Single Sequence Prediction over Reasoning Graphs for Multi-hop QA

Gowtham Ramesh[*], **Makesh Sreedhar**[*], **and Junjie Hu**
University of Wisconsin-Madison
{gramesh4,msreedhar,junjie.hu}@wisc.edu

Abstract

Recent generative approaches for multi-hop question answering (QA) utilize the fusion-in-decoder method (Izacard and Grave, 2021) to generate a single sequence output which includes both a final answer and a reasoning path taken to arrive at that answer, such as passage titles and key facts from those passages. While such models can lead to better interpretability and high quantitative scores, they often have difficulty accurately identifying the passages corresponding to key entities in the context, resulting in incorrect passage hops and a lack of faithfulness in the reasoning path. To address this, we propose a single-sequence prediction method over a local reasoning graph (SEQGRAPH)[1] that integrates a graph structure connecting key entities in each context passage to relevant subsequent passages for each question. We use a graph neural network to encode this graph structure and fuse the resulting representations into the entity representations of the model. Our experiments show significant improvements in answer exact-match/F1 scores and faithfulness of grounding in the reasoning path on the HotpotQA dataset and achieve state-of-the-art numbers on the Musique dataset with only up to a 4% increase in model parameters.

1 Introduction

Multi-hop Question Answering (QA) involves reasoning over multiple passages and understanding the relationships between those pieces of information to answer a question. Compared with single-hop QA, which often extracts answers from a single passage, multi-hop QA is more challenging as it requires a model to determine the relevant facts from multiple passages and connect those facts for reasoning to infer the final answer.

To tackle multi-hop QA, recent works have investigated large pretrained *generative* models (Lewis et al., 2020b; Roberts et al., 2020; Brown et al., 2020) and demonstrated their effectiveness over traditional *extractive* models (Chen et al., 2017). Compared with extractive models, the ability of generative models to effectively aggregate and combine evidence from multiple passages proves advantageous for multi-hop QA. In particular, Izacard and Grave (2021) propose a method called FID (Fusion-in-Decoder), which leverages passage retrieval with a generative model, such as T5 (Raffel et al., 2020) or BART (Lewis et al., 2020a), to achieve state-of-the-art performance on various single-hop QA tasks. However, this approach does not extend well to multi-hop QA tasks (Yavuz et al., 2022), as it sorely relies on a black-box generative model to generate answers directly without explicitly modeling the multi-hop reasoning process. Additionally, FID encodes multiple context passages independently for multi-hop QA, ignoring the structural and semantic relationship between these passages (Yu et al., 2022). Building on FID, PATH-FID (Yavuz et al., 2022) addresses the interpretability issue by training a model to generate a reasoning path that contains supporting pas-

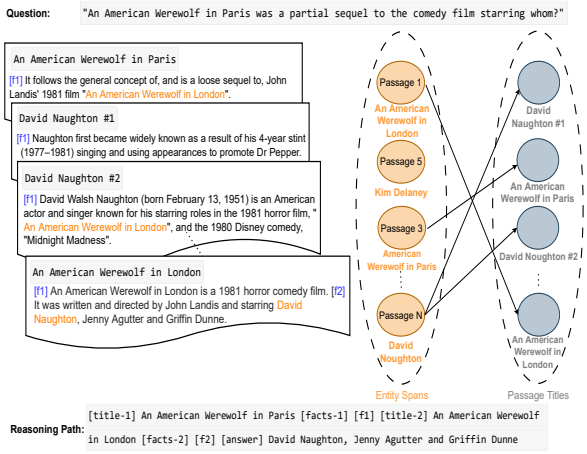

Figure 1: Localized graph construction connecting entity spans to corresponding passages in the context. If there are multiple passages with the same title, we connect the entity span to all such passages.

[*] Equal contribution
[1] Code/Models will be released at https://github.com/gowtham1997/SeqGraph

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 11466–11481
July 9-14, 2023

sage titles, facts, and the final answer. However, our analysis of PATH-FID outputs shows *disconnected reasoning* with incorrect passage hops in the model's reasoning path which affects final answer generation. Recently, there have been multiple techniques (Jiang and Bansal, 2019; Lee et al., 2021; Ye et al., 2021) to counter disconnected reasoning which operate at the dataset level, using adversarial training, adding extra annotations or using dataset rebalancing for training. While these approaches optimize models to mitigate disconnected reasoning (Trivedi et al., 2020), the performance on the original test set often suffers from a significant decrease.

In this paper, we propose a single-**seq**uence prediction method over a local reasoning **graph** (SEQGRAPH) that integrates a graph structure connecting key entities in each context passage to relevant subsequent passages for each question. Different from the prior works, our method not only mitigates the disconnected reasoning issue but also maintains robust performance on the original dataset. Intuitively, for each multi-hop question, our method leverages the structural relationship between different passages to learn structured representations through a graph neural network (GNN) (Hamilton et al., 2017; Kipf and Welling, 2017). The structured representations are fused to bias the generative model toward predicting a faithful, connected reasoning path which improves answer predictions. Our experiments on the HOTPOT-QA dataset (Yang et al., 2018) show clear improvements in exact-match(EM)/F1 scores compared to generative baselines in the *distractor* setting while minimizing disconnected reasoning quantified by the DIRE score (Trivedi et al., 2020). We also achieve the state-of-the-art performance on the MUSIQUE-Answerable test dataset (Trivedi et al., 2022a) with a 17-point improvement in answer F1 over the current best-performing model in the end-to-end (E2E) category.

To summarize, our contributions are as follows:

- We propose an interpretable single-**seq**uence prediction approach over local reasoning **graph**s, SEQGRAPH, to bias the model representations
- SEQGRAPH achieves notable performance improvements on two multi-hop QA benchmarks, HOTPOT-QA and MUSIQUE (SOTA), with only a minimal increase in the model size.
- SEQGRAPH reduces disconnected reasoning as measured by DIRE score while maintaining strong performance gains on the original dataset.

2 Preliminaries

Problem Setup: In a multi-hop QA task, each QA pair in a labeled dataset $\mathcal{D}$ is given along with a set of N passages, $\mathcal{P}_q = \{p_1, p_2, ..., p_N\}$, *i.e.*, $(q, a, \mathcal{P}_q) \in \mathcal{D}$, where a passage has its title and content $p_i = (t_i, c_i)$. The task is to learn a model parameterized θ to generate an answer string a for the given question q and $\mathcal{P}_q$. In this paper, we focus on the *distractor* setting, where $\mathcal{P}_q$ is given for each question and contains m distractors that are not useful to the answer prediction. Thus, this task requires a model to reason over multiple hops of the remaining $N - m$ relevant passages. In addition to predicting the final answer a, we also aim to train a model to predict a *reasoning path* R of important elements (*e.g.*, relevant passage titles, supporting facts in a passage) that lead to the final answer.

