R^2-CoD: Understanding Text-Graph Complementarity in Relational Reasoning via Knowledge Co-Distillation

Anonymous ACL submission

Abstract

Relational reasoning lies at the core of many NLP tasks, drawing on complementary signals from text and graphs. While prior research has investigated how to leverage this dual complementarity, a detailed and systematic understanding of text-graph interplay and its effect on hybrid models remains underexplored. We take an analysis-driven approach to investigate text–graph representation complementarity via a unified architecture that supports knowledge co-distillation (CoD). We explore five tasks involving relational reasoning that differ in how text and graph structures encode the information needed to solve that task. By tracking how these dual representations evolve during training, we uncover interpretable patterns of alignment and divergence, and provide insights into when and why their integration is beneficial.

1 Introduction

Incorporating modalities beyond the surface form of the text has shown promise for several challenging natural language processing (NLP) tasks. This is particularly true for **relational reasoning** based tasks where the objective is to understand or infer the semantic relationships within the input (Nastase et al., 2015). Examples of such tasks are relation extraction (Zhang et al., 2018b; Christopoulou et al., 2019; Guo et al., 2020), knowledge base question answering (KBQA) (Tian et al., 2024; Feng and He, 2025; Gao et al., 2025), and structured document interpretation or reasoning (Yao et al., 2018; Wang et al., 2023; Chen et al., 2025).

A common and effective way to encode relational structure is through graphs (Yao et al., 2018; Lee et al., 2023; Lin et al., 2025; Gururaja et al., 2023; Dutt et al., 2022), where nodes represent textual units and edges encode relationships, like semantic links or ontological structure. This explicit representation of structured information enables models to leverage signals that are complementary to or explicitly absent from the text.

While many tasks utilize this text-graph representation to improve performance, how they complement each other remains underexplored. Some systematic reviews (Stanton et al., 2021) observe that models fail to effectively integrate data from distinct modalities. This raises important open questions: How do text and graph representations relate to each other during learning? Do they converge toward similar representations, or diverge to encode distinct signals? And under what conditions is their integration most beneficial?

To address these questions, we adopt an analysis-oriented approach and introduce a unified framework for characterizing the alignment and complementarity between text and graph representations across tasks. We inspect how these dual representations relate and evolve with knowledge co-distillation (CoD) (Yao et al., 2024), an architectural framework that can generalize across a range of tasks where both text and graph inputs are available. We conduct this analysis across a diverse suite of five tasks involving relational reasoning spanning fine-grained, localized reasoning between entity pairs to multi-entity inference. To this end:

- We systematically analyze how text and graph representations complement each other under knowledge co-distillation (CoD) across five relational reasoning tasks.
- We identify consistent patterns ranging from complementarity to alignment and characterize how these patterns differ across tasks.
- We provide practical insights to inform the effective use of CoD.

2 Related work

Text–graph integration in NLP: Graphs have long played an important role in NLP, traditionally used to capture structure in tasks such as syntactic parsing, information retrieval, text min-

ing, and encode semantic representation through knowledge graphs, linguistic frameworks, and other semantic networks. Graph Neural Network (GNN)s (Scarselli et al., 2009) and their variants such as Graph Convolutional Neural Networks (GCNs) (Bruna et al., 2014) and Graph Attention (GAT) layers (Veličković et al., 2018) have become the de-facto way to integrate text and graph representations across a variety of tasks.

In text classification, graphs have been used to jointly model word and document relations (Yao et al., 2018) and to enhance transformers with structured information (Lin et al., 2021). Knowledge graphs provide support for reasoning and information retrieval for QA (Sun et al., 2018; Yasunaga et al., 2022; Lin et al., 2025). For document understanding, graph-based methods have been applied to paragraph recognition (Wang et al., 2022; Liu et al., 2022b), information extraction (Lee et al., 2023), and layout or structure analysis (Wang et al., 2023; Chen et al., 2025). More recently, such approaches have also been used to detect AI-generated content (Valdez-Valenzuela et al., 2025).

While text–graph integration has been widely used for performance gains, little is known about how their representations relate during learning. We analyze this relationship and how it is shaped by task characteristics and learning objectives.

Knowledge distillation (KD): One of the earliest works in this space was of Buciluǎ et al. (2006), i.e. a kind of model compression to facilitate efficient ensembling of complex classifiers. Hinton et al. (2015) refined it to distill knowledge from one model to another. Later, this type of directed, teacher-student knowledge distillation (KD) has seen usage in several NLP tasks (Sanh et al., 2019; Sun et al., 2019; Liang et al., 2020; Liu et al., 2022a). As opposed to distilling information from one model to another, Zhang et al. (2018a) proposed the idea of mutual learning where information is shared between models. Finally, Tian et al. (2020) introduced contrastive representational distillation, which later works (Sun et al., 2020; Fu et al., 2021) showed is effective at refining KD-loss for shared representational spaces.

Though KD is widely prevalent in NLP, its effectiveness in successfully compressing complex tasks remains unclear. Stanton et al. (2021) argues that a gap exists in our current understanding of KD, evident in the difficulty in obtaining model fidelity for certain types of teachers. Though it is known that KD's efficacy varies across models, the reason remains unknown.

Representation analysis: Representation analysis examines the internal representations learned by models to better understand how they encode and process information. Subsequently, a variety of tools have been developed for this purpose. These range from traditional methods such as Principal Component Analysis (PCA) (Ferrone and Zanzotto, 2020) and Canonical Correlation Analysis (CCA) for dimensionality reduction and visualization, to more targeted approaches such as classifier probes to test whether specific linguistic properties are encoded in model representations (Belinkov, 2021; Gupta et al., 2015). Recently, sparse autoencoders (Gao et al., 2024; Cunningham et al., 2023; Ng et al., 2011) have also been deployed for extracting interpretable features from model representations.

To support our goal of analyzing how text and graph representations are related during learning, we require lightweight, task-agnostic tools to enable consistent and interpretable comparisons across tasks. We thus adopt PCA and leverage distance-based metrics to answer our questions.

3 Task suite and formulations

We propose a spectrum of how the relationship between the text and the graph representations can vary as visualized in **Figure 1**. This spectrum ranges from cases where text and graph encode largely complementary information and preserve distinct representations (left), to cases where they tend to converge and form aligned representations (right). In between, partial alignment refers to the case where the representations become more similar but do not fully converge. To cover this spectrum, we select five relational reasoning tasks with diverse characteristics, such as 1) how explicitly the graph models the relation or structure that the task seeks to predict, 2) whether nodes have direct correspondence to textual spans, and 3) the scope of reasoning (e.g., local mention pairs versus global graph structure). This diversity enables us to examine how these variations shape text-graph representation relations. We outline the goal, input and output, an illustrated example, graph construction method, and the knowledge type of each task in Table 1.

Event temporal relation extraction (ETRE) The objective of ETRE is to predict the temporal

Task	Goal	Input	Output	Example	K-Type
ETRE	Predict temporal relation between two events	Text passage + Syntactic graph and Time-aware graph	Relation label (e.g., BEFORE/AFTER)	In: Atlanta nineteen ninety-six. A bomb <E1> blast </E1> shocks the Olympic games. One person is killed. January nineteen ninety-seven. Atlanta again. This time a bomb at an abortion clinic. More people are <E2> hurt </E2>. Out: Event E1 took place BEFORE Event E2.	Episodic
MLRE	Predict semantic relation between entities	Text passage + Dependency graph	Relation label (e.g. sibling)	In: The <E1> wood </E1> is used as fuel and to make posts for <E2> fences </E2>. Out: The relation between E1 and E2: material used	Episodic
FU	Predict token relationships in scanned forms	OCR tokens with layout info	Label over token pairs	We present an example in Figure 7	Episodic
RPP	Predict reasoning path over the KG for a question.	Question + KG subgraph	Reasoning Path	In: Question: What was Elie Wiesel's father's name? KG: `Elie Wiesel <E1> | <E1> book.author.book_editions_published <E2> | <E3> people.person.gender <E4> ...` Out: Reasoning Pattern Type: T2 — The answer is located a single-hop away from the two constraints. Entities ranked: `<E6>, <E4>, ...`	Static
KBQA entity-ranking	Extract answers from a KG for a question		Ranked list of candidate entities		

Table 1: For each task, we state the goal, the input/output format, an illustrative example, and the graph construction method. We also distinguish between tasks grounded in *episodic* knowledge (context-dependent and document-specific), and those involving *static* knowledge (holds independently of context) in the Knowledge(K)-Type column.

