UNDERSTANDING THE REASONING ABILITY OF LANGUAGE MODELS FROM THE PERSPECTIVE OF REASONING PATHS AGGREGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Pre-trained language models (LMs) are able to perform complex reasoning without explicit fine-tuning. To understand how pre-training with a next-token prediction objective contributes to the emergence of such reasoning capability, we propose that we can view an LM as deriving new conclusions by aggregating indirect reasoning paths seen at pre-training time. We found this perspective effective in two important cases of reasoning: logic reasoning with knowledge graphs (KGs) and math reasoning with math word problems (MWPs). More specifically, we formalize the reasoning paths as random walk paths on the knowledge/reasoning graphs. Analyses of learned LM distributions suggest that a weighted sum of relevant random walk path probabilities is a reasonable way to explain how LMs reason. Experiments and analysis on multiple KG and MWP datasets reveal the effect of training on random walk paths and suggest that augmenting unlabeled random walk reasoning paths can improve real-world multi-step reasoning performance.

1 INTRODUCTION

Recently, pre-trained large language models (LLMs) (Touvron et al., 2023a;b; Brown et al., 2020) have demonstrated remarkable capabilities in performing intricate reasoning tasks (Kojima et al., 2022). These tasks include problem-solving with world knowledge (Hendrycks et al., 2020; Suzgun et al., 2022), logical reasoning (Pan et al., 2023), and solving mathematical problems (Cobbe et al., 2021; Hendrycks et al., 2021). These models are typically not explicitly fine-tuned to solve these tasks. Recent research (Jain et al., 2023) also suggests that the supervised fine-tuning process following pre-training only learns a wrapper on top of the already existing model capabilities, instead of learning new ones. It is intriguing to understand how next-token prediction pre-training contributes to the emergence of such reasoning capability. A better understanding of this matter can also inspire new pre-training/fine-tuning techniques to improve these important abilities of LLMs.

It is well-known that LLMs acquire emergent abilities through extensive pre-training (Wei et al., 2022a). In this paper, we focus on elucidating the emergence of reasoning ability — the capacity to draw novel conclusions from existing knowledge, which has been less studied. Many recent works also attempt to understand this phenomenon. Some works focus on understanding Transformers' reasoning capability by construction (Liu et al., 2023; Chi et al., 2023; Feng et al., 2023). Others try to provide post hoc mechanistic explanations (Geiger et al., 2021; Wu et al., 2023; Hanna et al., 2023) or understanding inference time in-context learning reasoning (Li et al., 2023; Razeghi et al., 2022; Wang et al., 2023). Our study is more relevant to the line of work analyzing the contribution of pre-training data to LM reasoning (Bi et al., 2023; Chen et al., 2023; Xiao & Liu, 2023; Zhou et al., 2023; Ramesh et al., 2023). [1]

In contrast to these works, we adopt a Bayesian view and try to understand why next-token-prediction pre-training can unlock LMs' reasoning ability. More specifically, we hypothesize that LMs can aggregate the indirect reasoning paths seen at pre-training time, through the next-token-prediction training objective. In a real-world scenario, the reasoning path can be a piece of text argument connecting two concepts. We hypothesize that, at inference time, this enables an LM to jump

[1] A detailed discussion of related work can be found in the Appendix.

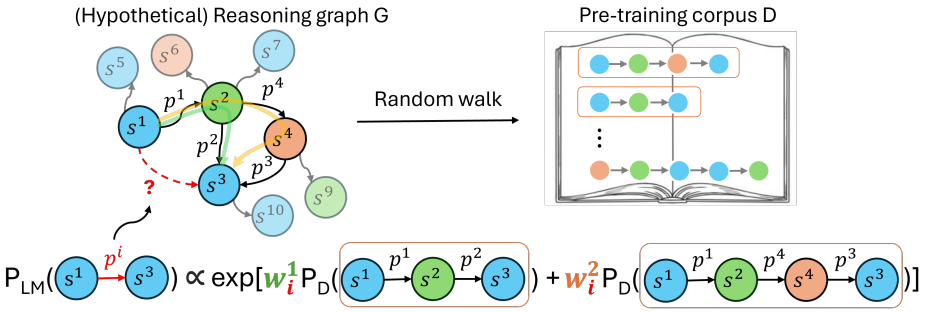


Figure 1: We hypothesize that the pre-training corpus can be viewed as generated from random walks on a reasoning graph over world knowledge/concepts. With each node s_i representing concepts, p_j can be viewed as arguments that connect them. Then we hypothesize that a language model (LM) training on such a corpus can be viewed as reasoning by a weighted aggregation of random walk paths that connect the entities in interest. P_{LM} denote the LM distribution while P_D denotes the random walk probability from the pre-training corpus. w_i^1 denotes the weight assigned to the first random walk path by the LM for argument p_i, and w_i^2 denotes the weight assigned to the second random walk path.

from one concept to another during its reasoning process, which could be verbalized by generating chain-of-thought (CoT) solutions (Wei et al., 2022b), or silent without generating outputs.

Prystawski et al. (2023) propose a different hypothesis that localized structure on variable dependencies in training data is important for LM reasoning, especially CoT reasoning. Our hypothesis implies a similar property of the pre-training data: when two concepts are related by a reasoning path, they are highly likely to cooccur in the data and thus form a graph-like localized structure. One drawback of Prystawski et al. (2023)'s work is that their experiments equate reasoning to conditional probability estimation of boolean variables with intermediate variables, which can be considered overly simplified compared to real-world reasoning processes. In our paper, we aim to produce a more realistic analysis of the effect of training data by closely examining two predominant types of reasoning: logical reasoning and mathematical reasoning. In these two reasoning scenarios, we first construct unsupervised random walk paths, which are used to (continually) pre-train the LM with next-token loss. Then we adopt the pre-trained LM to perform reasoning tasks on unseen examples.

For logical reasoning, we analyze a straightforward yet general reasoning scenario: reasoning over knowledge graphs. A knowledge graph (KG) stores facts in the form of triples (e_1, r, e_2), where e_1 and e_2 represent entities connected by the relationship r. KGs can be incomplete, lacking certain relations between existing entities. These missing relations can typically be inferred from the known triples stored in the KG by employing logical rules. For instance, the relation (A, `isGrandChildof`, C) can be derived from the triples (A, `isSonOf`, B) and (B, `isSonOf`, C). We formalize a reasoning path as a **random walk path** on the KG, which enables us to accurately compute its probability. We show that an LM pre-trained from scratch on random walk paths generated from a given KG can accurately deduce missing relation connections. We also analyze the KL divergence between LM output distributions and weighted/unweighted sums of random walk path probabilities, which are variances of the classic path ranking algorithm (PRA) (Lao et al., 2011). Our analysis suggests that the LM distribution shares many similarities with aggregating the probabilities of possible random walk paths in a logical-rule-aware manner, and is usually superior to them.