Multi-hop QA as Single Sequence Generation: Recent generative question answering (QA) approaches (e.g., FID (Izacard and Grave, 2021), PATH-FID (Yavuz et al., 2022)) utilize an encoder-decoder model as the backbone to generate answers in a single text sequence. In particular, FID is one of the popular formulations. Specifically, for each passage $p_i = (t_i, c_i) \in \mathcal{P}_q$ of a question q, FID encodes a combined sequence of the question, the passage title and contents into an embedding. These embeddings for all passages are concatenated as inputs to the decoder for generating the final answer.

PATH-FID builds upon this by explicitly modeling a reasoning path as part of the generation output in addition to the answer. Specifically, special index tokens $[f_i]$ are added to demarcate all sentences in each passage context. The sentences supporting the prediction of a final answer are considered facts. The decoder is then trained to generate the reasoning path R as a linearized sequence consisting of the passage titles and the index tokens of facts used within those passages to obtain the final answer. Figure 1 shows an example of a reasoning path.

Disconnected Reasoning in PATH-FID: Since the model predictions now include the reasoning path, we can analyze which facts in the passage are utilized by the model to determine the next passage to hop to and arrive at the final answer. For a perfectly faithful model, all predictions with correct answers should have correctly identified passages and facts. However, due to the presence

of shortcuts in the datasets as well as the model's predicted reasoning path not being faithful, we observe model predictions containing correct final answers but incorrect identification of passage titles or facts. This unfaithful prediction issue is referred to as *disconnected reasoning* (Trivedi et al., 2020). Different from PATH-FID, we use the presence of a local graph structure between different passages in the context to bias the representations of the model and help alleviate this problem.

3 Method

In this section, we describe our proposed method for solving disconnected reasoning for multi-hop QA in the *distractor* setting.

Overview: Our method first constructs a local graph over passage contexts for each question (§3.1), and integrates the graph information with the key entities to improve the generation of reasoning paths (§3.2). Different from prior works that encode all the passages independently, we connect the passages through the key pivot entities into a local graph for a question, which allows us to encode structural representations across passages by a graph neural network. These graph structured representations are then fused with the contextualized text representations from a text encoder, guiding the model to leverage structural information to alleviate disconnected reasoning over passages.

3.1 Graph Construction

In contrast to the *full-wiki* setting where a model must retrieve relevant passages from Wikipedia or a large corpus, the distractor setting provides the model with a list of N passages $\mathcal{P}_q$ consisting of $N - m$ relevant passages and m distractors for each question q. Conventionally, these passages are collected from Wikipedia, as Wikipedia remains one of the largest faithful knowledge sources available for public usage. Even for text passages out of Wikipedia, there are existing out-of-box entity linkers (e.g., SLING (Ringgaard et al., 2017), BLINK (Wu et al., 2020)) that can identify key entities from texts and link them to their Wikipedia pages. As a result, each provided passage may contain pivot entities with hyperlinks connecting to their corresponding Wikipedia pages. We exploit such entity hyperlinks to construct a local directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$ containing two types of nodes (*i.e.*, entities and passage titles) and links between these nodes. Specifically, for each pivot entity e in a passage p_i, we create a link from e to the title t_j of another passage p_j (denoted as $l_{e \to t_j}$) whenever the entity span e points to a Wikipedia article that contains the passage p_j.

For example, an entity span *"David Noughton"* appears in the passage context: *"An American Werewolf in London is a 1981 horror comedy film starring David Noughton, Jenny Agutter. ..."* This entity would be connected to a passage with the title of *"David Walsh Noughton"*, forming the link (David Noughton[Entity] $\rightarrow$ David Walsh Noughton[Passage]). If there are multiple passages with the title *"David Walsh Noughton"* among the N passages, the entity span would be connected to all of them with distinct links. Figure 1 shows an example of an entity-passage graph.

3.2 Entity-to-Passage Fusion

Next, we describe how we encode such a local directed graph into vector representations for all nodes and fuse these node representations with the contextualized text representations of the corresponding entities from the language model.

We utilize the same model as PATH-FID with a pre-trained T5 model as our backbone architecture. The input for this method consists of the N sequences, where each sequence is a concatenation of the question q, the title and contents of a passage p_i from the collection $p_i \in \mathcal{P}_q$ together with their indicator tokens, denoted as S_i below:

$$S_i := [\texttt{Question}]\ q\ [\texttt{Title}]\ t_i\ [\texttt{Content}]\ c_i \quad (1)$$

Given the T5's encoder of M transformer layers, we first encode S_i through the first L layers to obtain the intermediate hidden representations $\mathbf{Z}_i^L$ in Eq. (2), which capture the shallow contextualized information of the input sequence.

$$\mathbf{Z}_i^L = \text{TextEncoder}(S_i, L) \quad (2)$$

We utilize these shallow representations to initialize the node embeddings for a graph neural network. Specifically, we extract the representations of the entity spans or passage title spans (*i.e.*, nodes in the graph $\mathcal{G}$) from $\mathbf{Z}_i^L$ according to their span positions $[a, b]$ in S_i. Next, for a text span $S_{i,a:b}$ representing either an entity or a title in S_i, we average the extracted representations of the text span to obtain an initial node embedding, *i.e.*, $\mathbf{n} = \text{avg}(\mathbf{Z}_{i,a:b}^L)$. Finally, we stack the initial embeddings for all nodes denoted as $\mathbf{N}$ and apply