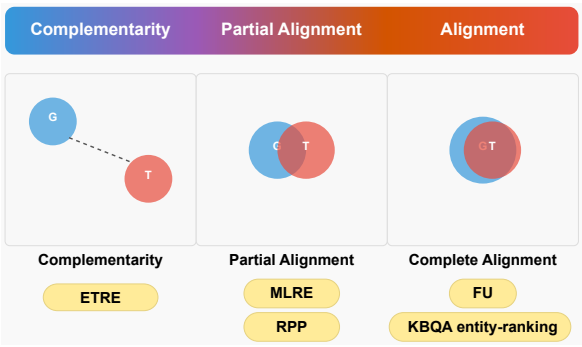


Figure 1: Task spectrum of representation relationships. Left: they remain distinct and complementary. Middle: they show some similarity but do not fully align. Right: they converge toward aligned representations. This spectrum motivates our task selection for analysis.

relationship y_{ij} between a pair of event mentions (i, j) within a short passage q using a fixed relation label set (e.g., *before*, *after*, *simultaneous*, *vague*). Because distinct layers of text encode time cues for short- and long-distance mention pairs, the model represents the text using linear transformers alongside associated graph $G(V, E)$. The graph contains nodes V for event mentions and time expressions, and edges E encoding structural relations. It is derived from the q using part-of-speech labeling and by applying temporal logic to limited date-time associations which can be extracted from q. Thus, while the graph does not explicitly encode the temporal relation being predicted or have direct correspondence with text spans it reflects long-distance structural dependencies not captured by linear transforms. We follow Yao et al. (2024) and use three benchmark datasets: TimeBank-Dense (TB-Dense) (Cassidy et al., 2014), TDDiscourse-Auto (TDDAuto) and TDDiscourse-Manual (TDDMan) (Naik et al., 2019).

Multilingual relation extraction (MLRE) In a similar vein, the task of MLRE involves identifying the semantic relation between a pair of entity mentions within a given sentence(s) q for a particular language. Each text input has its corresponding graph $G(V, E)$ generated by an off-the-shelf dependency parser (Qi et al., 2020) where V represents the words in the sentence(s) and the E represents the syntactic dependencies between the words. We initialize the nodes(words) in the graph by pooling across its constituent token embeddings, and further augment it with the structural information obtained from the graph's topology via Walklets (Perozzi et al., 2017). We emphasize that the dependency relations capture the explicit linguistic signals between words but do not encode the relation being predicted. We provide an example in the Appendix A.2. We use the RED^{fm} (Huguet Cabot et al., 2023) dataset which covers five languages.

Reasoning pattern prediction (RPP) Given a question q and its associated subgraph $G = (V, E)$ from the knowledge base, the goal is to infer the reasoning pattern or RP of the question q. Each pattern corresponds to a particular reasoning path, composed of single/multiple hops and single/multiple constraints. We provide detailed descriptions in Appendix A.1. The text input includes q and a linearized serialization of the subgraph G^{linear}, while the graph input uses the same question paired with the explicit graph structure G. Thus, both text and graph encode the same information but in structurally distinct forms. We initialize the nodes in each subgraph with Walklets embeddings following the same procedure as Dutt et al. (2022). RP prediction requires reasoning over the entire graph with respect to the question, rather than individual tokens or nodes. We use WebQSP dataset for our task (Yih et al., 2016; Xie et al., 2022).

KBQA entity-ranking We formulate extracting the correct answer(s) for a given question from its associated subgraph as a ranking problem. The model operates over a shared set of candidate entities and assigns a relevance score to each entity based on its likelihood of being the correct answer. The input is the same as in the reasoning pattern prediction task setting. To enable entity-level predictions from the text model, we extract an embedding for each candidate entity by identifying its corresponding span in the text and aggregating the token representations produced by the text encoder. Each entity thus has a one-to-one correspondence: it appears as a node $v_i \in V$ in the graph and as a token span $s_i \subseteq q$ in the text.

Form understanding (FU): This task involves identifying key–value relationships between textual spans extracted from scanned forms, such as "Date: 2024-12-01". Each input document is processed by OCR to yield textual tokens with bounding-box coordinates. The corresponding graph $G(V, E)$ encodes the visual layout of the document; V represents OCR tokens and E captures spatial relations between the tokens such as alignment, proximity, and reading order. Such a framework encodes positional cues central to the task objective, and establishes a one-to-one correspondence between the nodes and the OCR tokens. We adopt the experimental setup of Nourbakhsh et al. (2024) and include three multimodal datasets, i.e. SROIE (Huang et al., 2019), FUNSD (Jaume et al., 2019), and CORD (Park et al., 2019).

4 Unified framework for analysis

We propose a unified, task-agnostic framework, henceforth called R^2-CoD (**Figure 2**), to understand how text and graph representations relate during learning. We choose a framework that is generalizable to observe how information from text and graph are represented and integrated.

Across tasks, each instance corresponds to a text-graph pair, as defined in Section 3. These are encoded using modality-specific encoders: $h_{\text{t}} = f_{\text{t}}(q)$ and $h_{\text{g}} = f_{\text{g}}(G)$. We then create a hybrid representation h_{hybrid} through concatenation or residual connection to perform task-specific prediction and compute the task loss:

$$h_{\text{hybrid}} = f_{\text{fuse}}(h_{\text{t}}, h_{\text{g}}) \quad (1)$$

$$\mathcal{L}_{\text{task}} = \mathcal{L}(h_{\text{hybrid}}, y) \quad (2)$$

where y denotes the gold supervision and $\mathcal{L}(\cdot, \cdot)$ is the task-specific loss function. We present model configurations, loss function, and evaluation metrics used for each task in Table 5 in the Appendix.

To analyze text and graph representations, we require a shared space where they can be directly compared. Thus, we apply modality-specific MLP projection heads that learn to map each representation into a shared latent space during training:

$$z_{\text{text}} = \text{MLP}_{\text{t}}(h_{\text{t}}), z_{\text{graph}} = \text{MLP}_{\text{g}}(h_{\text{g}}). \quad (3)$$

4.1 Contrastive co-distillation

While learning a shared space enables comparison, it cannot solely influence how text and graph will complement one another. We thus apply a contrastive knowledge co-distillation (CoD) objective (Yao et al., 2024) which combines a contrastive loss with a stop-gradient operation (Chen and He, 2021) to explicitly encourage bidirectional knowledge transfer. Such a formulation allows us to observe how the information encoded in one modality influences the other during mutual learning.

Formally, the contrastive loss l_{cl} between the teacher t and the student s representations is:

$$l_{\text{cl}}(t, s) = -\log \frac{e^{sim(t,s)/\tau}}{\sum_u \mathbb{1}_{[u \neq t]}\, e^{sim(t,u)/\tau}} \quad (4)$$

where u indicates representations from the training data other than t and s, $sim(.,.)$ is cosine similarity, τ is the temperature scaling parameter (Tian et al., 2022). Note that the notions of "teacher" and

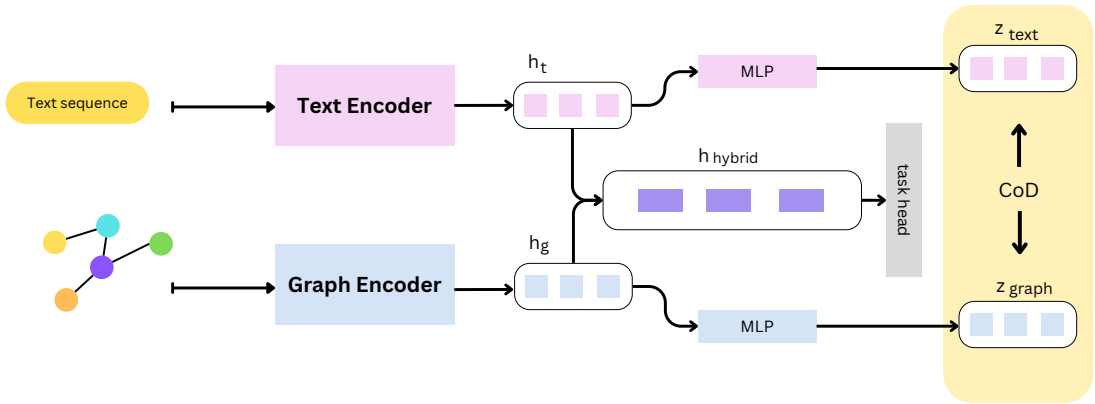


Figure 2: Our unified framework for analyzing how text and graph representations complement each other. A text sequence and its corresponding graph are processed by separate encoders. Their outputs are used in two ways: (1) combined as hybrid inputs for task prediction, and (2) projected into a shared space where a contrastive co-distillation (CoD) objective encourages mutual learning and enables representation-level analysis.