For mathematical reasoning, we focus on a more complex case of reasoning: solving math word problems (MWPs). Since it is very challenging to pre-train an LM from scratch to perform well on MWPs, which require both math deduction and language understanding, we propose to continue training on a pre-trained base LM. Based on the insights obtained from the KG reasoning analysis, We propose to create **random walk reasoning paths** from existing CoT training data, and test the effectiveness of next-token-prediction training on these unlabeled reasoning paths. More specifically, we construct a reasoning graph by regarding the reasoning state at each CoT step as the graph node. Then we reorder and reconnect the existing CoT steps to form the random walk paths on the graph. Experiment results on three MWP datasets, GSM8K (Cobbe et al., 2021), AQUA (Ling et al., 2017), SVAMP (Patel et al., 2021), show consistent improvement compared to vanilla supervised fine-tuning, and a similar effect of random walk path length as in the KG reasoning case is observed.

Our findings can be summarized as follows: (a) We show in both reasoning scenarios that our weighted random walk reasoning paths aggregation hypothesis is one (of many) valid ways to explain how LMs may gain their reasoning ability; (b) We show that LMs can utilize unlabeled reasoning paths highly efficiently and show the potential of incorporating the random walk idea to real-world (continue) pre-training.

2 LOGICAL REASONING

We first analyze a well-controlled case of logic reasoning, knowledge graph (KG) reasoning, by pre-training a small Transformer over random walk paths from KGs. The KL divergence between aggregated random walk path probabilities and LM distribution shows that LM is very close to a weighted aggregation. We also show that KL divergence reflects how LMs assign weights to logical rules. We find that there is usually an optimal random walk path length for training LMs. These observations support our reasoning paths aggregation hypothesis.

2.1 PROBLEM SETTING

Consider a knowledge graph $\mathcal{G} = \{(e_1^i, r^i, e_2^i)\}_{i=1}^N$ consisting of N triples, such that the head entity e_1^i and tail entity e_2^i are related by r^i for all i. Let $\mathcal{R}$ denote the set of all possible relations and $\mathcal{E}$ denote the set of all entities. Our goal is to predict a set of unseen triples $\mathcal{T} = \{(e_1^j, r^j, e_2^j)\}_{j=1}^m$, $e_1^j, e_2^j \in \mathcal{E}$, $r^j \in \mathcal{R}$, by training a Transformer based generative language model (LM) from scratch on the given knowledge graph $\mathcal{G}$. To translate a triple into a sentence (i.e. a sequence of tokens), We add each entity e^i and relation r^i as a new token (<e_i> and <r_i>) to the Transformer's vocabulary and translate each triple into a three-token sentence "<e_i> <r_j> <e_k>.".

2.2 LANGUAGE MODEL PRE-TRAINING

We construct the training data by performing random walks on the given KG $\mathcal{G}$. More specifically, we randomly sample a start entity $e \sim U(\mathcal{E})$, where $U(\cdot)$ denotes the uniform distribution. Then we perform a random walk on $\mathcal{G}$ from e by sampling the next node with $e' \sim U(C(e))$, and stop at a maximum path length L_{max}. Then we translate each triple into a sentence and concatenate all the sentences in the sampled random walk path to become a paragraph. The paragraphs are then concatenated together and separated by the special end-of-sequence token to form text chunks of the same length. The training loss function is the next-token prediction loss:

$$\mathcal{L}_{LM}(\theta) = \sum_{\mathcal{D}} \sum_{t=1}^{T} \log \frac{\exp\left(f_\theta(w_{t+1}|w_{1:t})\right)}{\sum_{w\in\mathcal{V}} \exp\left(f_\theta(w|w_{1:t})\right)} \tag{1}$$

Here, θ denotes the LM parameters [2]. $w_i \in \mathcal{V}$ represents a token in the LM vocabulary $\mathcal{V}$, and $w_{1:T}$ is a token sequence in the training data $\mathcal{D}$, where T is the length of a text chunk. To test the reasoning ability of a pre-trained LM, we format the testing triples as sentence completion tasks. For example, the triple (e_1, r, e_2) will be translated to the prompt "<e_1> <r>) ", and let the LM predict the next token, then verify the prediction with the ground truth e_2. Note that, here the raw LM output distribution is over all entities and relations. To make the LM distribution more well-defined and simplify the following analysis, we take the LM output logits over all entities and define the LM output distribution as:

$$P_{\text{LM}}(e_2|e_1, r) = \frac{\exp\left(f_\theta(e_2|e_1, r)\right)}{\sum_{e\in\mathcal{E}} \exp\left(f_\theta(e|e_1, r)\right)} \tag{2}$$

2.3 RANDOM WALK PATHS AGGREGATION

Recall that our hypothesis is LM can aggregate the reasoning paths seen at the pre-training time. In the KG setting, we can explicitly define how the reasoning/random walk paths are aggregated. Inspired by the classic path ranking algorithm PRA (Lao et al., 2011), **we define the aggregation of**

[2]We use a randomly initialized GPT-2 model (Radford et al., 2019).

random walk paths P_w as the exponential of a weighted sum of the probabilities of all appropriate random walk paths connecting the two target entities. More specifically, we are interested in a distribution $P_w(e_2|e_1, r)$ for unseen (e_1, r, e_2) in the form of:

$$P_w(e_2|e_1, r) = \frac{\exp(S_w(e_2|e_1, r)/T)}{\sum_{e\in\mathcal{E}} \exp(S_w(e|e_1, r)/T)} \tag{3}$$

Here $S_w(e_2|e_1, r)$ is a score/logits of e_2. $T > 0$ is a temperature to rescale the weighted logits S_w so that it can match the scale of LM logits f_θ (In practice, we take $T = 0.01$.), and that $P_w(e_2|e_1, r)$ and $P_{\text{LM}}(e_2|e_1, r)$ are more comparable. The score $S_w(e_2|e_1, r)$ is defined to be a weighted sum of the probability of following all possible logical rules going from e_1 to e_2:

$$S_w(e_2|e_1, r) = \sum_{h\in\mathcal{H}} w_r(h)P(e_2|e_1, h)$$

Here $\mathcal{H}$ denotes the set of all possible logical rules, and $h \in \mathcal{H}$ is a specific logical rule. $w_r(h)$ is the weight assigned to rule h when inferring relation r. For example, a rule for inferring the `locatedIn` relation can be $h : (e_1, \texttt{neighborOf}, e_3) \wedge (e_3, \texttt{locatedIn}, e_2)$. Formally, for a target relation r, we consider logic rules with conjunctive form. $\forall\{e_i\}_{i=0}^n \subset \mathcal{E}$,

$$(e_0, r, e_n) \leftarrow (e_0, r_1, e_1) \wedge (e_1, r_2, e_2) \wedge ... \wedge (e_{n-1}, r_n, e_n)$$