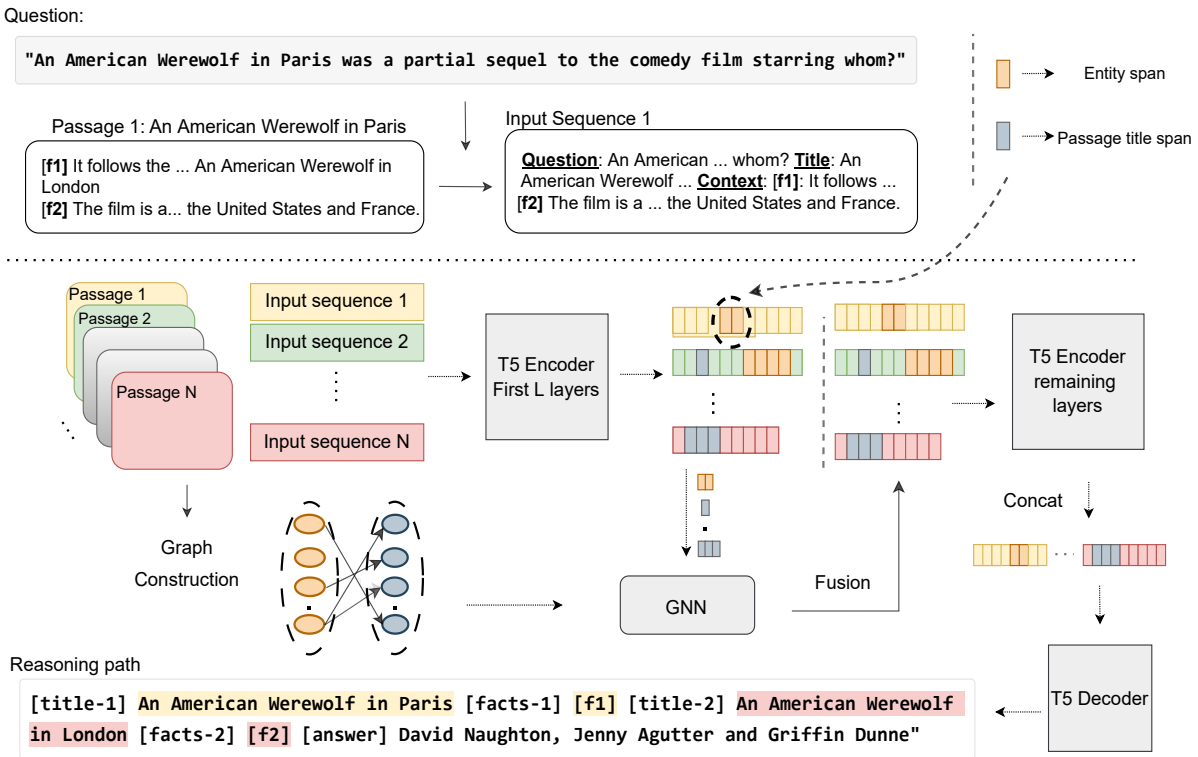

Figure 2: Given a question and the supporting passages, we construct a localized entity-passage graph. The representations from the L^{th} layer of the language model is used to initialize the weights of a graph neural network(GNN) and it is used to perform message passing on the constructed local graph. The representations for the entity spans and titles from the GNN are added to the LM representations and passed through the remaining $M - L$ layers of the encoder. The T5 decoder performs cross-attention on the final hidden states from the encoder and generates the reasoning path with the final answer.

a graph neural network (GNN) to further encode the structural embeddings on the graph $\mathcal{G}$:

$$\mathbf{Z}^G = \text{GraphEncoder}(\mathbf{N}, \mathcal{G}) \quad (3)$$

As we record the text span position $[a, b]$ for each node in $\mathcal{G}$, we can leverage the node embeddings $\mathbf{Z}^G$ to construct a new structured representation $\mathbf{Z}^G_i$ (with the same size as $\mathbf{Z}^L_i$) for each sequence S_i where we fill in the node embeddings from $\mathbf{Z}^G$ to their corresponding text span positions $[a, b]$ in S_i and fill in 0 to the other non-span positions.

Finally, we fuse the contextualized text representations $\mathbf{Z}^L_i$ from the text encoder and the structured node representations $\mathbf{Z}^G_i$ by an aggregation operator $\oplus$, and pass them to the remaining layers of the text encoder to obtained the fused representations $\mathbf{S}_i$ for each input sequence S_i:

$$\mathbf{S}_i = \text{TextEncoder}(\mathbf{Z}^G_i \oplus \mathbf{Z}^L_i, M - L) \quad (4)$$

In this work, the aggregation operator used is a simple addition. Complex aggregation mechanisms such as learning a weighted combination of the representations can be explored in future work.

We concatenate the fused representations $\mathbf{S}_i$ from all of the N context sequences to form $\mathbf{S} = [\mathbf{S}_1; \mathbf{S}_2 \cdots; \mathbf{S}_N]$. Subsequently, $\mathbf{S}$ is passed as inputs to the T5 decoder that estimates the conditional probability $P_\theta(R|\mathbf{S})$ of predicting a reasoning path R. Depending on the annotations in different datasets, a reasoning path R can take various formats. For example, the reasoning path takes the form "$R :=$ [title] t_i [facts] f_i [answer] a" for HOTPOT-QA and "$R :=$ [title] t_i [intermediate_answer] ans$_i$ [answer] a" for MUSIQUE. We also investigate variants of reasoning paths for MUSIQUE in our experiments. As we can construct ground-truth reasoning paths R^* during training, the model is optimized using a cross-entropy loss between the conditional probability $P_\theta(R|\mathbf{S})$ and R^*.

4 Experimental Setting

In this section, we elaborate on the datasets, the baseline models and the variants of SEQGRAPH we consider for our experiment settings. We consider two multi-hop QA datasets, HOTPOT-QA and MUSIQUE. Since SEQGRAPH is primarily focused only on improving the efficacy of encoding, we consider only the *distractor* setting for both datasets. Table 4 shows the standard train/dev/test statistics.

HOTPOT-QA: The final answer to each question in the distractor setting is extracted from 10 passages. The dataset includes two main types of questions: bridge (80%) and comparison (20%). Bridge questions often require identifying a bridge entity in the first passage to correctly hop to the second passage that contains the answer, while comparison questions do not have this requirement. Each question is also provided with annotations of 2 supporting passages (2-hop) and up to 5 corresponding relevant sentences as their supporting facts.

MUSIQUE: MUSIQUE has questions that range in difficulty from 2 to 4-hops and six types of reasoning chains. MUSIQUE uses a stringent filtering process as well as a bottom-up technique to iteratively combine single-hop questions from several datasets into a k-hop benchmark that is more difficult than each individual dataset and significantly less susceptible to the disconnected-reasoning problem. Unlike HOTPOT-QA, MUSIQUE does not provide annotations of relevant sentences but provides supporting passage titles, question decomposition(decomposition of a multi-hop question into simpler 1-hop sub-questions) and also intermediate answers to the decomposed questions. Given this variety, we use the following reasoning path variants to train the model to generate:

- DA: Question decomposition and final answer
- SA: Supporting titles and final answer
- SIA: Supporting titles, intermediate answers and final answer
- DSIA: Question decomposition, supporting titles, intermediate answers and final answer

Table 6 shows an example of different reasoning paths. While the last variant (predicting every decomposition/intermediate answer or support title) is more interpretable, it encounters the challenge of producing a long sequence. SIA is our best-performing reasoning path variant which is used for all of our results and analysis.