"student" are interchangeable and fully symmetric: one scenario treats the text projection behaves as the teacher supervising the graph projection, while in another the graph projection supervises the text projection. This bidirectional design ensures that either modality can act as teacher or student at each step, thus mutually distilling knowledge from each other. Hence, the full CoD loss is computed as

$$\mathcal{L}_{\text{CoD}} = \frac{1}{2}\sum_i [l_{\text{cl}}(z_i^{\text{text}}, \hat{z}_i^{\text{graph}}) + l_{\text{cl}}(z_i^{\text{graph}}, \hat{z}_i^{\text{text}})] \tag{5}$$

where $\hat{\cdot}$ is the stop gradient operator (Chen and He, 2021) that sets the input variable to a constant. Finally, we combine this with the task loss to enable end-to-end model optimization:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda \mathcal{L}_{\text{CoD}} \tag{6}$$

where λ controls the weight of the CoD signal. CoD serves as a task-agnostic framework to facilitates learning and analysis over dual modalities.

4.2 Measuring representation relations

To evaluate how text and graph relate during learning, we need tools that can surface both alignment and divergence in the shared space. Our goal is to characterize the degree to which text and graph converge, remain distinct, or shift in their relationship throughout training under CoD.

To support visual interpretation, we apply PCA, which reduces the projected embeddings $(z_{\text{text}}, z_{\text{graph}})$ into a two dimensional space and reveals their spatial arrangement at various stages of training, i.e. whether there is clustering, separation, or overlap between modalities.

For more precise measurement, we also compute batch-level cosine similarity between paired representations, along with average within- and between-modality distances based on cosine distance. Together, these measures capture both the directional and spatial properties of different modalities in the learned representation space.

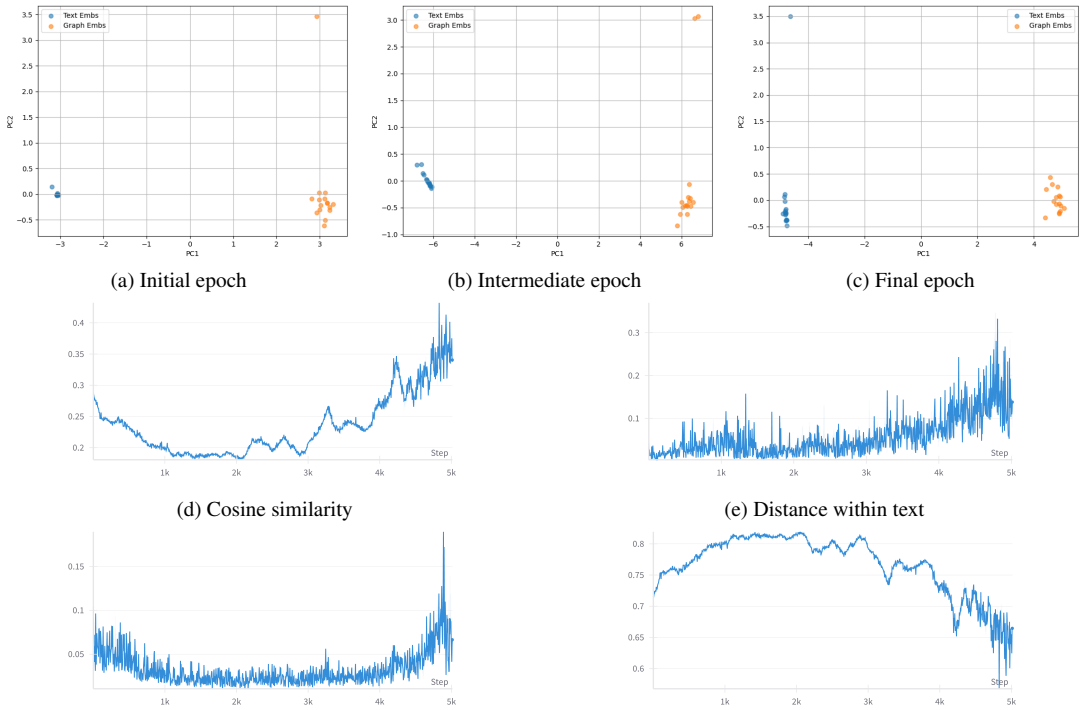
5 Analysis and discussions

5.1 RQ1: Does combining text and graph representations improve performance?

We examine whether integrating textual and graph-based representations improves task performance, and whether CoD facilitates more effective integration. We compare four model configurations: (1) text-only, (2) graph-only, (3) hybrid without CoD, and (4) hybrid with CoD. We present the main results in Table 2. The task suite statistics and training times in Table 7 illustrate that CoD introduces minimal additional cost. Additional results for different model combinations in Table 8 further demonstrate the generalizability of our CoD framework across different text and graph models.[1]

Across the tasks, we observe that hybrid models consistently outperform the text-only and graph-only baselines, and that incorporating the CoD loss leads to further gains. The only exception is for MLRE where the hybrid approaches achieve performance comparable to the the text-based baseline, possibly because the graph representations fail to capture any complementary signals. Prior work has

[1]For consistency, we report results either from our own experiments or from existing work when the same architecture is adopted.



(a) Initial epoch (b) Intermediate epoch (c) Final epoch

(d) Cosine similarity (e) Distance within text

(f) Distance within graph (g) Distance between text and graph

Figure 3: Results for ETRE on the TDDMan dataset. PCA visualizations (top) at initial, intermediate, and final training stages, and corresponding distance-based metrics (bottom).

Task	Dataset	Text	Graph	CoD	T+G
ETRE	TDDAuto	61.6	34.6	**77.1**	<u>68.9</u>
FU	FUNSD	33	22	**38**	<u>35</u>
MLRE	RED^{fm}	**79.7**	48.6	78.6	<u>79.5</u>
RPP	WebQSP	62.4	63.2	**65.9**	<u>65.6</u>
KBQA	WebQSP	80.7	52.2	**83.8**	<u>83.5</u>

Table 2: Task performance (averaged across three seeds) for text-only (Text), graph-only (Graph), hybrid with CoD (CoD), and hybrid without CoD, i.e. with only the text and graph representations (T+G). Best performance in **bold**, second-best <u>underlined</u>. We present results for one representative dataset per task due to resource constraints. Similar trends hold for other datasets. For FU, the model was pretrained on a 1,000-example subset of its original pretraining corpus.

demonstrated how large-scale pretraining enables transformer models to encode syntactic information within their parameters (Starace et al., 2023; Liu et al., 2024) and thus employing off-the-shelf parsers to capture dependency information shows little promise (Sachan et al., 2021).

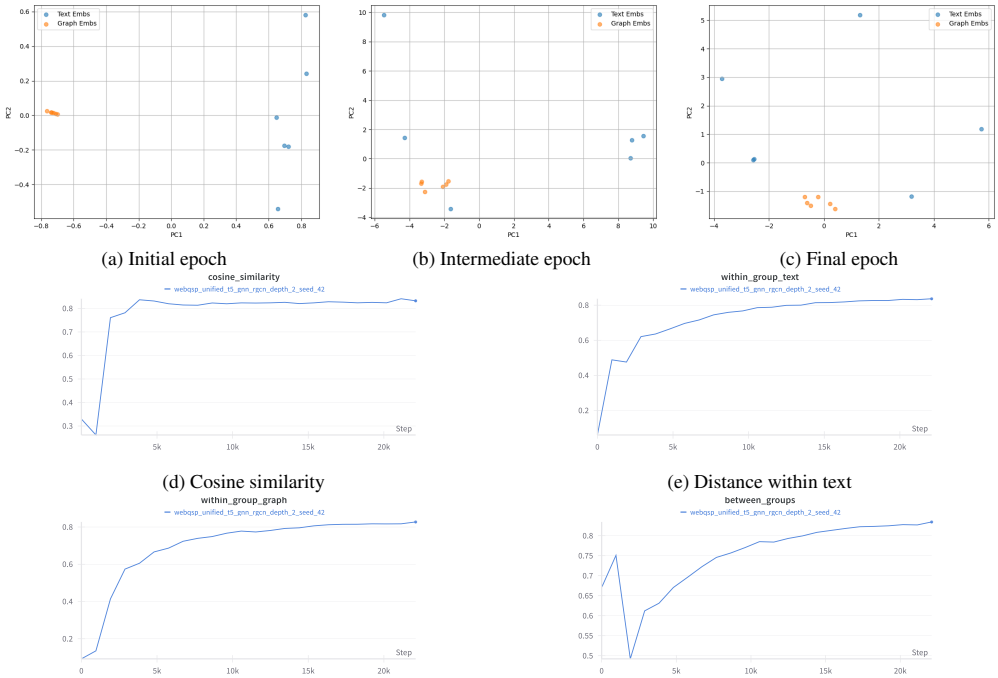
For the KBQA tasks, where text and graph inputs aim to encode the same information albeit coming from two different formats, i.e. the linearized format (for the text) versus the topological structure (for the graph), CoD offers only marginal gains over the default hybrid setting. In contrast, tasks like FU where text and graph encode different information (form content versus layout structure from OCR), CoD shows more improvement.