where $(e_{i-1}, r_i, e_i) \in \mathcal{G}$. We abbreviate such rule by $h = [r_1, r_2, ..., r_n]$. We can formalize the set of all possible logic rules by $\mathcal{H} = \{[r_1, r_2, ..., r_n]|n \geq 1, r_i \in \mathcal{R}\}$. Then the probability of following a specific logic rule $h \in \mathcal{H}$ between e_1 and e_2 during the random walk would be the sum of the probability of all possible random walk paths from e_1 to e_2 following the rule $h = [r_1, r_2, ..., r_n]$:

$$P(e_n|e_0, h) = \sum_{(e_0,r_1,e_1)...(e_{n-1},r_n,e_n)\in\mathcal{P}_h} \prod_{i=1}^{n} P(e_i|e_{i-1}, r_i)$$

where $\mathcal{P}_h$ denotes all paths from the KG following h. Following the pre-training data generation, we perform a uniform random walk. i.e. $P(e_i|e_{i-1}, r_i) = 1/|C(e_{i-1})|$. Then the rule probability $P(e_2|e_1, h)$ can be computed directly from the KG. To learn the rule weights w_r, we first observe

$$P_w(e_2|e_1, r) = \frac{P_w(r|e_1, e_2)}{\sum_{e\in\mathcal{E}} P_w(r|e_1, e)},$$

if we sample e_1 and e_2 independently and uniformly. Recall Equation (3), we can model $P_w(r|e_1, e_2) \propto \exp S_w(e_2|e_1, r)$. We can even further simplify it into a binary classification problem $p_i = P_w(\mathbb{1}_{r^i=r}|e_1^i, e_2^i)$. Then we can use w_r to parameterize a logistic regression model minimizing:

$$\mathcal{L}_r(w) = -\sum_i [y_i \ln p_i + (1 - y_i) \ln (1 - p_i)] + \lambda|w|,$$

where $p_i = \frac{\exp S_w(e_2^i|e_1^i,r)}{1+\exp S_w(e_2^i|e_1^i,r)}$, and the binary label $y_i = \mathbb{1}_{r^i=r}$. $\lambda|w|$ is a regularization term, and we can take any appropriate norm on w. At training time, we sample positive triples with relation r and negative triples with other relations from $\mathcal{G}$ as training data. We search over the graph to compute their probability of being reached by each rule $P(e_2|e_1, h)$ to compute p_i.

For computation efficiency, we only want to search for a subset of more possible logical reasoning rules $\mathcal{H}_r$ in the test set for each relation r, and assign $w_r(h) = 0$ for $h \notin \mathcal{H}_r$. Note that a rule can be infinitely long, so we set a maximum rule length $n \leq N_{max}$. To obtain $\mathcal{H}_r$, we search over $\mathcal{G}$, and record all paths between any two entities that are connected with the relation r, and shorter than N_{max}. We then collect the rules that have more than m valid paths.

A simplified version of P_w would be letting $w_r(h) = 1$ for all h and r. And **we define this unweighted aggregation distribution to be** P_s:

$$P_s(e_2|e_1, r) = \frac{\exp(\sum_{h\in\mathcal{H}_r} P(e_2|e_1, h)/T)}{\sum_{e\in\mathcal{E}} \exp(\sum_{h\in\mathcal{H}_r} P(e|e_1, h)/T)} \tag{4}$$

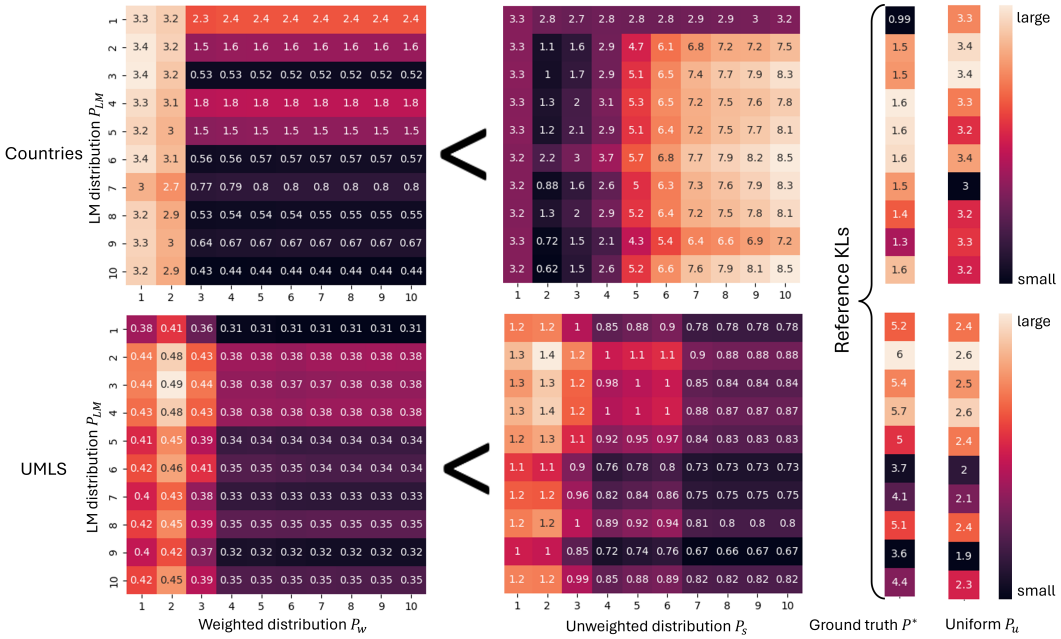


Figure 2: KL divergence between various reference distributions and LM distribution, with different maximum random walk lengths, averaged over Countries (top) and UMLS (bottom) testing set, respectively. The rows correspond to the LM distribution $P_{\text{LM}}(e_2|e_1, r)$ with maximum pre-training random walk path lengths (L_{max}) ranging from 1 to 10. From left to right, the columns correspond to the weighted aggregation distribution $P_w(e_2|e_1, r)$ with maximum random walk path lengths (N_{max}) from 1 to 10, the unweighted aggregation distribution $P_s(e_2|e_1, r)$ with maximum random walk path lengths (N_{max}) from 1 to 10, the reference distribution $P^*(e_2|e_1, r)$, and the uniform distribution $P_u(e_2)$, respectively. A **darker color** represents a **smaller KL value**, meaning that the two distributions are closer. In general, $KL[P_w, P_{\text{LM}}]$ is always smaller than $KL[P_s, P_{\text{LM}}]$, which implies that LM is learning the difference in rule importance. $KL[P^*, P_{\text{LM}}]$ and $KL[P_u, P_{\text{LM}}]$ serve as anchor points to show the scale of KL values. $KL[P^*, P_{\text{LM}}]$ is generally high because the probability mass concentrates on correct answers, thus it can be very different from the LM distribution. Thus $KL[P^*, P_{\text{LM}}]$ shows how peaky the LM distribution is, and $KL[P_u, P_{\text{LM}}]$ shows how flat the LM distribution is.