4.1 Models in Comparison

Our main baselines are generative approaches to multi-hop QA that include and build upon the FID approach. For all of the models, we use the pre-trained T5 encoder-decoder as the backbone and consider two sizes—base and large variants.

- FID: Model generation includes only the final answer.
- PATH-FID: Model generation includes the reasoning path as well as the final answer.
- SEQGRAPH: Model that utilizes a fusion of representations from the language model and the Graph Neural Network. Similar to PATH-FID, we train the model to generate the reasoning path in addition to the final answer.

4.2 Evaluation Metrics

For both HOTPOT-QA and MUSIQUE, we use the standard quantitative metrics of exact-match and F1 scores to evaluate the quality of predicted answers. For models that predict the reasoning path in addition to the final answer, we can quantify how accurately they can identify the supporting facts (or supporting titles for MUSIQUE) using the Support-EM and Support-F1 scores Yang et al. (2018).

To quantify the level of disconnected reasoning, we compute dire F1 scores on the answer spans (**Answer**), supporting paragraphs (**Supp**$_\text{p}$), supporting sentences (**Supp**$_\text{s}$), joint metrics (**Ans+Supp**$_\text{p}$, **Ans+Supp**$_\text{s}$) of the Dire HOTPOT-QA subset.

4.3 Implementation details

We train all models using an effective batch size of 64. We use an initial learning rate of 1e-4, a linear rate scheduler, a warmup of 2,000 steps (1,000 steps for MUSIQUE), and finetune the models for 10 epochs. For SEQGRAPH, we use GAT (Veličković et al., 2017) for our GNN layers. A maximum sequence length of 256 tokens is used for constructing the input. All experiments have been conducted on a machine with either 4×40G A100 GPUs or 4×80G A100 GPUs. A detailed list of hyperparameters can be found in Appendix E.

5 Results and Analysis

In this section, we present the main results of the baselines and our proposed approach on HOTPOT-QA and MUSIQUE (§5.1), and then perform fine-grained analysis thereafter.

Model	HotpotQA				MuSiQue			
	Answer		Support		Answer		Support	
	EM	F1	EM	F1	EM	F1	EM	F1
FiD-Base	61.84	75.20	-	-	29.38	39.97	-	-
Path-FiD-Base	62.03	75.69	60.45	86.00	34.71	44.93	57.30	80.18
SeqGraph-Base	**64.19**	**77.60**	**62.44**	**87.72**	**37.36**	**47.11**	**58.05**	**80.39**
FiD-Large	65.59	79.39	-	-	36.04	46.66	-	-
Path-FiD-Large*	65.80	78.90	59.30	85.70	-	-	-	-
Path-FiD-Large	65.33	79.00	61.52	86.88	42.28	53.86	62.14	82.45
SeqGraph-Large	**66.51**	**81.62**	**63.24**	**88.28**	**46.01**	**56.88**	**65.12**	**83.65**

Table 1: Performance on the dev set of HotpotQA and MuSiQue. Since FiD does not predict a reasoning path, we do not compute the Support EM and F1 scores. Path-FiD-Large* indicates the numbers reported from Yavuz et al. (2022), while the other numbers are from our reimplementation

5.1 Multi-hop Performance

The quantitative performance of the models in terms of exact-match and F1 scores for both the final answer and the predicted supports are shown in Table 1. We find that across both model sizes (Base and Large), explicitly predicting the reasoning path helps Path-FiD in improving the answer EM and F1 scores over the vanilla FiD approach. By biasing the model with graph representations, SeqGraph outperforms the baselines on both the HotpotQA and the MuSiQue datasets.

SeqGraph achieves a 2-point improvement in both answer and support EM when considering the base variant and 1.5 point improvement for the large variant on the dev set of HotpotQA.

On the more challenging MuSiQue dataset, we observe stronger results from SeqGraph where it records up to a 4-point improvement in both answer and support scores across both model sizes on the dev set. On the test set (in Table 8 of the appendix), the current best performing approach is a two stage RoBERTa/ Longformer-Large model, Select-Answer, where the passage selection/ranking and answer generation stage is optimized separately using different models. SeqGraph-Large achieves state-of-the-art numbers on Answer-F1 with a 5-point improvement over the Select-Answer model[2] even though it is a single stage approach. When comparing with the top score in the end-to-end (E2E) category which all of our models belong to, SeqGraph gets a massive 17-point improvement in answer F1 and a 9-point improvement in support F1 establishing the efficacy of our approach. It should also be noted that all of the current models on the leaderboard are discriminative approaches with an encoder-only model (Longformer-Large) encoding a very long context length of 4,096, while all of our models are generative in nature with a much smaller context length of 256. MuSiQue is also designed to be more challenging than HotpotQA and explicitly tackles the issue of disconnected reasoning during dataset curation, making it harder for the model to take shortcuts and cheat. The larger performance improvements of SeqGraph on MuSiQue compared to HotpotQA showcases the advantage of our proposed approach, providing promising results for further research in this direction to mitigate disconnected reasoning.

5.2 Faithfulness of Reasoning Paths

We follow Yavuz et al. (2022) to perform analysis at the passage and individual fact level to determine how faithful the generated reasoning paths are across different models.

Predicted Answer in Predicted Titles/Support: *how often are the predicted answers found in one of the predicted passages or in the predicted supporting facts*[3].