5.2 RQ2: How do text and graph representations relate during learning?

Using the representation analysis framework detailed in Section 4, we observe three qualitatively distinct trends in the spatial relationship between text and graph representations that aligns with the task spectrum proposed in Figure 1: complementarity, partial alignment, and complete alignment.

Complementarity (ETRE): The text and graph representations remain well-separated throughout training, signifying that they contribute distinct, complementary signals rather than converging towards a shared embedding space.

The PCA visualization confirms this complementarity. In Figure 3, text and graph representations consistently occupy distinct regions. We attribute this separation to distinctiveness in how text and graph encode task-relevant information. In ETRE, the text representation provides local semantic cues around event mentions, while the graph encodes



(a) Initial epoch (b) Intermediate epoch (c) Final epoch

(d) Cosine similarity (e) Distance within text

(f) Distance within graph (g) Distance between text and graph

Figure 4: Results for reasoning pattern prediction on the WebQSP dataset.

structural information in an attempt to quantify semantic temporal and discourse relations. These structural and semantic divergences could lead text and graph representations to retain independent representation space.

Partial alignment (MLRE and RPP): We observe that MLRE and RPP exhibit moderate convergence between text and graph. PCA visualizations (Figure 11 and 4) show that the text and graph representations move closer in the shared space during training, yet remain largely separable. This suggests that text and graph are aligning but do not collapse into a single unified cluster. This behavior aligns with the task objective: while the inputs encode equivalent information, the objective is to classify the reasoning path traversed in the graph, not specific tokens or nodes. Thus, the text and graph representations can evolve in parallel without needing to fully align. This allows each of them to retain its inductive biases while adapting to shared learning signals through CoD.

Complete alignment (FU and KBQA): FU and KBQA appear near the alignment end of our spectrum. In both tasks, text and graph representations show strong convergence. E.g., PCA visualizations in Figure 5 show a clear alignment trajectory: initial representations are moderately separated in the shared space, but progressively draw closer during training. By the final epochs, the paired embeddings often form overlapping clusters.

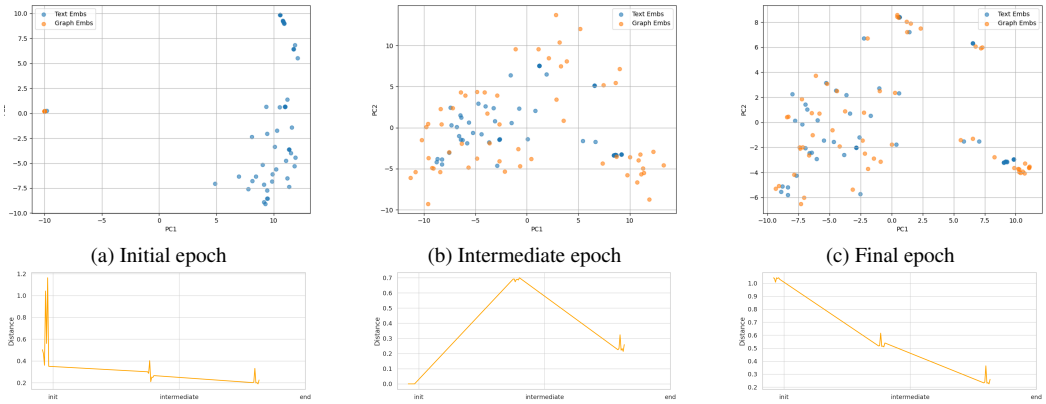
We explain this finding by establishing the fine-grained correspondence in input structure between text and graph for both tasks. Each graph node $v_i \in V$ has a clear textual counterpart as a token span $s_i \subseteq q$. In FU, OCR tokens are linked to spatially grounded nodes, while in entity ranking, candidate answer entities are matched between graph nodes and text tokens. This one-to-one correspondence likely encourages representations to align.

To complement the PCA-based categorization, we further analyze cosine similarity and distance metrics to offer quantitative insight into the degree of alignment.

Cosine similarity increases consistently across tasks due to the CoD objective, which encourages directional agreement between text and graph representations. However, in contrast to other tasks, tasks characterized by complementarity like ETRE exhibits a weaker increase, whose cosine similarity remains bounded to 0.4.

Alignment is indicated by how closely between-group distances match within-group ones. In ETRE, the between-group distance remains consistently higher than the within-group distances. In

[2]We present distance metrics for three representative training phases due to resource constraints.



(a) Initial epoch (b) Intermediate epoch (c) Final epoch

(d) Distance within text (e) Distance within graph (f) Distance between text and graph

Figure 5: Results for form understanding on the CORD dataset.[2]

both partial and complete alignment tasks, within-group distances increase over training. However, their between-group trends diverge: in partial alignment tasks like MLRE and RPP, the between-group distance also increases, whereas in complete alignment tasks such as FU and KBQA, the between-group distance steadily decreases and eventually approaches the within-group distances.

5.3 RQ3: How do task characteristics shape the effects of CoD?

Building on the representation patterns observed in RQ2, we now examine what task-specific characteristics may shape how CoD influences learning. We only consider the cases where adding in CoD consistently brought about performance gains.

Same input, different task objectives Although RPP and KBQA share the identical input, they differ in task objectives: the former identifies reasoning patterns in the subgraph on a global level, whereas the latter scores individual entities at a local level. Despite this shared input, the learned representations behave differently under CoD. RPP demonstrates partial convergence, while KBQA shows strong alignment. This contrast suggests that the level at which reasoning is required, in this case global vs. local, can shape how the representations align in the representation space.

Same reasoning scope, different graph construction: ETRE and FU both involve localized reasoning between pairs: event mentions in ETRE and field spans in forms. However, their graph constructions differ in how directly they support the task. In FU, edges explicitly capture spatial layout relations that closely match the key–value associations being predicted. In ETRE, the graph encodes distinct layers of linguistic cues (e.g., syntax, discourse), which support but do not directly define the target temporal relation. Under CoD, FU shows complete alignment while ETRE demonstrates complementarity. This indicates that how well the graph structure reflects the task objective can influence whether CoD promotes complementarity or alignment.

With or without token-node correspondence In FU and KBQA entity-ranking, there is a strong one-to-one correspondence between graph nodes and text token spans. This provides a scaffold that supports representational convergence, which is reinforced through CoD. This is in contrast to complementary information encoded in ETRE, where representations remain more distinct, and CoD preserves separation. This highlights that explicit token–node correspondence could act as a structural prior that facilitates CoD towards alignment.

6 Conclusion

We analyze how text and graph representations complement each other during learning within a unified, task-agnostic framework using contrastive co-distillation (CoD) as a lens. We select five diverse relational reasoning tasks and observe a spectrum of representational behaviors from alignment to complementarity shaped by differences in task structure, such as whether the graph encodes the prediction target explicitly, whether nodes correspond directly to textual spans, and whether reasoning operates at a local or global level. These findings improve our understanding of text-graph representation relations and offer practical insights into applying CoD in structured NLP tasks.

7 Limitations

Task coverage While our selected five relational reasoning tasks covers a broad spectrum of complementarity and alignment patterns, extending the framework to other tasks may reveal additional representational behaviors.

Analysis metrics We rely on PCA visualizations and cosine/distance-based metrics for representation analysis. These methods provide interpretable trends but may not capture all fine-grained or non-linear interactions between text and graph, which could be explored with advanced probing or disentanglement techniques.

8 Ethical considerations

Bias propagation Our framework builds on pre-trained text and graph encoders, which may inherit and amplify biases present in the underlying data sources.

References

Yonatan Belinkov. 2021. Probing classifiers: Promises, shortcomings, and advances. *Preprint*, arXiv:2102.12452.

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2014. Spectral networks and locally connected networks on graphs. *Preprint*, arXiv:1312.6203.

Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Misil. 2006. Model compression. In *In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541.

Dan Busbridge, Dane Sherburn, Pietro Cavallo, and Nils Y. Hammerla. 2019. Relational graph attention networks. *Preprint*, arXiv:1904.05811.

Taylor Cassidy, Bill McDowell, Nathanael Chambers, and Steven Bethard. 2014. An annotation framework for dense event ordering. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 501–506.

Xinlei Chen and Kaiming He. 2021. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15750–15758.

Yufan Chen, Ruiping Liu, Junwei Zheng, Di Wen, Kunyu Peng, Jiaming Zhang, and Rainer Stiefelhagen. 2025. Graph-based document structure analysis. *Preprint*, arXiv:2502.02501.

Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2019. Connecting the dots: Document-level neural relation extraction with edge-oriented graphs. *Preprint*, arXiv:1909.00228.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2023. Sparse autoencoders find highly interpretable features in language models. *Preprint*, arXiv:2309.08600.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Ritam Dutt, Kasturi Bhattacharjee, Rashmi Gangadharaiah, Dan Roth, and Carolyn Rose. 2022. Perkgqa: Question answering over personalized knowledge graphs. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 253–268.