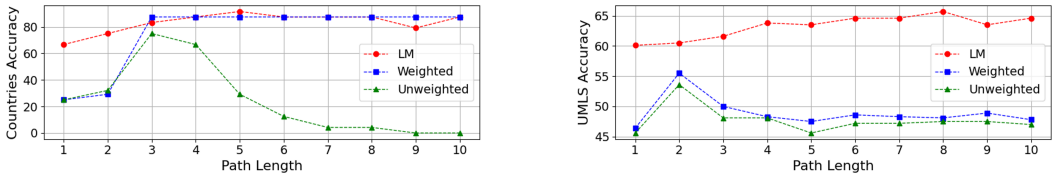
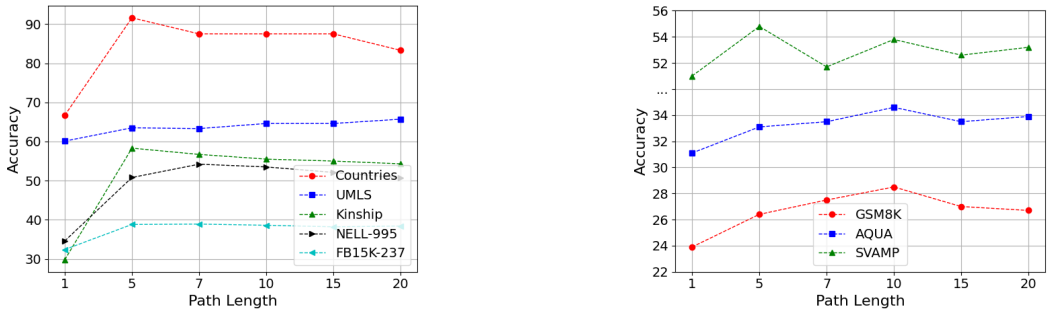


Figure 3: Testing accuracy w.r.t. various maximum pre-training random walk lengths ($1 \leq L_{max} \leq 10$) on Countries (left) and UMLS (right) datasets, respectively. For Countries, the LM (P_{LM}) performance converges to the weighted aggregation (P_w) performance, while for UMLS, LM consistently outperforms both weighted (P_w) and unweighted (P_s) aggregation performance. This shows LM can learn a better logical rule weighting scheme than weighted aggregation in more complex KGs.

2.4 KL Divergence and Prediction Accuracy

To better understand the similarity between LM and the random walk aggregation algorithm as described in the previous section, we propose to compute and analyze the KL divergence between them: $KL[P_w(\mathbf{e}|e_1, r), P_{\text{LM}}(\mathbf{e}|e_1, r)]$, where $\mathbf{e}$ is a random variable taking values in $\mathcal{E}$. To better understand the meaning of the computed KL divergence, we derive an upper bound of it by writing $P_{\text{LM}}(e_2|e_1, r)$ as marginalization over rules $P_{\text{LM}}(e_2|e_1, r) = \sum_{h \in \mathcal{H}} P(e_2|e_1, h) P_{\text{LM}}(h|e_1, r)$. Similarly, we can write $P_w(e_2|e_1, r) = \sum_{h \in \mathcal{H}} P(e_2|e_1, h) P_w(h|e_1, r)$. Then by the Log sum inequality, we can see that the KL divergence of the rule importance is an upper bound of the computed KL:



(a) Logical reasoning with KG

(b) Math reasoning with MWP

Figure 4: Testing accuracy of LM trained on different random walk path lengths L_{max}. Each line corresponds to a different dataset and thus is not directly comparable. We want to highlight the common trend here that each line peaks at some optimal path length.

Proposition 2.1. *If LM effectively learned the random walk data distribution through pre-training,*

$$KL[P_w(\mathbf{e}|e_1, r), P_{LM}(\mathbf{e}|e_1, r)] \leq KL[P_w(\mathbf{h}|e_1, r), P_{LM}(\mathbf{h}|e_1, r)]$$

Here $\mathbf{h}$ is a random variable taking values in $\mathcal{H}$. This means the KL divergence reflects how LM assigns probabilities to possible logical rules based on the given prompt, which implies how the LM learns to do logical reasoning.

KL computation We compute the KL divergence between the weighted aggregation distribution $P_w(e_2|e_1, r)$ as defined in Equation (3) and the LM distribution $P_{\text{LM}}(e_2|e_1, r)$ as defined in Equation (2), abbreviated as $KL[P_w, P_{\text{LM}}]$. We then compare it with the KL divergence between the unweighted aggregation distribution $P_s(e_2|e_1, r)$ as defined in Equation (4) and the LM distribution, abbreviated as $KL[P_s, P_{\text{LM}}]$. To better understand the effect of random walk length, we consider maximum random walk path length ranging from 1 to 10, for both the aggregation distribution ($1 \leq N_{max} \leq 10$) and the LM distribution ($1 \leq L_{max} \leq 10$). We then compute a pairwise KL between each of them and show the results as a heatmap. To better anchor the computed KL divergence, we also compute the KL divergence $KL[P^*, P_{\text{LM}}]$ between a reference distribution P^* and the LM distribution P_{LM}, and KL divergence $KL[P_u, P_{\text{LM}}]$ between the uniform distribution P_u and the LM distribution P_{LM}. Here P^* **is uniform over all correct answers**, and P_u **is uniform over all possible answers**. The described KL divergences for Countries (top) and UMLS (bottom) testing sets are shown in Figure 2. More interpretations of these quantities can be found in the caption.

Accuracy We also compute the prediction accuracy using each method and plot it w.r.t to path length ($1 \leq L_{max} \leq 10$). Note that there could be more than one correct answer for a query (e_1, r). We say the prediction is correct as long as it is one of the correct answers. The described testing accuracy for Countries (left) and UMLS (right) is shown in Figure 3, where **LM** is $\arg\max P_{\text{LM}}$, **Weighted** is $\arg\max P_w$, and **Unweighted** is $\arg\max P_s$. In general, LM predictor P_{LM} performs on par/better than weighted aggregation P_w, and significantly better than the unweighted aggregation P_s. This shows that LM likely learns a better logical rule weighting scheme than P_w.

2.5 Results and Analysis

We consider five KG datasets in total: Countries (Bouchard et al., 2015), UMLS (Kok & Domingos, 2007), Kinship (Denham, 2020), NELL-995 (Xiong et al., 2017), and FB15K-237 (Toutanova et al., 2015). We take the smallest two for KL divergence analysis for their lower time complexity. We show LM prediction accuracy for all datasets with different pre-training path lengths.

KL divergence with Countries In Figure 2 (top), we can see that when the maximum path length for computing the aggregated distribution (columns) is three, there is a sudden drop in $KL[P_w, P_{\text{LM}}]$. This is because the ground truth path length to reach the correct answers in the testing set is three (fixed when constructing the dataset). Both the weighted and unweighted aggregation of random walk paths have low accuracy with path lengths less than three as shown in Figure 3 (left). The behavior of the path aggregation method is not well-defined at this stage and thus can result in an abnormal KL trend. On the other hand, LM yields a non-trivial accuracy when trained with a path length smaller

than three, which shows LM's ability to generalize beyond the pre-training reasoning length. This echoes the findings in Xiao & Liu (2023); Zhou et al. (2023), that Transformers can generalize to longer sequences than training sequences.