Gold Answer in Predicted Titles/Support: *how often are the gold answers found in one of the predicted passages or in the predicted supporting facts.*

Predicted Answer in Gold Titles/Support: *how often are the predicted answers found in one of the gold passages or in the gold supporting facts.*

Figure 3 shows the described faithfulness metric scores on HotpotQA. We find that SeqGraph

[2]https://leaderboard.allenai.org/musique_ans/

[3]We do this analysis only on Bridge type questions where the final answer span can be found in context passages, unlike comparison questions where the final answer is usually *yes/no*

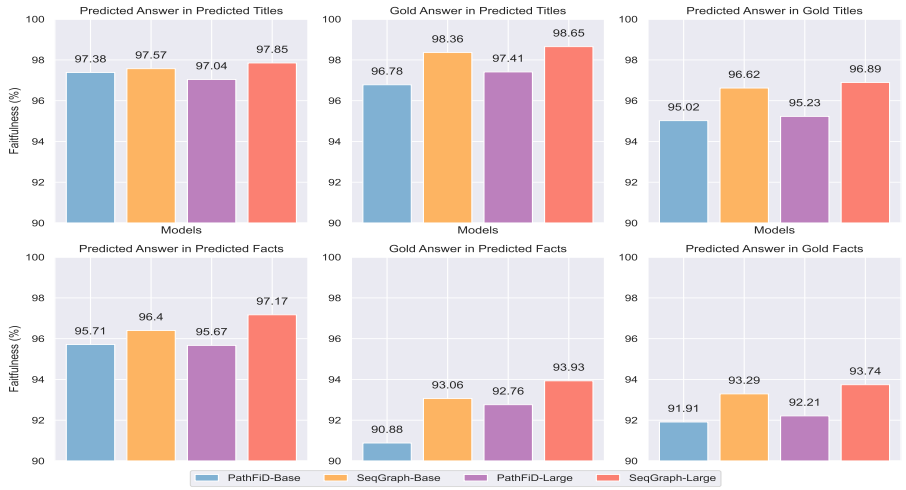

Figure 3: Comparison of model faithufulness on HOTPOT-QA. We find that SEQGRAPH improves over PATH-FID consistently across all categories.

Model	Answer ↓	Supp_P ↓	Supp_S ↓	Ans + Supp_P ↓	Ans + Supp_S ↓
FID-Base	51.1	-	-	-	-
PATH-FID-Base	45.5	48	49.1	22.6	24.3
SEQGRAPH-Base	44.7	46.2	45.4	21.8	22.8
FID-Large	53.5	-	-	-	-
PATH-FID-Large	48.8	48.3	49.7	24.3	26.4
SEQGRAPH-Base	45.7	45.9	45.3	22.3	23.4

Table 2: DIRE score (F1 scores) for various models on the probe dataset of HOTPOT-QA indicating the extent of disconnected reasoning. Lower the score, the better the model.

is more faithful with a 0.5-1.5% improvement over PATH-FID across all considered categories.

5.3 Performance vs Number of hops

We break down the final answer exact-match and F1 scores based on how many supporting facts(or titles for Musique) are required to answer the question. Figure 5 shows this performance breakdown for HOTPOT-QA and Figure 6 shows it for MUSIQUE. We observe that SEQGRAPH improves over PATH-FID in the cases where the support includes two or three supporting facts (or titles), but the answer EM takes a hit when the number of supporting facts(titles) ≥ 4. We notice that SEQGRAPH has a higher support EM over PATH-FID in such cases where shortcuts may exist in the dataset and PATH-FID relies on those shortcuts to get a higher answer EM but a lower support EM. Section §5.4 quantifies the extent to which PATH-FID suffers from disconnected reasoning as compared to SEQGRAPH.

5.4 Probing Disconnected Reasoning

HOTPOT-QA suffers from information leakage in the form of reasoning shortcuts leading to *disconnected reasoning*. This affects the generalization capability of such models and inflates the performance on the evaluation sets. Table 4 shows some qualitative examples of disconnected reasoning in PATH-FID that are avoided by SEQGRAPH

Trivedi et al. (2020) construct a probe of HOTPOT-QA by splitting the two supporting paragraphs for the original question across two questions. If the model can answer modified questions correctly without the complete context, it suggests that the model uses disconnected reasoning for the original question. By measuring the performance of a model on such a dataset, we arrive at the DIRE score with a higher value implying more disconnected reasoning. Table 2 shows the DIRE scores for the various models. We see that SEQGRAPH resorts to lower disconnected reasoning compared to PATH-FID while maintaining strong performance gains on the original evaluation set.

5.5 Comparison with PathFiD+

Yavuz et al. (2022) extend PATH-FID and introduce PATH-FID + to improve the cross-passage interactions before feeding to the FiD decoder and show an improvement of 7 EM points and achieve state-of-the-art results on HOTPOT-QA distractor

Question: What is the name of the executive producer of the film that has a score composed by Jerry Goldsmith?
Answer: Ronald Shusett
Passages:
Alien (soundtrack): [f1] The iconic, avant-garde score to the film Alien was composed by Jerry Goldsmith … [f2] The … cues.
Alien (film): [f1] Alien is a 1979 … [f3] … co-authored with Ronald Shusett … [f6] Shusett was executive producer …
L.A. Confidential: [f1] L.A. Confidential … [f2] … score was composed by Jerry Goldsmith.
Lionheart (1987 film): [f1] Lionheart, … [f4] The composer Jerry Goldsmith wrote the score …
…
Gold Reasoning Path: [title-1] Alien (soundtrack) [facts-1] [f1] [title-2] Alien (film) [facts-2] [f3] [f6] [answer] Ronald Shusett
Pathfid-large Pred: [title-1] L.A. Confidential [facts-1] [f1] [f2] [title-2] Lionheart (1987 film) [facts-2] [f1] [f2] [f4] [answer] Steven Spiel
SeqGraph Large Pred: [title-1] Alien (soundtrack) [facts-1] [f1] [title-2] Alien (film) [facts-2] [f6] [answer] Shusett

---

Question: What is the first two words of the fifth studio album of Joseph Edgar Foreman?
Answer: The Hungry
Passages:
The Hungry Hustlerz: Starvation Is Motivation: [f1] The Hungry Hustlerz …album by Afroman
Afroman: [f1] Joseph Edgar Foreman … stage name Afroman … [f3] …
Como Ama una Mujer: [f1] … fifth studio album … by actress Jennifer Lopez … [f4] …
…
Gold Reasoning Path: [title-1] Afroman [facts-1] [f1] [f2] [title-2] The Hungry … Is Motivation [facts-2] [f1] [answer] The Hungry
Pathfid-large Pred: [title-1] Afroman [facts-1] [f1] [title-2] Como Ama una Mujer [facts-2] [f1] [answer] "Como Ama una Mujer"
SeqGraph Large Pred: [title-1] Afroman [facts-1] [f1] [title-2] The Hungry … Is Motivation [facts-2] [f1] [answer] The Hungry Hustlerz: Starvation Is Motivation

