK-Db demonstrates more consistent knowledge transfer across all pretraining strategies, with degradations only observed for the ROPES and WIQA tasks when using masking-based strategies. In contrast, PublicKB exhibits greater variance and inconsistency in knowledge transfer, reducing accuracy on aNLI by an average of -3% and on ROPES by -6% across all strategies.

How does causal knowledge directly impact zero-shot QA for specific CALM-Bench tasks?

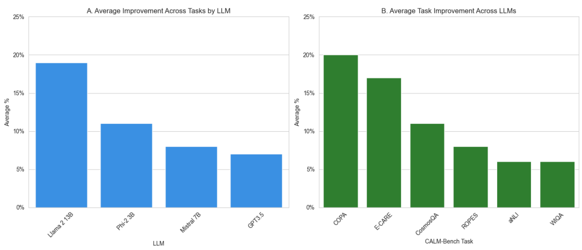

Figure 11: We present the observed improvements that CALM-KB provides the LLMs for zero-shot QA in RAG setting. Subfigure A. provides the average improvement across all task for each evaluated LLM. Subfigure B. shows which tasks benefit most from CALM-KB across all LLMs.

In Table 6, we present the zero-shot QA results for CALM-Bench tasks in the RAG setting, with further analysis in Figure 11. We find that CASK-Db improves QA accuracy across all LLMs. Llama 2 13B benefits the most, achieving an average improvement of 19% across all tasks, with the largest gains on COPA and E-CARE. In contrast, GPT-3.5 benefits the least, likely due to redundancy, as it was used to generate CASK-Db. However, GPT-3.5 still shows a relative improvement of 7%, demonstrating the utility of CASK-Db.

Across all LLMs, COPA and E-CARE show the most improvement with CASK-Db, averaging an 18.5% increase, while aNLI and WIQA benefit the least, with an average improvement of 9%.

	aNLI	COPA	CosmosQA	E-Care	ROPES	WIQA
Baseline	*0.58*	*0.74*	*0.47*	*0.68*	*0.62*	*0.67*
Knowledge: Public KB						
Random Masking	0.56	0.73	0.52	0.63	0.57	0.68
Concept Masking	0.55	0.78	0.52	0.64	0.59	0.7
SyntheticQA	0.57	0.75	0.54	0.69	0.59	0.7
Knowledge: CALM-KB (ours)						
Random Masking	0.59	0.77	0.55	0.71	0.6	0.67
Concept Masking	0.6	0.78	0.57	0.73	0.6	0.68
Synthetic QA	0.61	0.81	0.6	0.73	0.63	0.78

Table 5: Evaluation of CASK-Db in the knowledge injection setting for CALM-Bench tasks. Improvements over the baseline are shaded in green, while regressions are shaded in red.

	aNLI		COPA		CosmosQA		E-Care		ROPES		WIQA	
	Base	*+KB*	Base	*+KB*	Base	*+KB*	Base	*+KB*	Base	*+KB*	Base	*+KB*
GPT3.5	0.74	*.76*	0.95	*.98*	0.74	*.77*	0.80	*.89*	0.44	*.46*	0.58	*.66*
Llama 2 13B	0.51	*.58*	0.55	*.75*	0.31	*.36*	0.52	*.73*	0.57	*.59*	0.51	*.53*
Mistral 7B	0.68	*.75*	0.9	*.95*	0.46	*.51*	0.77	*.85*	0.56	*.60*	0.47	*.49*
Phi-2 3B	0.53	*.52*	0.57	*.76*	0.33	*.37*	0.50	*.53*	0.48	*.55*	0.47	*.48*

Table 6: Evaluation of CASK-Db for zero-shot QA in a RAG setting. Improvements over the baseline are shaded in green, while regressions are shaded in red.

Task	Example	Size	Format	Domain
aNLI (Bhagavatula et al., 2020)	1: Jessie wants to save the planet. 2: This summer has been the hottest in all history. ***Which hypothesis best explains the provided observations?*** A) Jessie decides to buy a new truck. B) Jessie decides to sell her truck and use public transportation instead.	174,226 Train: 169,654 Val: 1,532 Test: 3,040	MC	social, world
COPA (Gordon et al., 2012)	Air pollution in the city worsened. ***What is the most plausible cause?*** A) Factories increased their production. B) Factories shut down.	1,000 Train: 500 Test: 500	MC	world
CosmosQA (Huang et al., 2019)	Two things happened today in Beijing. First off, incoming journalists were amazed to find China had successfully lifted the brown haze in city. Skies were crystal blue and the air felt noticeably lighter. ***Why did the sky appear clearer?*** A) None of the above choices. B) The citizens learned to ignore the gloomy skies. C) The citizens made an effort to cut down on pollution. D) A large storm had recently passed.	35,210 Train: 25,262 Val: 2,985 Test: 6,963	MC	social, world
E-Care (Du et al., 2022)	The city is determined to control air pollution. ***What is the effect?*** A) They have to reduce the number of automobiles. B) Environmental pollution has been increased.	17,051 Train: 14,929 Test: 2,122	MC	social, world, science
ROPES (Lin et al., 2019)	There are two planets, Glarnak and Bornak, that share the same atmospheric composition. The planets have nearly identical ecosystems and topography. The main difference between the two planets is the level of global warming on each planet. Glarnak is experiencing a strong impact from global warming. Bornak, though, is experiencing practically no effects of global warming. ***Which planet has more pollutants in the atmosphere?***	14,322 Train: 10,924 Val: 1,688 Test: 1,710	open	science, world
WIQA (Tandon et al., 2019)	1. A seed is in soil. 2. The seed germinates. 3. The plant grows roots. 4. The plant grows out of the ground. 5. The plant gets bigger. 6. The plant flowers. 7. The flower produces fruit. 8. The fruit releases seeds. 9. The plant dies. ***Suppose less pollution in the environment happens, how will it affect the population of plants?*** A) More B) Less C) No Effect	39,705 Train: 29,808 Val: 6,894 Test: 3,003	MC	science, world

Table 7: CALM-Bench is a multi-task causal QA benchmark consisting of six diverse QA tasks requiring both causal reasoning and knowledge.