As shown in Figure 2 (top left), the weighted aggregation scheme P_w converges to a stable distribution, likely by putting most weights on shorter rules when using N_{max} is large. The LM distribution P_{LM} becomes closer to P_w when L_{max} becomes larger. On the other hand, $KL[P_s, P_{\text{LM}}]$ stably increases when the path length for P_s becomes larger. This echoes the accuracy trends as shown in Figure 3 (left). For the countries dataset, since it only has two relations, longer random walk paths introduce more noise than useful information. Thus by increasing the path length the unweighted aggregation scheme P_s becomes less and less effective. Both P_w and P_{LM} learn to assign a small weight to the long/noisy paths, and thus do not experience an accuracy drop.

KL divergence with UMLS In Figure 2 (bottom), we can see that when the maximum path length for computing the aggregated distribution (columns) is larger than 3, the weighted aggregation scheme P_w also converges to a stable distribution. To investigate why path length 3 is unique, we find the average path length corresponding to the largest number of valid paths for each relation in the testing set is 3.14. We find the average path length corresponding to the largest weight assigned by P_w when $N_{max} = 10$ is 2.75. This confirms that path length three is likely a good rule length for many relations. However, from Figure 3 (right), we can see that both weighted (P_w) and unweighted (P_s) aggregation peaked at path length two instead of three. We believe this is because when the rule length becomes larger (i.e. larger than two), the validity of a rule would be more head entity (e_1) dependent. Using only relation-dependent weight $w_r(h)$ as in P_w is likely insufficient. This also explains why LM constantly outperforms both path aggregation methods: LM likely learns a rule importance function that depends both on the head entity and the relation.

Different from the Countries dataset, UMLS' $KL[P_s, P_{\text{LM}}]$ does not increase when the path length for P_s increases. Instead, $KL[P_s, P_{\text{LM}}]$ follows a similar trend as $KL[P_w, P_{\text{LM}}]$, while in general $KL[P_w, P_{\text{LM}}]$ is smaller than $KL[P_s, P_{\text{LM}}]$. Similarly, in Figure 3 (right), the weighted (P_w) and unweighted (P_s) aggregation has a similar performance, while P_w is slightly better. This shows that the logical rule weights learned by P_w are similar between different rules, so it has similar effects (KL and accuracy) as the unweighted version P_s. The LM also has a flatter distribution, as we can see for UMLS $KL[P^*, P_{\text{LM}}] < KL[P_u, P_{\text{LM}}]$ while for Countries $KL[P^*, P_{\text{LM}}] > KL[P_u, P_{\text{LM}}]$. This is likely because UMLS is more complex than Countries (49 v.s. 2 relations), thus many longer paths and rules are similarly useful for prediction, making the LM distribution flatter.

Prediction accuracy v.s. pre-training path length We briefly touched on how the pre-training random walk path length L_{max} affects the LM distribution in the analysis above. In general, a longer path length improves the prediction accuracy and decreases $KL[P_w, P_{\text{LM}}]$. This shows that LM can improve the logical rule weight assignment when trained with a longer path length. To further investigate this problem, we pre-train LM on longer random walk path lengths with more KG datasets.

In Figure 4a, we show the LM prediction accuracy v.s.$L_{max} \in \{1, 5, 7, 10, 15, 20\}$, trained on five different KG datasets. In general, there is a large performance gain from a path $L_{max} = 1$ to $L_{max} = 55$. Note that when $L_{max} = 1$, we randomly sample individual triples from a KG. i.e. There are no reasoning paths in the training data. So it is important to have reasoning paths with a non-trivial length in the pre-training data, to enable the LM's reasoning ability. By extending the maximum length from 10 to 20, we can see that there is a slight drop in the Countries dataset. Similarly, in most datasets, there is a small decrease after an optimal path length. This is likely because a too-long random walk path would contain more noise/unrelated triples for reasoning. i.e. It is less likely to be useful for predicting the head and tail entity relation in a path aggregation sense. On the other hand, we can understand this from a localized data structure perspective (Prystawski et al., 2023): a sufficiently long random walk path makes any two entities similarly possible to appear in the same path, thus hurting the local dependency in the training data.

3 MATH REASONING

After carefully analyzing the logical reasoning on KGs, we want to apply and verify the obtained insights on a more interesting and realistic case of reasoning: math reasoning with textual descriptions, which is usually called math word problems (MWPs). We continue training a pre-trained LM with

Algorithm 1 Random Walk on Latent Graph

Input: CoT dataset $\mathcal{D}$, latent graph $\mathcal{G}$, maximum path length L_{max}.
Randomly initialize current node $a = A_k$. Initialize path $p = []$
repeat
  Randomly choose a CoT step $r_j^i \in a$.
  Uniformly sample m from $[1, L]$.
  Append $r_j^i, r_{j+1}^i, ..., r_{\min\{j+m,n^i\}}^i$ to path p.
  Suppose $r_{\min\{j+m,n^i\}}^i \in A_l$. Set $a = A_l$.
until len$(p) \geq L_{max}$.

random walk reasoning paths and show that these unlabeled paths consistently benefit math reasoning performance on three datasets. We also observe a similar optimal random walk path length effect as in the logical reasoning case, which is associated with the intrinsic reasoning length of different datasets. These results support our reasoning path aggregation hypothesis and imply principles for constructing/augmenting pre-training data.

3.1 Problem Setting

Suppose we have training data $\mathcal{D} = \{(x^i, r_1^i, r_2^i, ..., r_{n^i}^i, y^i)\}_i$, where x^i is a question described in the text that needs to be solved by math technique. $r_1^i, r_2^i, ..., r_{n^i}^i$ is a chain-of-thought (CoT) solution, where r_j^i is one reasoning step. y^i is the ground truth answer to the question. Since MWP datasets are hard to collect and usually small in size, a model is not likely to generalize to new questions by aggregating reasoning paths over this small set of CoT reasoning paths. Fine-tuning on a pre-trained LM can effectively mitigate this problem since the LM has already seen many other reasoning paths at the pre-training time, but more unlabeled reasoning paths specific to this task would likely improve the testing performance if the path aggregation hypothesis still holds for this task.

3.2 Random Walk on Latent Reasoning Graph

We assume that CoT paths $r_1^i, r_2^i, ..., r_{n^i}^i$ can be regarded as random walk paths sampled from a reasoning graph $\mathcal{G}$, where the nodes are the reasoning states at each step r_j^i. The reasoning state can be regarded as a belief that will be updated after each reasoning step. Denote the last hidden state pre-trained LM we are going to tune by f_θ. To represent the reasoning state for each step r_j^i, we propose to use f_θ to cumulatively encode all the steps before r_j^i, and then average over the sequence dimension, to obtain a fixed dimensional vector $s_j^i = \text{avg } f_\theta(x^i, r_1^i, r_2^i, ..., r_j^i)$.