---

Question: What creature of American folklore gained notoriety in 1964?
Answer: Dewey Lake Monster
Passages:
Dewey Lake Monster: [f1] The Dewey Lake Monster … June 1964 … [f2] … also referred to as the Michigan Bigfoot and Sister Lake ..
Bigfoot: [f1] Bigfoot … creature of American folklore … [f3] …
Chessie (sea monster): [f1] In American folklore, Chessie is a sea monster … 1977 and more in the 1980s …
…
Gold Reasoning Path: [title-1] Bigfoot [facts-1] [f1] [title-2] Dewey Lake Monster [facts-2] [f1] [f2] [answer] Dewey Lake Monster
Pathfid-large Pred: [title-1] Chessie (sea monster) [facts-1] [f6] [title-2] Dewey Lake Monster [facts-2] [f1] [answer] Dewey Lake Monster
SeqGraph Large Pred: [title-1] Bigfoot [facts-1] [f1] [title-2] Dewey Lake Monster [facts-2] [f1] [f2] [answer] Bigfoot

Figure 4: Qualitative Analysis of Disconnected Reasoning in HOTPOT-QA. Correct/Incorrect hops from entity spans to Passage titles for different cases are shown. In the first two cases, disconnected reasoning by PATH-FID leads to incorrect final answer while SEQGRAPH gets the path and answer correct. The third case shows PATH-FID getting the final answer right despite the reasoning path being disconnected while SEQGRAPH gets the connected reasoning path right.

dataset. However, we find the following limitations of the approach:

Hop-assumption: PATH-FID + adds pairs of contexts as input to the FID encoder, which assumes a fixed number of hops (in case of HOTPOT-QA, two) and doubles the input sequence length, leading to increased training time.

Multi-step: To efficiently encode pairs of passages (instead of inefficient $\binom{N}{2}$ passages, where N is the total number of passages), PATH-FID + also needs to run the vanilla PATH-FID or train another model to choose the first relevant context $P*$ to jump to and then construct pairs ($P*$, P_n). This makes it inefficient and not scalable to questions with higher hops or complex datasets like MUSIQUE

In contrast, our approach does not make any assumptions about the number of hops and is scalable. It produces output in a single shot without requiring multiple steps or increased sequence length. While PATH-FID + may achieve stronger performance in 2-hop HOTPOT-QA, our proposed method is more general, efficient and scalable, making it a more practical solution for real-world applications and also easily extendable to open-domain setting.

6 Related Works

Multihop question answering requires a model to perform reasoning over multiple pieces of information, utilizing multiple sources and inferring relationships between them to provide a correct answer to a given question. There have been various approaches and datasets proposed for training QA systems, such as HotpotQA (Yang et al., 2018), IIRC(Ferguson et al., 2020) and Musique (Trivedi et al., 2022b).

In the HOTPOT-QA full-wiki setting, the task is to find relevant facts from all Wikipedia articles and then use them to complete the multi-hop QA task. Retrieval models play an important role in this setting, such as DPR (Karpukhin et al., 2020), which focuses on retrieving relevant information in the semantic space. Other methods, such as Entities-centric (Das et al., 2019), and Golden Retriever (Qi et al., 2019), use entities mentioned or reformulated in query keywords to retrieve the next hop

document. Additionally, PathRetriever (Asai et al., 2020) and HopRetriever (Li et al., 2020) use RNN to select documents to form a paragraph-level reasoning path iteratively. The above methods mainly focus on the open-domain setting (full-wiki) and improve the retriever's performance and do not address the disconnected reasoning problem.

Multiple techniques (Jiang and Bansal, 2019; Lee et al., 2021; Ye et al., 2021) to counter disconnected reasoning operate at the dataset level, using adversarial training, adding extra annotations or using dataset augmentations to get a balanced train set and prevent the model from cheating.

We highlight differences between our approach and other related works on HOTPOT-QA-distractor and other works that combine language models with graphs below :

Generative approaches: Our generative-FiD approach differs from others using KG/GNN (Ju et al., 2022; Yu et al., 2022) as we use an entity-passage graph with Wikipedia hyperlinks. Also, our focus is primarily on the distractor setting of multi-hop QA, while other baselines (Ju et al., 2022; Yu et al., 2022) are either single-hop or improving retrieval in open-domain setting

Pipeline vs single-stage: Other baselines (Tu et al., 2019; Chen et al., 2019; Qiu et al., 2019; Wang et al., 2021; Li et al., 2023) use a pipeline approach with distinct encoder models in the reasoning process, while we use a single-stage, one-shot prediction process without assumptions on the number of hops.

Graph construction: Other methods (Tu et al., 2019; Qiu et al., 2019) select relevant passages heuristically from among distractors to construct graphs. However, we construct our entity-passage graph on all passages (including distractors) and fuse the representations in the encoder.

While a direct comparison with pipeline-based approaches is not possible or fair, we provide comparisons in Table 3 for completeness.

Model	F1	Support F1
DFGN(Qiu et al., 2019)	69.69	81.62
SAE-Large(Tu et al., 2019)	80.75	87.38
SEQGRAPH-Base (T5-base)	77.6	87.72
SEQGRAPH-Large (T5-large)	81.62	88.28
C2FM-F1(Wang et al., 2021) (Electra large + DebertaV2 xx-large)	84.65	90.08
FE2H(Li et al., 2023) (iterative Electra Large + Albert-xxlarge-v2)	84.44	89.14

Table 3: F1 scores of different related works on HOTPOT-QA distractor dataset

7 Conclusion

In this paper, we propose SEQGRAPH, an approach that utilizes the structured relationship between passages in the context of multi-hop questions to reduce disconnected reasoning. We construct a localized entity-passage graph using Wikipedia hyperlinks, encode it using a GNN, and fuse the structured representations with the text encoder for predicting a reasoning path. Our approach results in strong performance gains in terms of both answer and support EM/F1 on HOTPOT-QA and reduces disconnected reasoning measured using DIRE score. We also obtain state-of-the-art performance on the more challenging MUSIQUE benchmark with a 17-point improvement in answer F1 over the current best end-to-end(E2E) model. Experimenting with sophisticated methods of encoding the graph structure and fusing the text and graph representations can be explored in future work.

Limitations

We identify the following limitations of our work:

Longer Output Sequences While outputting the reasoning path as a single short sequence makes the model more interpretable, it increases the challenge of producing a long /coherent sequence when the question is complex (more than 3 hops). Producing a longer sequence also increases the inference time. Simplifying this output while not sacrificing interpretability is a good future direction

Entity Identification Our method needs wikipedia outlinks or a entity linker to construct a localized graph for every question. Generalizing this step by pretraining the model to do entity linking (Févry et al., 2020; Sun et al., 2021; Verga et al., 2020) might eliminate the need to use an external module.