Relation	Atomic	CauseNet	ConceptNet	WikiData
cause-effect	Causes	cause-effect	/r/Causes	has cause (P828) has effect (P1542) immediate cause of (P1536) has immediate cause (P1478)
has-contributing-factor	n/a	n/a	n/a	has contributing factor (P1479)
reacts-to	oReact xReact	n/a	n/a	n/a
precedes	isBefore	n/a	/r/HasPrerequisite	follows (P155)
has-subevent	isAfter hasFirstSubEvent hasLastSubEvent	n/a	/r/HasSubevent /r/HasFirstSubevent /r/HasLastSubevent	followed by (P156)
magnifies	n/a	n/a	n/a	n/a
mitigates	n/a	n/a	n/a	n/a
has-intent	Desires xNeed xReason xWant/CausesDesire	n/a	/r/CausesDesire	n/a

Table 8: A mapping of relations in CALM-Schema to public knowledge resources. CALM-Schema provides the most complete representation of causal systems and is compatible with external resources as well.

Relation	Template
cause-effect	$head can lead to $tail. sometimes $head can result in $tail. $head may cause $tail. $tail can sometimes be a consequence of $head.
has-contributing-factor	due to $head, $tail can occur. $head is a contributing factor to $tail. $head plays a role in $tail. $head can contribute to $tail. $tail can be influenced by $head.
has-requirement	$head is a prerequisite for $tail. $tail cannot occur without $head. $head is necessary for []$tail. without $head, $tail is not possible. $head must be present for $tail to happen.

Table 9: Sample sentence templates for triple linearization

Causal System Generation Prompt
Analyze the given scenario to identify the underlying causal system, then generate knowledge triples to describe this system. Each triple should be formatted with a leading dash, e.g. "- [Head Predicate]; [Relation]; [Tail Predicate]". Ensure that the head and tail predicates are general, not containing pronouns or specific referents. Utilize only these relations: cause-effect, has-contributing-factor, has-requirement, has-subevent, precedes, reacts-to, has-intent, magnifies, and mitigates. Focus the triples on general actions, events, or conditions, along with their expected outcomes or influences within a causal system. Avoid specific names and personal pronouns. Create a concise title (2-3 words) and a generic description that captures the essence of the general causal system, emphasizing clarity and brevity. The relations are defined as follows: [...] Task: 1. Concisely describe the identified causal system. 2. Generate a brief title for the causal system. 3. Produce knowledge triples based on the scenario. Maintain consistency in the head and tail entities across triples, and incorporate as many of the 8 relevant relations as possible. Example Scenario: [...] Input: [...]

Table 10: Causal System Generation Prompt

Knowledge Store VectorDB

```
import chromadb
from chromadb.utils import embedding_functions


# Specify retriever model
embedder = embedding_functions.SentenceTransformerEmbeddingFunction(
model_name="all-mpnet-base-v2"
)

client = chromadb.PersistentClient(path="knowledge-cache/")
db = client.create_collection(
name="causal-kb",
embedding_function=embedder,
metadata={
"hnsw:space": "cosine",
}
```

Table 11: ChromaDB config for knowlege store

Relation	Head Template	Tail Template
cause-effect	What is the cause of $tail$? If $tail happens, what was the cause$?	What is the effect of $head$? What happens as a result of $head$?
has-contributing-factor	What contributes to $tail$? Which factor plays a role in $tail$?	What is the contributing factor of $head$? What is $head a contributing factor of$?
has-requirement	What is required for $tail$? What must happen for $tail to occur$?	What is required for $tail$? What must happen for $tail to occur$?

Table 12: Sample QA templates used for *SynetheticQA*

QLoRA Configuration

nf4_config = BitsAndBytesConfig(
load_in_4bit=True,
bnb_4bit_quant_type="nf4",
#bnb_4bit_use_double_quant=True,
bnb_4bit_compute_dtype=torch.bfloat16
)

Table 13: QLoRA configuration for loading LLMs into memory.

Model	Template
Phi-2	Instruct: Answer the question provided the scenario below. Do not provide an intro or concluding remarks in your response. Do not provide an explanation. Just provide an answer. For multiple-choice return the letter and answer only. Input: [[input]] Output:
Mistral	[INST] Answer the question provided the scenario below. Do not provide an intro or concluding remarks in your response. Do not provide an explanation. Just provide an answer. For multiple-choice return the letter and answer only. Input: [[input]] [/INST] Output:
Llama 2	[INST] Do not provide an intro or concluding remarks in your response. Be as concise as you can be when responding. Answer the question provided the scenario below. Do not provide an intro or concluding remarks in your response. Do not provide an explanation. Just provide an answer. For multiple-choice return the correct answer. Example: What is the capital of France? Options: a) Paris b) London c) Berlin d) Rome Output: a) Paris Input: [[input]] [/INST] Output:
GPT 3.5	Answer the question below. Do not provide an explanation. Provide both the letter and answer option. Use the prefix "output:" and then provide the answer. Input [[input]]

Table 14: Prompt Templates used for RAG experiments