Ritam Dutt, Sopan Khosla, Vinayshekhar Bannihatti Kumar, and Rashmi Gangadharaiah. 2023. GrailQA++: A challenging zero-shot benchmark for knowledge base question answering. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 897–909, Nusa Dua, Bali. Association for Computational Linguistics.

Tengfei Feng and Liang He. 2025. RGR-KBQA: Generating logical forms for question answering using knowledge-graph-enhanced large language model. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3057–3070, Abu Dhabi, UAE. Association for Computational Linguistics.

Lorenzo Ferrone and Fabio Massimo Zanzotto. 2020. Symbolic, distributed, and distributional representations for natural language processing in the era of deep learning: A survey. *Frontiers in Robotics and AI*, 6.

Hao Fu, Shaojun Zhou, Qihong Yang, Junjie Tang, Guiquan Liu, Kaikui Liu, and Xiaolong Li. 2021. Lrc-bert: latent-representation contrastive knowledge distillation for natural language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12830–12838.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. 2024. Scaling and evaluating sparse autoencoders. *Preprint*, arXiv:2406.04093.

Shengxiang Gao, Jey Han Lau, and Jianzhong Qi. 2025. Beyond seen data: Improving kbqa generalization through schema-guided logical form generation. *Preprint*, arXiv:2502.12737.

Yu Gu, Sue Kase, Michelle Vanni, Brian Sadler, Percy Liang, Xifeng Yan, and Yu Su. 2021. Beyond iid: three levels of generalization for question answering

on knowledge bases. In *Proceedings of the Web Conference 2021*, pages 3477–3488.

Zhijiang Guo, Yan Zhang, and Wei Lu. 2020. Attention guided graph convolutional networks for relation extraction. *Preprint*, arXiv:1906.07510.

Abhijeet Gupta, Gemma Boleda, Marco Baroni, and Sebastian Padó. 2015. Distributional vectors encode referential attributes. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 12–21, Lisbon, Portugal. Association for Computational Linguistics.

Sireesh Gururaja, Ritam Dutt, Tinglong Liao, and Carolyn Rose. 2023. Linguistic representations for fewer-shot relation extraction across domains. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7502–7514.

Geoffrey Hinton, Oriol Vinyals, Jeff Dean, and 1 others. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7).

Zheng Huang, Kai Chen, Jianhua He, Xiang Bai, Dimosthenis Karatzas, Shijian Lu, and CV Jawahar. 2019. Icdar2019 competition on scanned receipt ocr and information extraction. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1516–1520. IEEE.

Pere-Lluís Huguet Cabot, Simone Tedeschi, Axel-Cyrille Ngonga Ngomo, and Roberto Navigli. 2023. RED^{fm}: a filtered and multilingual relation extraction dataset. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4326–4343, Toronto, Canada. Association for Computational Linguistics.

Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. 2019. Funsd: A dataset for form understanding in noisy scanned documents. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 2, pages 1–6. IEEE.

Longquan Jiang and Ricardo Usbeck. 2022. Knowledge graph question answering datasets and their generalizability: Are they enough for future research? In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3209–3218.

Chen-Yu Lee, Chun-Liang Li, Hao Zhang, Timothy Dozat, Vincent Perot, Guolong Su, Xiang Zhang, Kihyuk Sohn, Nikolay Glushnev, Renshen Wang, Joshua Ainslie, Shangbang Long, Siyang Qin, Yasuhisa Fujii, Nan Hua, and Tomas Pfister. 2023. FormNetV2: Multimodal graph contrastive learning for form document information extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9011–9026, Toronto, Canada. Association for Computational Linguistics.

Kevin J Liang, Weituo Hao, Dinghan Shen, Yufan Zhou, Weizhu Chen, Changyou Chen, and Lawrence Carin. 2020. Mixkd: Towards efficient distillation of large-scale language models. In *International Conference on Learning Representations*.

Jiacheng Lin, Kun Qian, Haoyu Han, Nurendra Choudhary, Tianxin Wei, Zhongruo Wang, Sahika Genc, Edward W Huang, Sheng Wang, Karthik Subbian, Danai Koutra, and Jimeng Sun. 2025. Gt2vec: Large language models as multi-modal encoders for text and graph-structured data. *Preprint*, arXiv:2410.11235.

Yuxiao Lin, Yuxian Meng, Xiaofei Sun, Qinghong Han, Kun Kuang, Jiwei Li, and Fei Wu. 2021. BertGCN: Transductive text classification by combining GNN and BERT. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1456–1462, Online. Association for Computational Linguistics.

Chang Liu, Chongyang Tao, Jiazhan Feng, and Dongyan Zhao. 2022a. Multi-granularity structural knowledge distillation for language model compression. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1001–1011.

Shuang Liu, Renshen Wang, Michalis Raptis, and Yasuhisa Fujii. 2022b. Unified line and paragraph detection by graph convolutional networks. *Preprint*, arXiv:2203.09638.

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. *Preprint*, arXiv:1907.11692.

Zhu Liu, Cunliang Kong, Ying Liu, and Maosong Sun. 2024. Fantastic semantics and where to find them: Investigating which layers of generative LLMs reflect lexical semantics. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14551–14558, Bangkok, Thailand. Association for Computational Linguistics.

Aakanksha Naik, Luke Breitfeller, and Carolyn Rose. 2019. Tddiscourse: A dataset for discourse-level temporal ordering of events. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 239–249.

Vivi Nastase, Rada Mihalcea, and Dragomir R. Radav. 2015. A survey of graphs in natural language processing. *Natural Language Engineering*, 5:665–698.

Andrew Ng and 1 others. 2011. Sparse autoencoder. *CS294A Lecture notes*, 72(2011):1–19.

Armineh Nourbakhsh, Zhao Jin, Siddharth Parekh, Sameena Shah, and Carolyn Rose. 2024. AliGATr: Graph-based layout generation for form understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13309–13328, Miami, Florida, USA. Association for Computational Linguistics.

Seunghyun Park, Seung Shin, Bado Lee, Junyeop Lee, Jaeheung Surh, Minjoon Seo, and Hwalsuk Lee. 2019. Cord: a consolidated receipt dataset for post-ocr parsing. In *Workshop on Document Intelligence at NeurIPS 2019*.

Bryan Perozzi, Vivek Kulkarni, Haochen Chen, and Steven Skiena. 2017. Don't walk, skip! online learning of multi-scale network embeddings. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, pages 258–265.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer. *Preprint*, arXiv:1910.10683.

Devendra Sachan, Yuhao Zhang, Peng Qi, and William L Hamilton. 2021. Do syntax trees help pre-trained transformers extract information? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2647–2661.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2017. Modeling relational data with graph convolutional networks. *Preprint*, arXiv:1703.06103.

Samuel Stanton, Pavel Izmailov, Polina Kirichenko, Alexander A. Alemi, and Andrew G. Wilson. 2021. Does knowledge distillation really work? *Advances in neural information processing systems*, 34:6906–6919.

Giulio Starace, Konstantinos Papakostas, Rochelle Choenni, Apostolos Panagiotopoulos, Matteo Rosati, Alina Leidinger, and Ekaterina Shutova. 2023. Probing LLMs for joint encoding of linguistic categories. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7158–7179, Singapore. Association for Computational Linguistics.

Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William W. Cohen. 2018. Open domain question answering using early fusion of knowledge bases and text. *Preprint*, arXiv:1809.00782.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for bert model compression. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4323–4332.

Siqi Sun, Zhe Gan, Yuwei Fang, Yu Cheng, Shuohang Wang, and Jingjing Liu. 2020. Contrastive distillation on intermediate representations for language model compression. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 498–508.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2020. Contrastive representation distillation. In *International Conference on Learning Representations*.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2022. Contrastive representation distillation. *Preprint*, arXiv:1910.10699.

Yuhang Tian, Dandan Song, Zhijing Wu, Changzhi Zhou, Hao Wang, Jun Yang, Jing Xu, Ruanmin Cao, and HaoYu Wang. 2024. Augmenting reasoning capabilities of LLMs with graph structures in knowledge base question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11967–11977, Miami, Florida, USA. Association for Computational Linguistics.

Andric Valdez-Valenzuela, Helena Gómez-Adorno, and Manuel Montes-y Gómez. 2025. Text graph neural networks for detecting AI-generated content. In *Proceedings of the 1stWorkshop on GenAI Content Detection (GenAIDetect)*, pages 134–139, Abu Dhabi, UAE. International Conference on Computational Linguistics.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. *Preprint*, arXiv:1710.10903.