References

Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. 2020. Learning to retrieve reasoning paths over wikipedia graph for question answering. In *International Conference on Learning Representations*.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.

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.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Jifan Chen, Shih-Ting Lin, and Greg Durrett. 2019. Multi-hop question answering via reasoning chains. *ArXiv*, abs/1910.02610.

Rajarshi Das, Ameya Godbole, Dilip Kavarthapu, Zhiyu Gong, Abhishek Singhal, Mo Yu, Xiaoxiao Guo, Tian Gao, Hamed Zamani, Manzil Zaheer, and Andrew McCallum. 2019. Multi-step entity-centric information retrieval for multi-hop question answering. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 113–118, Hong Kong, China. Association for Computational Linguistics.

James Ferguson, Matt Gardner, Hannaneh Hajishirzi, Tushar Khot, and Pradeep Dasigi. 2020. IIRC: A dataset of incomplete information reading comprehension questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1137–1147, Online. Association for Computational Linguistics.

Thibault Févry, Livio Baldini Soares, Nicholas FitzGerald, Eunsol Choi, and Tom Kwiatkowski. 2020. Entities as experts: Sparse memory access with entity supervision. In *Conference on Empirical Methods in Natural Language Processing*.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Yichen Jiang and Mohit Bansal. 2019. Avoiding reasoning shortcuts: Adversarial evaluation, training, and model development for multi-hop QA. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2726–2736, Florence, Italy. Association for Computational Linguistics.

Mingxuan Ju, Wenhao Yu, Tong Zhao, Chuxu Zhang, and Yanfang Ye. 2022. Grape: Knowledge graph enhanced passage reader for open-domain question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 169–181, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.

Kyungjae Lee, Seung-won Hwang, Sang-eun Han, and Dohyeon Lee. 2021. Robustifying multi-hop QA through pseudo-evidentiality training. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6110–6119, Online. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Patrick S. H. 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. 2020b. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Shaobo Li, Xiaoguang Li, Lifeng Shang, Xin Jiang, Qun Liu, Chengjie Sun, Zhenzhou Ji, and Bingquan Liu. 2020. Hopretriever: Retrieve hops over wikipedia to answer complex questions.

Xin-Yi Li, Wei-Jun Lei, and Yu-Bin Yang. 2023. From easy to hard: Two-stage selector and reader for multi-hop question answering. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Peng Qi, Xiaowen Lin, Leo Mehr, Zijian Wang, and Christopher D. Manning. 2019. Answering complex open-domain questions through iterative query generation.

Lin Qiu, Yunxuan Xiao, Yanru Qu, Hao Zhou, Lei Li, Weinan Zhang, and Yong Yu. 2019. Dynamically fused graph network for multi-hop reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6140–6150, Florence, Italy. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Michael Ringgaard, Rahul Gupta, and Fernando CN Pereira. 2017. Sling: A framework for frame semantic parsing. *arXiv preprint arXiv:1710.07032*.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Haitian Sun, Patrick Verga, Bhuwan Dhingra, Ruslan Salakhutdinov, and William W Cohen. 2021. Reasoning over virtual knowledge bases with open predicate relations. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9966–9977. PMLR.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2020. Is multihop QA in DiRe condition? measuring and reducing disconnected reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8846–8863, Online. Association for Computational Linguistics.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022a. MuSiQue: Multi-hop Questions via Single-hop Question Composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022b. MuSiQue: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Ming Tu, Kevin Huang, Guangtao Wang, Jing Huang, Xiaodong He, and Bowen Zhou. 2019. Select, answer and explain: Interpretable multi-hop reading comprehension over multiple documents. In *AAAI Conference on Artificial Intelligence*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph attention networks.

Pat Verga, Haitian Sun, Livio Baldini Soares, and William W. Cohen. 2020. Facts as experts: Adaptable and interpretable neural memory over symbolic knowledge. *ArXiv*, abs/2007.00849.

Jiyue Wang, Pei Zhang, Qianhua He, Yanxiong Li, and Yongjian Hu. 2021. Revisiting label smoothing regularization with knowledge distillation. *Applied Sciences*, 11(10).

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable zero-shot entity linking with dense entity retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6397–6407, Online. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Semih Yavuz, Kazuma Hashimoto, Yingbo Zhou, Nitish Shirish Keskar, and Caiming Xiong. 2022. Modeling multi-hop question answering as single sequence prediction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 974–990, Dublin, Ireland. Association for Computational Linguistics.

Xi Ye, Rohan Nair, and Greg Durrett. 2021. Connecting attributions and QA model behavior on realistic counterfactuals. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5496–5512, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Donghan Yu, Chenguang Zhu, Yuwei Fang, Wenhao Yu, Shuohang Wang, Yichong Xu, Xiang Ren, Yiming Yang, and Michael Zeng. 2022. KG-FiD: Infusing knowledge graph in fusion-in-decoder for open-domain question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4961–4974, Dublin, Ireland. Association for Computational Linguistics.

Dataset	Train	Validation	Test
HotpotQA - distractor	90,447	7,405	7,405
Musqiue - Answerable	19,938	2,417	2,459

Table 4: Number of samples in each data split for HOTPOT-QA and MUSIQUE.

A Breakdown of Performance by Question Type - HOTPOT-QA

Model	Bridge	Comparison
FID-Base	60.8	65.97
PATH-FID-Base	61.19	65.37
SEQGRAPH-Base	63.6	66.51
PATH-FID-Large	63.72	71.68
SEQGRAPH-Large	65.21	71.69

Table 5: Performance breakdown of Answer-EM by question type on dev set of HOTPOT-QA

B Reasoning Path variants in MUSIQUE

HotpotQA	
Question:	What is the name of the executive producer of the film that has a score composed by Jerry Goldsmith?
Answer:	Ronald Shusett
Reasoning Path:	[title-1] Alien (soundtrack) [facts-1] [f1] [title-2] Alien (film) [facts-2] [f6] [answer] Ronald Shusett
Musique	
Question:	Who is the spouse of the Green performer?
Answer:	Miquette Giraudy
Reasoning Path:	
DA:	[question-1] Who is the performer of Green? [question-2] Who is the Spouse of #1? [answer] Miquette Giraudy
SA:	[title-1] Green (Steve Hillage album) [title-2] Miquette Giraud [answer] Miquette Giraudy
SIA:	[title-1] Green (Steve Hillage album) [answer-1] Steve Hillage [title-2] Miquette Giraudy [answer] Miquette Giraudy
DSIA:	[question-1] Who is the performer of Green? [title-1] Green (Steve Hillage album) [answer-1] Steve Hillage [question-2] Who is the Spouse of #1? [title-2] Miquette Giraudy [answer] Miquette Giraudy