Assuming similar s_j^i's are sampled from the same node of the latent reasoning graph, we propose to cluster [3] similar s_j^i's together to form a node. Suppose we have constructed a graph $\mathcal{G}$ from the CoT dataset $\mathcal{D}$, with nodes $A_1, A_2, ..., A_K$, where K is predefined by the clustering algorithm. Each CoT step would be classified into a node. i.e. $r_j^i \in A_m$ for some $m \in [1, k]$. Then we can perform random walks on the graph by using the original CoT as links between the nodes as shown in Algorithm 1. Then we record the random walk paths produced by Algorithm 1 and do next-token-prediction training on them for M steps. To make sure the LM can produce a final answer, we do another $N - M$ step of supervised fine-tuning (SFT) on the original dataset $\mathcal{D}$, for some $N > M$.

3.3 Experiments

Datasets. We conduct experiments on three math word problem (MWP) datasets: GSM8K (Cobbe et al., 2021), AQUA (Ling et al., 2017), SVAMP (Patel et al., 2021). [4]

Training We do LoRA (Hu et al., 2021) parameter efficient training in 8 bits with Llama 2 models (Touvron et al., 2023b). If not specified, we default to using the 7B version.

Results. In Table 1, we demonstrate the effectiveness of our proposed method against the supervised fine-tuning (SFT) baseline. We train both our method and SFT with $N = 2500$ steps in total. The

[3] In practice we use K-meanings clustering.
[4] Dataset details can be found in Appendix.

Model	Method	GSM8K	AQUA	SVAMP	Avg.
7B	SFT	26.8	30.0	53.3	36.7
	Ours	**28.5**	**34.6**	**55.8**	**39.6**
13B	SFT	37.1	35.0	66.4	46.2
	Ours	**41.2**	**37.4**	**69.0**	**49.2**

Table 1: Testing accuracy of different size Llama 2 models continue pre-trained with our random walk paths and then supervised fine-tuned. The supervised fine-tuning baseline (SFT) is fine-tuned by the same number of total steps. Results are reported on three math word problem (MWP) datasets.

#Steps	GSM8K	AQUA	SVAMP	Avg.
0	26.8	30.0	53.3	36.7
200	27.5	30.1	53.6	37.1
500	**28.5**	**34.6**	**55.8**	**39.6**
1000	24.9	32.3	51.6	36.3

#Nodes	GSM8K	AQUA	SVAMP	Avg.
0	26.8	30.0	53.3	36.7
10	26.8	30.3	54.8	37.3
50	26.6	29.9	54.7	37.1
100	**28.5**	**34.6**	**55.8**	**39.6**
200	26.6	31.1	52.5	36.7

Table 2: Ablation on the number of random walk training steps M (left) and the number of clusters/nodes K (right).

first $M = 500$ steps of our method are continually pre-trained on random walk data, and then we do 2000 steps of SFT on the original dataset. Our method is about 3% better in accuracy on average over three MWP datasets and two Llama 2 models.

Then we investigate the effect of random walk path length L_{max} by plotting accuracy v.s. path lengths. In Figure 4b, we observe that each dataset has a performance peak at a certain random walk length. While both AQUA and GSM8K peak at path length 10, the SVAMP dataset peaks at path length 5. This is likely related to the different intrinsic reasoning lengths for different datasets. The average length of CoTs in AQUA, GSM8, and SVAMP training sets are 4.79, 3.72, and 1.36, respectively. The reasoning length required for SVAMP is significantly shorter than the other two datasets, thus explaining the earlier peaking. As we discussed in the logical reasoning case, a long random walk may introduce more noise than useful information. Note that even the LM performance can drop after the optimal path length, it is always better than training with path length one. i.e. multi-step random walk always helps.

We also do ablation studies on two critical hyperparameters of our method: the number of steps training on random walk paths M and the number of clusters/nodes K. In Table 2 (left), we show that the optimal number of training steps M is 500 for all three datasets. Since the generated random walk reasoning paths are not natural within small corpora, e.g. the subject might be suddenly changed from one step to another, training too many steps might make the LM overfit the unwanted artifacts. In Table 2 (right), we show that the optimal number of clusters is 100 for all three datasets. Here 0 clusters mean the SFT baseline. Since the datasets we use are small in scale, clustering with a large number of clusters may introduce more noise than useful matchings. We hypothesize that this could be solved by using a larger dataset and more number of clusters/nodes K: in this case, the steps within each node will be more intrinsically similar. This also hints at the potential of our method in the actual pre-training stage: we can view each example in the pre-training corpus as a reasoning path and apply our method.

4 Conclusion

In conclusion, we aim to understand reasoning abilities in language models (LMs), from the perspective of aggregating reasoning paths from pre-training data. The findings shed light on the origins of LLMs' remarkable reasoning capabilities, showcasing the importance of pre-training in acquiring these skills. The construction of the pre-training sequence, such as organizing it as "chains" or random walks on the graph, was found to significantly impact the effectiveness of reasoning. The study also revealed that LM behavior is similar to reason over known facts by aggregating relevant reasoning paths. These insights contribute to our understanding of the underlying mechanisms behind LLMs' reasoning abilities and lead to a potential pre-training data augmentation technique to boost reasoning performance.

REFERENCES

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q Jiang, Jia Deng, Stella Biderman, and Sean Welleck. Llemma: An open language model for mathematics. *arXiv preprint arXiv:2310.10631*, 2023.

Zhen Bi, Ningyu Zhang, Yinuo Jiang, Shumin Deng, Guozhou Zheng, and Huajun Chen. When do program-of-thoughts work for reasoning? *arXiv preprint arXiv:2308.15452*, 2023.

Guillaume Bouchard, Sameer Singh, and Théo Trouillon. On approximate reasoning capabilities of low-rank vector spaces. In *AAAI Spring Symposia*, 2015.

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

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=YfZ4ZPt8zd.

Ta-Chung Chi, Ting-Han Fan, Alexander I Rudnicky, and Peter J Ramadge. Transformer working memory enables regular language reasoning and natural language length extrapolation. *arXiv preprint arXiv:2305.03796*, 2023.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. *arXiv preprint arXiv:1711.05851*, 2017.

Woodrow Denham. Artificial intelligence / machine learning research using the australian aboriginal alyawarra kinship dataset: Partial bibliography 2004-2020. *Mathematical Anthropology and Cultural Theory*, 2020.

Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. Towards revealing the mystery behind chain of thought: A theoretical perspective. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=qHrADgAdYu.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal abstractions of neural networks. *Advances in Neural Information Processing Systems*, 34:9574–9586, 2021.

Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for earning in raph domains. In *Proceedings of the International Joint Conference on Neural Networks*, volume 2, pp. 729 – 734 vol. 2, 01 2005. ISBN 0-7803-9048-2. doi: 10.1109/IJCNN.2005.1555942.