Jilin Wang, Michael Krumdick, Baojia Tong, Hamima Halim, Maxim Sokolov, Vadym Barda, Delphine Vendryes, and Chris Tanner. 2023. A graphical approach to document layout analysis. *Preprint*, arXiv:2308.02051.

Renshen Wang, Yasuhisa Fujii, and Ashok C. Popat. 2022. Post-ocr paragraph recognition by graph convolutional networks. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2533–2542.

Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I. Wang, Victor Zhong, Bailin Wang, Chengzu Li, Connor Boyle, Ansong Ni, Ziyu Yao, Dragomir Radev, Caiming Xiong, Lingpeng Kong, and 4 others. 2022. Unified-SKG: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. In *Proceedings of the 2022 Conference on Empirical*

Methods in Natural Language Processing, pages 602–631, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Hao-Ren Yao, Luke Breitfeller, Aakanksha Naik, Chunxiao Zhou, and Carolyn Rose. 2024. Distilling multi-scale knowledge for event temporal relation extraction. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 2971–2980.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2018. Graph convolutional networks for text classification. *Preprint*, arXiv:1809.05679.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2022. Qa-gnn: Reasoning with language models and knowledge graphs for question answering. *Preprint*, arXiv:2104.06378.

Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–206, Berlin, Germany. Association for Computational Linguistics.

Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. 2018a. Deep mutual learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4320–4328.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018b. Graph convolution over pruned dependency trees improves relation extraction. *Preprint*, arXiv:1809.10185.

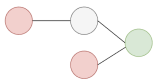
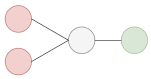
RP	Illustration	Definition	Example Question	S-expression
T-0		A single-hop path from the constraint to the answer.	What is the name of money in Brazil?	(JOIN (R location.country.currency_used) m.015fr)
T-1		A two-hop path from the constraint to the answer.	Where does the Queen of Denmark live?	(JOIN (R people.place_lived.location) (JOIN (R people.person.places_lived) m.0g2kv))
T-2		Two single-hop paths arising from two different constraints and converging to the same answer.	What was Elie Wiesel's father's name?	(AND (JOIN people.person.gender m.05zppz) (JOIN (R people.person.parents) m.02vsp))
T-3		Two paths (one single-hop and another two-hop) arising from two different constraints and converging to the same answer.	Where did Joe Namath attend college?	(AND (JOIN common.topic.notable_types m.01y2hnl) (JOIN (R education.education.institution) (JOIN (R people.person.education) m.01p_3k)))
T-4		Two two-hop paths arising from two different constraints and converging to an intermediate common node before reaching the answer.	Who does Zach Galifianakis play in The Hangover?	(JOIN (R film.performance.character) (AND (JOIN film.performance.film m.0n3xxpd) (JOIN (R film.actor.film) m.02_0d2)))

Table 3: Reasoning patterns with their corresponding definitions, example questions, and S-expressions.

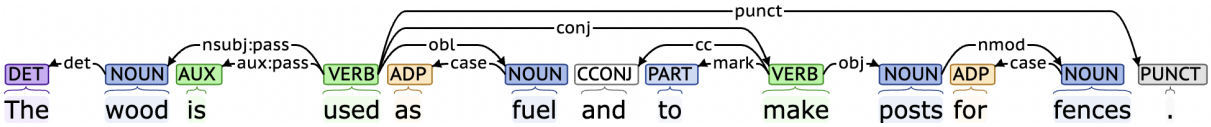


Figure 6: Example depicting the supplemental information provided by the *dependency tree*. The entities of interest are wood and fences, having the relationship material_used. The path *wood* ← *used* → *make* → *posts* → *fences* elicits this relationship.

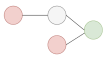
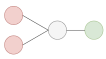
RP	Illustration	i.i.d.	Comp	Z.S.	Total
T-0		50.3	0.0	49.7	54.5
T-1		37.3	44.3	18.4	23.5
T-2		17.1	47.1	35.7	5.2
T-3		83.3	6.7	10.0	2.2
T-4		12.8	81.5	5.6	14.5
	ALL	40.8	24.9	34.3	100.0

Table 4: Distribution of reasoning patterns over the generalization splits (i.i.d., compositional (Comp), zero-shot (Z.S.)) of our modified WebQSP dataset.

A Task suite details

A.1 Data processing for reasoning pattern prediction and KBQA entity-ranking

We use the WebQSP dataset (Yih et al., 2016) for our two KBQA related experiments, i.e. reasoning pattern prediction and entity-ranking. An exploratory analysis of WebQSP highlighted a significant overlap of relations and classes across the train and test splits. Subsequently, we employed the approach of Jiang and Usbeck (2022) to obtain development and test splits that characterize different generalization levels in equal proportion. The three generalization levels for KBQA tasks include i.i.d, compositional, and zero-shot.

The i.i.d. case implies that the questions observed during inference follow similar logical templates to those during training; for example the questions "Who was the author of Oliver Twist?" and "Who wrote Pride and Prejudice?" follow similar logical templates. We contrast this with the compositional case, where questions in the test split operate over the same set of relations that were present in the training set (such as the "written-by" relation), but different logical templates. For example, the questions "Who wrote Pride and Prejudice?" and "Who wrote both The Talisman and It?" re-

Task	Text model	Graph model	Loss function	Metric
ETRE	RoBERTa[1]	{1,2,3}-layer RGAT[4]	cross-entropy (CE)	weighted F1
Form understanding	RoBERTa[1]	2-layer RGAT[4]	binary CE	F1
MLRE	mBERT-base[2]	2-layer RGCN [5]	CE	macro F1
Reasoning pattern prediction	T5-base[3]	2-layer RGCN[5]	CE	macro F1
KBQA answer-ranking	T5-base[3]	2-layer RGCN[5]	binary CE	Hits@K[6]

Table 5: Model configurations, training objectives, and evaluation metrics for each task. The text and graph model backbones listed in this table are used for the primary results in Table 2.

Task	LR	Batch size	Drop out	Temp.	Max input len	GNN layers	GNN hidden dim
ETRE (TDDMan)	1e-5	16	0.1	0.1	–	2	256
ETRE (TDDAuto)	1e-5	32	0.1	0.04	–	3	256
ETRE (TB-Dense)	1e-5	32	0.1	0.9	–	1	256
MLRE	1e-5	16	0.2	0.1	512	2	768
Reasoning pattern prediction	5e-5	6	0.2	0.1	512	2	768
KBQA entity-ranking	5e-5	4	0.2	0.1	1024	2	768
Form understanding	*Same settings as in* Nourbakhsh et al. (2024)						

Table 6: Hyperparameters used across tasks. Temperature refers to τ in CoD. All experiments use a shared space dimension of 2048.

quire reasoning over the same relation "written-by" but follows different reasoning paths, since the former involves only one constraint or entity, whereas the latter involves two. Finally, questions in the zero-shot split operate over new or unseen relations that were not present in the training dataset. For example, the questions "Who wrote Pride and Prejudice?" and "Who directed Pride and Prejudice in 2005?" involves different relations, i.e. "written-by" and "directed-by" respectively. We defer the readers to past work (Gu et al., 2021; Jiang and Usbeck, 2022; Dutt et al., 2023) for a more thorough description of the different generalization splits.

We characterize the complexity of the reasoning pattern to answer a given KBQA question based on Dutt et al. (2023). Given the modified version of WebQSP dataset, we identify the following five reasoning patterns that accounted for $\geq 97\%$ of the dataset across all splits. We describe the different reasoning patterns in Table 3 and outline their distribution in the our modified WebQSP dataset in Table 4.

To accommodate the input length constraints of models like T5, we simplify the representation of knowledge base entities in the linearized graph input. Instead of using full entity identifiers (e.g., `m.02896`), we assign short, unique placeholder tokens (e.g., `<E1>`, `<E2>`) to each entity as a part of the tokenizer vocabulary. This helps reduce the input sequence length and avoids unwanted subword tokenization. In addition, we ensure that these placeholder tokens are assigned consistently across modalities: the same entity is represented as node v_i in the graph and as token `<Ei>` in the linearized text.

A.2 MLRE dependency parsing illustration

See Figure 6.

A.3 FU example

We adapt an example to showcase the FU task from Nourbakhsh et al. (2024) in Figure 7.