Table 6: Reasoning path variants for HOTPOT-QA and MUSIQUE. Relevant passage titles are marked in blue, supporting facts in orange, intermediate answer/final answer is marked in green and the decomposed questions are marked in brown

The different reasoning path variants that can be constructed based on ground truth can be found in Table 6. Results of training baselines on these different variants can be found in Table 7

C Performance by Number of Hops - Graphs

We hypothesize that the answer F1 of SEQGRAPH on questions with ≥ 4 hops gets impacted due to presence of shortcuts since the support F1 score is higher than PATH-FID.

D Comparison of Musique-Answerable test F1 scores

Table 8 shows the comparison of our models with the current best performing ones on the MUSIQUE-Answerable test set leaderboard. Our End-to-End single stage model SEQGRAPH-large trained to output title + intermediate answers (SIA) outperforms the Longformer-Large(Beltagy et al., 2020) End-to-End model by 17 points in answer F1 and by 9-points in support F1. Furthermore, we also outperform the current state-of-the-art SA model which is a two stage model (Roberta Large(Liu et al., 2019) + Longformer Large) by 5 points on Answer F1 and 3 points on Support F1.

Model	Answer-EM	Answer-F1	Support-EM	Support-F1
SA	32.02	41.76	47.04	76.23
DA*	31.61	41.4	XX	XX
SIA	34.71	44.93	57.3	80.18
DSIA	33.35	43.08	53.5	78.79

Table 7: Results on different variants of MUSIQUE reasoning paths. *Since DA does not predict a reasoning path with titles, we do not compute the Support EM and F1.

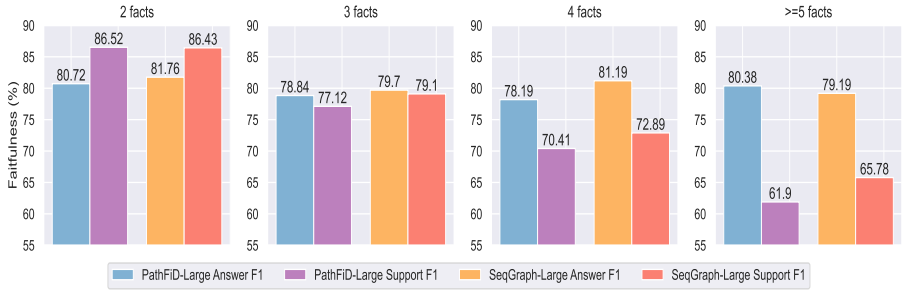

Figure 5: Performance on dev set of HOTPOT-QA decomposed by number of supporting facts.

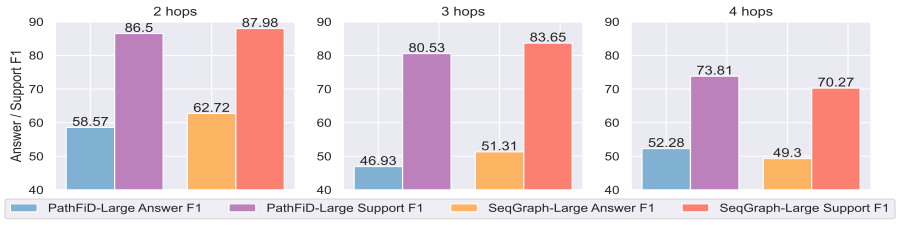

Figure 6: Performance on dev set of MUSIQUE decomposed by number of hops.

Model	**Answer F1**	**Support F1**
Select+Answer (SA) Model	52.3	75.2
Step Execution by Select+Answer (EX(SA)) Model	49	**80.6**
Step Execution by End2End (EX(EE)) Model	46.4	78.1
End2End (EE) Model	40.7	69.4
FID-Large	48.4	XX
PATH-FID-SIA-Large	54.8	77.9
SEQGRAPH-SIA-Large	**57.6**	78.4

Table 8: Current best performing models on the leaderboard (Longformer-Large variants vs our baselines vs SEQGRAPH

parameter	FID-LARGE	PATH-FID-LARGE
initialization	t5-large	t5-large
learning rate	1e-4	1e-4
learning rate schedule	linear	linear
effective batch size	64	64
gradient checkpointing	yes	yes
maximum input length	256	256
maximum output length	32	64
warmup steps	2000	2000
gradient clipping norm	1.0	1.0
training steps	16000	16000
weight decay	0.01	0.01
optimizer	adamw	adamw

Table 9: Hyperparameters for experiments on HotpotQA Distractor setting.

parameter	FID-LARGE	PATH-FID-LARGE-SIA
initialization	t5-large	t5-large
learning rate	1e-4	1e-4
learning rate schedule	linear	linear
effective batch size	64	64
gradient checkpointing	yes	yes
maximum input length	256	256
maximum output length	32	90
warmup steps	1000	1000
gradient clipping norm	1.0	1.0
training steps	6500	6500
weight decay	0.01	0.01
optimizer	adamw	adamw

Table 10: Hyperparameters for experiments on Musique-Answerable setting.

parameter	SEQGRAPH-LARGE
GNN	GAT(Veličković et al., 2017)
GNN Hidden Dimension	1024
GNN Number of layers	2
GNN dropout	0.2
Number of heads	8
Layer for fusion L	3

Table 11: Additional Graph related hyperparameters for SeqGraph

E Hyperparameter Settings

Tables 9, 10, 11 detail the hyperparameters we use for FID,PATH-FID and SEQGRAPH for HOTPOT-QA and MUSIQUE.

The 2-layer GNN module is 17M parameters for the large model and 9.5M for the base, accounting for only upto 4% increase in model parameters.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Left blank.

☐ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Left blank.

☑ A4. Have you used AI writing assistants when working on this paper?
General paraphrasing of content

B ☑ Did you use or create scientific artifacts?

4

☑ B1. Did you cite the creators of artifacts you used?
4

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Left blank.

C ☑ Did you run computational experiments?

5

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
5

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Appendix E

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Left blank.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Left blank.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.