Michael Hanna, Ollie Liu, and Alexandre Variengien. How does gpt-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. *arXiv preprint arXiv:2305.00586*, 2023.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2020.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Samyak Jain, Robert Kirk, Ekdeep Singh Lubana, Robert P Dick, Hidenori Tanaka, Edward Grefenstette, Tim Rocktäschel, and David Scott Krueger. Mechanistically analyzing the effects of fine-tuning on procedurally defined tasks. *arXiv preprint arXiv:2311.12786*, 2023.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213, 2022.

Stanley Kok and Pedro Domingos. Statistical predicate invention. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pp. 433–440, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595937933. doi: 10.1145/1273496.1273551. URL https://doi.org/10.1145/1273496.1273551.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. MAWPS: A math word problem repository. In Kevin Knight, Ani Nenkova, and Owen Rambow (eds.), *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1152–1157, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1136. URL https://aclanthology.org/N16-1136.

Ni Lao, Tom Mitchell, and William W. Cohen. Random walk inference and learning in a large scale knowledge base. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 529–539, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. URL https://aclanthology.org/D11-1049.

Yingcong Li, Kartik Sreenivasan, Angeliki Giannou, Dimitris Papailiopoulos, and Samet Oymak. Dissecting chain-of-thought: A study on compositional in-context learning of mlps. *arXiv preprint arXiv:2305.18869*, 2023.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 158–167, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1015. URL https://aclanthology.org/P17-1015.

Bingbin Liu, Jordan T. Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. Transformers learn shortcuts to automata. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=De4FYqjFueZ.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Eran Malach. Auto-regressive next-token predictors are universal learners. *arXiv preprint arXiv:2309.06979*, 2023.

Shen-yun Miao, Chao-Chun Liang, and Keh-Yih Su. A diverse corpus for evaluating and developing English math word problem solvers. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 975–984, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.92. URL https://aclanthology.org/2020.acl-main.92.

Kanishka Misra, Cicero Nogueira dos Santos, and Siamak Shakeri. Triggering multi-hop reasoning for question answering in language models using soft prompts and random walks. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 972–985, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.62. URL https://aclanthology.org/2023.findings-acl.62.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. Show your work: Scratchpads for intermediate computation with language models, 2022. URL https://openreview.net/forum?id=iedYJm92o0a.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 3806–3824, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.248. URL https://aclanthology.org/2023.findings-emnlp.248.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2080–2094, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.168. URL https://aclanthology.org/2021.naacl-main.168.

Ben Prystawski, Michael Y. Li, and Noah Goodman. Why think step by step? reasoning emerges from the locality of experience. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=rcXXNFVlEn.

Meng Qu, Junkun Chen, Louis-Pascal Xhonneux, Yoshua Bengio, and Jian Tang. Rnnlogic: Learning logic rules for reasoning on knowledge graphs. *arXiv preprint arXiv:2010.04029*, 2020.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Rahul Ramesh, Mikail Khona, Robert P Dick, Hidenori Tanaka, and Ekdeep Singh Lubana. How capable can a transformer become? a study on synthetic, interpretable tasks. *arXiv preprint arXiv:2311.12997*, 2023.

Yasaman Razeghi, Robert L Logan IV, Matt Gardner, and Sameer Singh. Impact of pretraining term frequencies on few-shot reasoning. *arXiv preprint arXiv:2202.07206*, 2022.

Matthew Richardson and Pedro Domingos. Markov logic networks. *Mach. Learn.*, 62(1–2):107–136, feb 2006. ISSN 0885-6125. doi: 10.1007/s10994-006-5833-1. URL https://doi.org/10.1007/s10994-006-5833-1.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15*, pp. 593–607. Springer, 2018.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. Representing text for joint embedding of text and knowledge bases. In Lluís Màrquez, Chris Callison-Burch, and Jian Su (eds.), *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1499–1509, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1174. URL https://aclanthology.org/D15-1174.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. Towards understanding chain-of-thought prompting: An empirical study of what matters. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2717–2739, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.153. URL https://aclanthology.org/2023.acl-long.153.

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

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022a.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022b.

Zhengxuan Wu, Atticus Geiger, Thomas Icard, Christopher Potts, and Noah Goodman. Interpretability at scale: Identifying causal mechanisms in alpaca. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=nRfClnMhVX.

Changnan Xiao and Bing Liu. Conditions for length generalization in learning reasoning skills. *arXiv preprint arXiv:2311.16173*, 2023.

Wenhan Xiong, Thien Hoang, and William Yang Wang. Deeppath: A reinforcement learning method for knowledge graph reasoning. *arXiv preprint arXiv:1707.06690*, 2017.

Fan Yang, Zhilin Yang, and William W Cohen. Differentiable learning of logical rules for knowledge base reasoning. *Advances in neural information processing systems*, 30, 2017.

Kaiyu Yang, Aidan M Swope, Alex Gu, Rahul Chalamala, Peiyang Song, Shixing Yu, Saad Godil, Ryan Prenger, and Anima Anandkumar. Leandojo: Theorem proving with retrieval-augmented language models. *arXiv preprint arXiv:2306.15626*, 2023.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Chuanqi Tan, and Chang Zhou. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*, 2023.

Hattie Zhou, Arwen Bradley, Etai Littwin, Noam Razin, Omid Saremi, Josh Susskind, Samy Bengio, and Preetum Nakkiran. What algorithms can transformers learn? a study in length generalization. *arXiv preprint arXiv:2310.16028*, 2023.

Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, and Jian Tang. Neural bellman-ford networks: A general graph neural network framework for link prediction. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 29476–29490. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/f6a673f09493afcd8b129a0bcf1cd5bc-Paper.pdf.

A PROOF

Proposition A.1. *If LM effectively learned the random walk data distribution through pre-training, we have*

$$KL[P_w(\mathbf{e}|e_1, r), P_{LM}(\mathbf{e}|e_1, r)] \leq KL[P_w(\mathbf{h}|r), P_{LM}(\mathbf{h}|e_1, r)]$$

Proof. Recall that

$$P_{\text{LM}}(e_2|e_1, r) = \sum_{h\in\mathcal{H}} P(e_2|e_1, h)P_{\text{LM}}(h|e_1, r)$$

and

$$P_w(e_2|e_1, r) = \sum_{h\in\mathcal{H}} P(e_2|e_1, h)P_w(h|e_1, r).$$

By log sum inequality, we have:

$$\begin{aligned}
&\text{KL}[P_w(\mathbf{e}|e_1, r), P_{\text{LM}}(\mathbf{e}|e_1, r)] \\
=&\sum_{e_2\in\mathcal{E}} P_w(e_2|e_1, r) \log \frac{P_w(e_2|e_1, r)}{P_{\text{LM}}(e_2|e_1, r)} \\
\leq&\sum_{e_2\in\mathcal{E}}\sum_{h\in\mathcal{H}} P(e_2|e_1, h)P_w(h|e_1, r)\frac{P(e_2|e_1, h)P_{\text{LM}}(h|e_1, r)}{P(e_2|e_1, h)P_w(h|e_1, r)} \\
=&\sum_{e_2\in\mathcal{E}}\sum_{h\in\mathcal{H}} P(e_2|e_1, h)P_w(h|e_1, r)\frac{P_{\text{LM}}(h|e_1, r)}{P_w(h|e_1, r)} \\
=&\sum_{h\in\mathcal{H}}(\sum_{e_2\in\mathcal{E}} P(e_2|e_1, h))P_w(h|e_1, r)\frac{P_{\text{LM}}(h|e_1, r)}{P_w(h|e_1, r)} \\
=&\sum_{h\in\mathcal{H}} P_w(h|e_1, r)\frac{P_{\text{LM}}(h|e_1, r)}{P_w(h|e_1, r)} \\
=&\text{KL}[P_w(\mathbf{h}|r), P_{\text{LM}}(\mathbf{h}|e_1, r)]
\end{aligned}$$