B Task experiments details

We present the experimental details for different tasks. In Table 5, we outline the loss function that we are optimizing, the corresponding evaluation metric, and the backbone architectures used for the primary results reported in Table 2: the transformer model that encodes the textual information, and the specific GNN architecture that encodes the graph information. In Table 6, we provide hyperparameters values for our experiments. We also present statistics on the task suite datasets and training times in Table 7. All datasets we used are publicly available, and we follow the licensing terms and

Task	Dataset	Train	Test	Number of labels	Training time
ETRE	TDDMan	4,000	1,500	5	28 min
	TDDAuto	32,609	4,258	5	3h 40min
	TB-Dense	4,032	1,427	6	26 min
MLRE	REDFM (en)	8,504	1,235	32	6h 7min
	REDFM (es)	5,194	733	32	2h 30min
	REDFM (fr)	5,452	975	32	3h 14min
	REDFM (de)	5,909	811	32	2h 46min
	REDFM (it)	4,597	1,086	32	2h 38min
Reasoning pattern prediction	WebQSP	3,014	1,343	5	1h
KBQA answer-ranking	WebQSP	3,014	1,343	Number of gold answers	3h
Form understanding	SROIE	626	347	4	10h
	FUNSD	149	50	4	4h 36min
	CORD	800	100	30	17h 47min

Table 7: Task suite statistics and training times. We train for 1000 epochs for form understanding.

intended use of each.

C Extended CoD results

To further demonstrate the robustness and generality of CoD, we apply it to new model combinations on two representative tasks: reasoning pattern prediction and ETRE (Table 8). We also demonstrate additional CoD performance across each language data for MLRE in Table 9.

D Extended visualization results across tasks

D.1 ETRE results

See Figure 8 and Figure 9 for results on TimeBank-Dense and TDDAuto datasets, respectively. See Figure 10 for results on TDDMan dataset when no CoD is applied.

D.2 MLRE results

See Figure 11 for PCA plots, and Figure 12 for cosine similarity and distance metrics results.

D.3 FU results

See Figure 13 and Figure 14 for results on SROIE and FUNSD datasets, respectively.

D.4 RPP results

See Figure 15 for Reasoning Pattern Prediction task without CoD applied.

D.5 KBQA entity-ranking results

See Figure 16 and Figure 17 for results for KBQA entity-ranking with and without CoD applied, respectively.

[1] Liu et al. (2019)
[2] Devlin et al. (2019)
[3] Raffel et al. (2023)
[4] Busbridge et al. (2019)
[5] Schlichtkrull et al. (2017)
[6] K indicates the number of correct answers for an instance.

(a) Reasoning pattern prediction				
Text encoder	Graph encoder	Hybrid (CoD)	Text only	Graph only
T5	RGCN	**0.6190**	0.5700	0.5840
T5	RGAT	**0.6120**	0.5700	0.4966
BERT	RGCN	**0.5999**	0.5835	0.5840
BERT	RGAT	**0.5956**	0.5835	0.4966
GPT-2	RGCN	**0.6022**	0.5614	0.5840
GPT-2	RGAT	**0.6049**	0.5614	0.4966

(b) Event temporal relation extraction (ETRE)			
Text encoder	Graph encoder	Hybrid (CoD)	Text only
TDDMan			
BERT	GCN	0.411	**0.447**
BERT	RGCN	0.384	**0.447**
BERT	RGAT	**0.481**	0.447
RoBERTa	GCN	0.435	**0.445**
RoBERTa	RGCN	**0.452**	0.445
RoBERTa	RGAT	**0.551**	0.445
TDDAuto			
BERT	GCN	**0.631**	0.624
BERT	RGCN	**0.647**	0.624
BERT	RGAT	**0.683**	0.624
RoBERTa	GCN	**0.748**	0.689
RoBERTa	RGCN	0.665	**0.689**
RoBERTa	RGAT	**0.771**	0.689
TB-Dense			
BERT	GCN	**0.790**	0.775
BERT	RGCN	**0.782**	0.775
BERT	RGAT	**0.810**	0.775
RoBERTa	GCN	**0.805**	0.767
RoBERTa	RGCN	**0.847**	0.767
RoBERTa	RGAT	**0.856**	0.767

Note that we did not record numbers for the graph-only approach because the graph approach for this task yields incredibly poor results without the incorporation of linear transformers (Yao et al., 2024).

Table 8: Additional results for (a) Reasoning pattern prediction and (b) ETRE using different text and graph encoder backbones. CoD consistently improves over baselines across all combinations in Reasoning pattern prediction, and improves 78% of the times across all 18 cases for ETRE. These results demonstrate CoD's generality across diverse model architecture combinations.

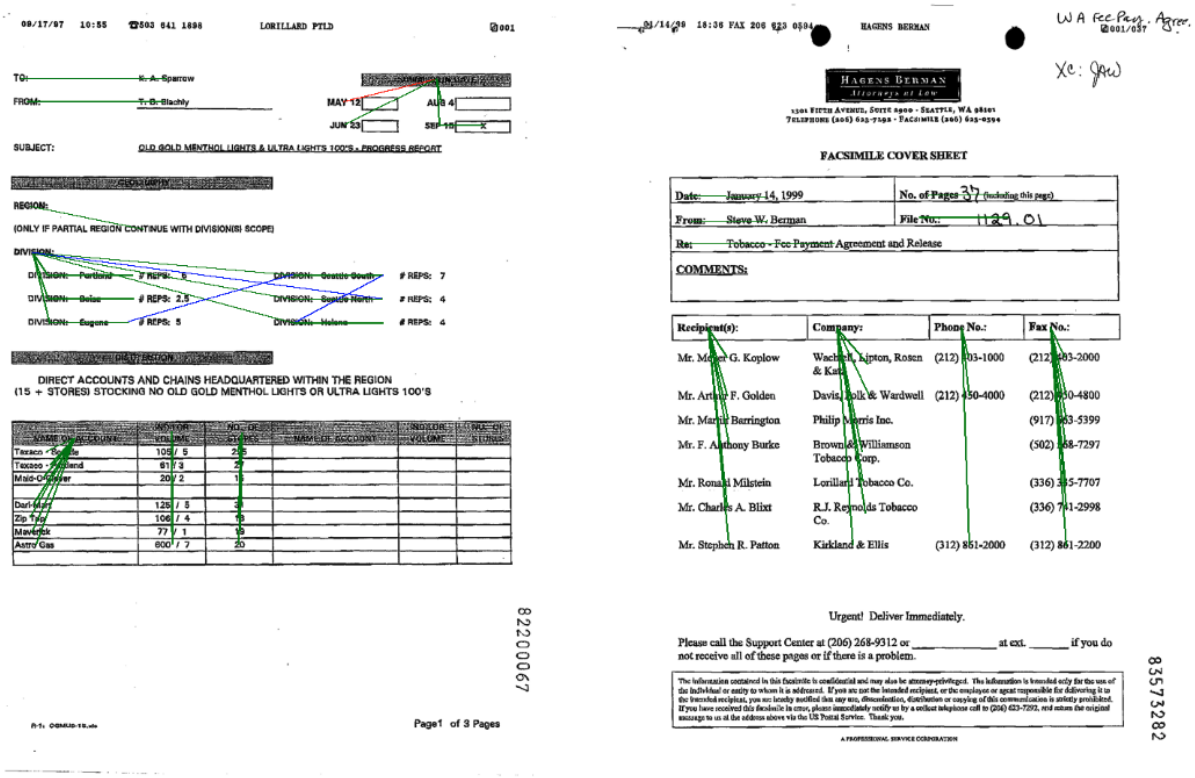
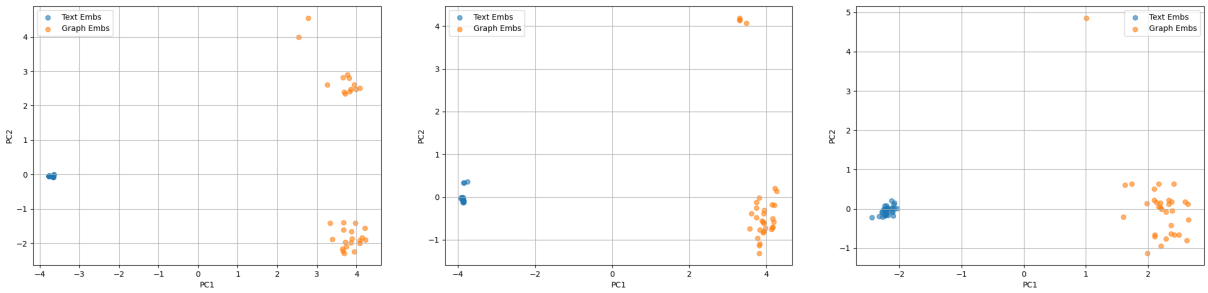


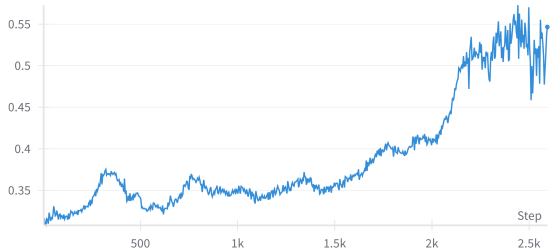
Figure 7: An example of FU task from the FUNSD dataset, adapted from Nourbakhsh et al. (2024). Green links show correct predictions. Red links show false negatives. Blue links show false positives.