□

B DETAILED DISCUSSION OF RELATED WORK

Theory on LM reasoning Many recent works are investigating LM's reasoning ability. Geiger et al. (2021); Wu et al. (2023) aims to find the causal abstraction of an LM. (Hanna et al., 2023) tries to find circuit for year-span-prediction. Liu et al. (2023); Chi et al. (2023); Feng et al. (2023) show that CoTs enable fixed-size Transformers to perform certain types of reasoning tasks. Li et al. (2023); Razeghi et al. (2022); Wang et al. (2023) try to understand inference time in-context CoT reasoning. Our study is more relevant to the line of work analyzing the contribution of pre-training data to LM reasoning. Bi et al. (2023) analyzes how code data affect program-of-thoughts (Chen et al., 2023) reasoning ability. Xiao & Liu (2023); Zhou et al. (2023) study how reasoning length generalizes from training data. Ramesh et al. (2023) studies LMs' compositional generalization ability. Our hypothesis also echos the conclusion of Malach (2023) that reasoning paths in training data enable supervision on intermediate steps with next-token-prediction objective, and also increase the length complexity, thus reducing time/sample complexity at training time. Prystawski et al. (2023) propose a different hypothesis that localized structure on dependencies between variables in training data is important for LM reasoning, especially CoT reasoning. Our proposed hypothesis echoes theirs and can be shown effective on more realistic data and tasks.

Logic/knowledge graph reasoning Existing methods can be divided into three categories: rule-based, GNN-based (Gori et al., 2005), and LM-based. Markov Logic Network (MLN) (Richardson & Domingos, 2006) and path ranking algorithm (PRA) (Lao et al., 2011) are two classical methods that assign weights to different logical rules. Neural Logic Programming (Yang et al., 2017) and

RNN-logic (Qu et al., 2020) are two neural methods that combine the explainability of learned logical rules and the high performance of neural networks. R-GCN (Schlichtkrull et al., 2018) and NBFNet (Zhu et al., 2021) are two GNN-based methods that train a GNN on the KG and use the obtained triple embeddings. These two category methods either rely on random walks to find paths or use random walks to train GNNs. Recently, LM-based methods are shown to be highly effective on not only KG reasoning (Misra et al., 2023), but more general logical reasoning problems with text descriptions (Pan et al., 2023).

Math reasoning Recently, LM-based math reasoning models have shown to be highly effective (Azerbayev et al., 2023; Yang et al., 2023). Many works have focused on generating high-quality CoT training to improve LM's math reasoning performance (Wang et al., 2022; Nye et al., 2022; Yuan et al., 2023). However, they all rely on the annotated Q-A pairs to generate corresponding paths with LM, which limits the size of augmented data and requires large LMs to do the CoT generation. Our proposed method does not need supervised seed data and thus can be extended to the vast amount of pre-training data. Our method is also lightweight, which only requires a small/medium LM to produce the step embeddings and then do clustering on them.

C Experiment Details

C.1 Datasets

Knowledge graph datasets For KL analysis, we focus on two KGs: Countries (Bouchard et al., 2015) and UMLS (Kok & Domingos, 2007), as they have a reasonable time complexity to compute the aggregated probabilities for long paths. The Countries (Bouchard et al., 2015) contains two relations (`locatedIn` and `neighborOf`) and 227 entities, including countries, regions, and subregions. We use the hardest version (S3) of the Countries. The Unified Medical Language System (UMLS) (Kok & Domingos, 2007) is a more complex KG built from biomedicine knowledge, containing 49 relations and 135 entities. Example entities are diseases and antibiotics, and example relations are treats and diagnoses.

We add three more datasets for computing the prediction accuracy v.s. different random walk path lengths: Kinship (Denham, 2020), NELL-995 (Xiong et al., 2017), and FB15K-237 (Toutanova et al., 2015). The Kinship dataset contains 104 entities and 26 kinship relationships among members of the Alyawarra tribe from Central Australia. The NELL-995 dataset contains 75,492 entities and 200 relations, which is built from the Web via an intelligent agent called Never-Ending Language Learner. The FB15K-237 dataset contains 14,505 entities and 237 relations derived from Freebase. We adopt a processed version of these datasets from Das et al. (2017).

Math word problem datasets We conduct experiments on three math word problem (MWP) datasets: GSM8K (Cobbe et al., 2021), AQUA (Ling et al., 2017), SVAMP (Patel et al., 2021). The Grade School Math dataset (**GSM8K**) contains 8.5K examples of linguistically diverse grade school math world problems. The **AQUA**-RAT dataset contains 100K samples of mathematical problems, along with sequences of human-readable mathematical expressions in natural language. The **SVAMP** dataset is a testing set consisting of elementary-level MWPs. The training set is a combination of simpler MWPs: MAWPS (Koncel-Kedziorski et al., 2016) and ASDiv-A (Miao et al., 2020) with 3.5k training examples in total.

C.2 Training details

Logical reasoning We train randomly initialized GPT-2 (Radford et al., 2019) (124M parameters) with batch size 16 and learning rate 5e-4 using AdamW optimizer (Loshchilov & Hutter, 2017) on one 24G Titan GPU.

Math reasoning We continually train Llama 2 (Touvron et al., 2023b) (7B and 13B parameter versions) with batch size 16 and learning rate 2e-4 using AdamW optimizer (Loshchilov & Hutter, 2017) on one 40G A100 GPU.

D IMPACT STATEMENT

Understanding the reasoning processes of large language models (LLMs) through the lens of aggregating indirect reasoning paths holds potential implications for identifying and mitigating potential biases within LLMs. By formalizing reasoning as random walk paths on knowledge and reasoning graphs, this approach not only elucidates the mechanisms through which LLMs derive conclusions but also sheds light on data and reasoning paths that contribute to their outputs. This insight is crucial for recognizing biases embedded in the training data or in the reasoning process itself. Recognizing these biases is the first step toward developing more equitable and transparent models. By augmenting models with unbiased, unlabeled random walk reasoning paths, we can potentially reduce the influence of biased reasoning patterns and improve the fairness and reliability of LLMs in real-world applications. This research advances our understanding of LLM reasoning capabilities and their implications for bias, paving the way for more responsible AI development and deployment.