Language	Text only	Graph only	Hybrid + CoD	Hybrid + no-CoD
de	**80.41** $\pm$ 0.61	47.13 $\pm$ 2.76	80.35 $\pm$ 0.71	79.55 $\pm$ 0.40
en	**85.94** $\pm$ 1.41	52.21 $\pm$ 0.56	84.57 $\pm$ 2.25	84.74 $\pm$ 1.07
es	**80.49** $\pm$ 0.61	51.21 $\pm$ 1.47	76.64 $\pm$ 1.09	80.26 $\pm$ 0.44
fr	77.47 $\pm$ 0.73	45.62 $\pm$ 1.60	**78.80** $\pm$ 0.58	78.31 $\pm$ 0.78
it	74.25 $\pm$ 0.36	46.61 $\pm$ 1.98	72.67 $\pm$ 1.40	**74.76** $\pm$ 1.02
Avg	**79.71** $\pm$ 3.95	48.55 $\pm$ 3.21	78.61 $\pm$ 4.17	79.53 $\pm$ 3.32

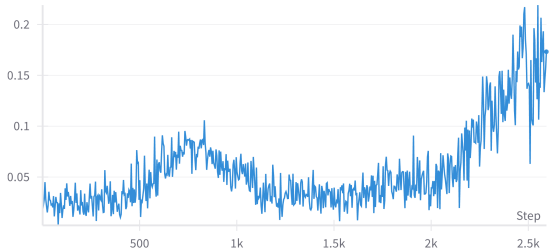
Table 9: F1 score results on MLRE task for the RED^{fm} dataset.



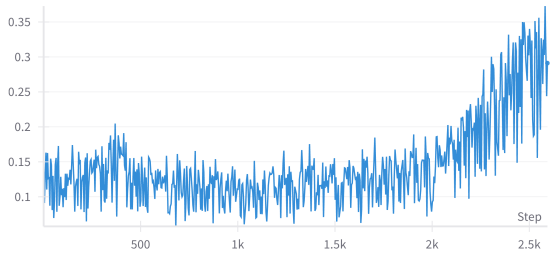
(a) Initial epoch (b) Intermediate epoch (c) Final epoch



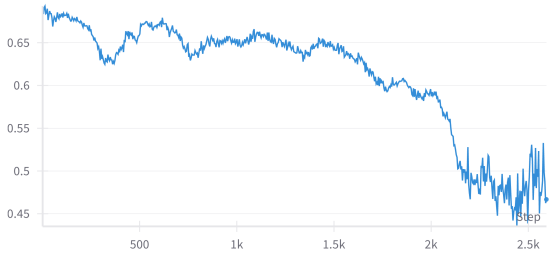
(d) Cosine similarity



(e) Distance within text

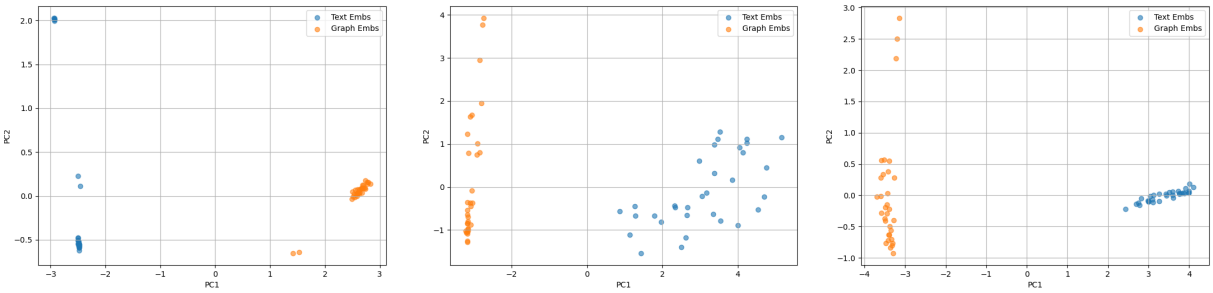


(f) Distance within graph

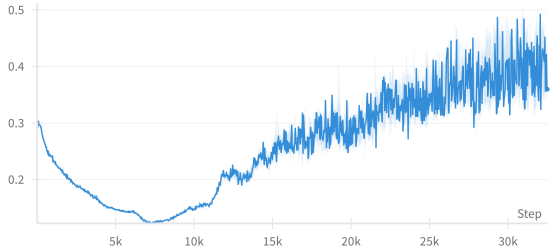


(g) Distance between text and graph

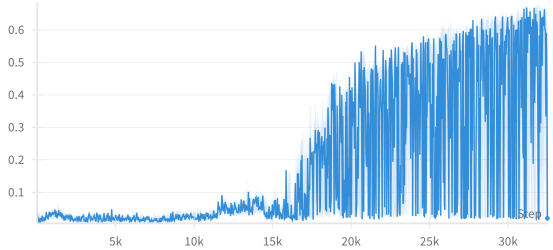
Figure 8: Results for ETRE on the TimeBank-Dense dataset.



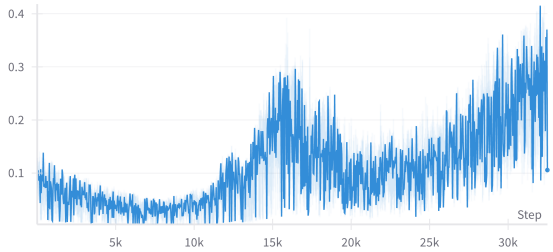
(a) Initial epoch (b) Intermediate epoch (c) Final epoch



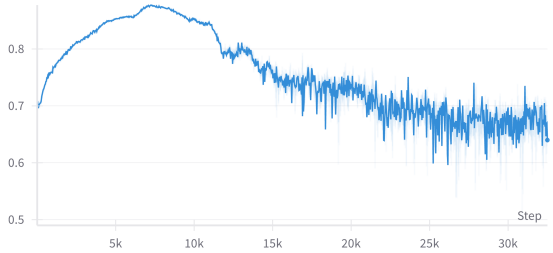
(d) Cosine similarity



(e) Distance within text

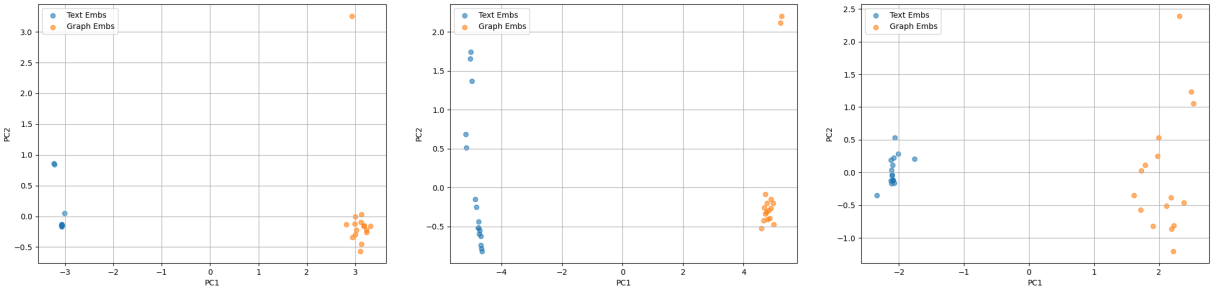


(f) Distance within graph

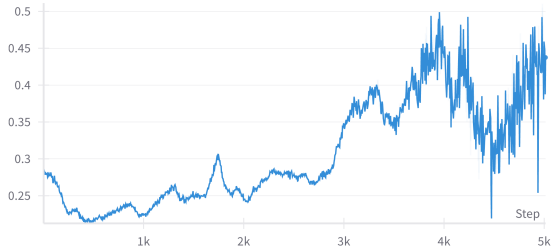


(g) Distance between text and graph

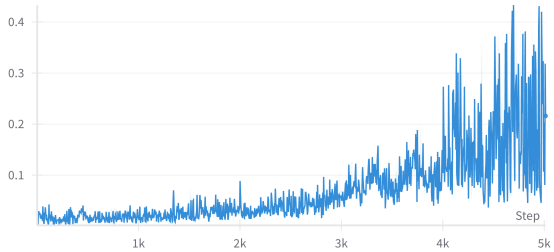
Figure 9: Results for ETRE on the TDDAuto dataset.



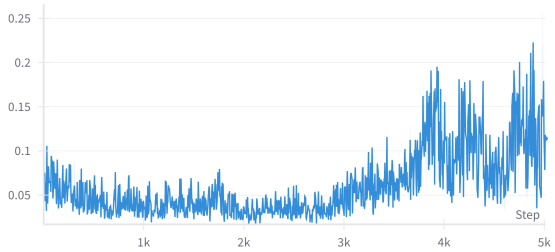
(a) Initial epoch (b) Intermediate epoch (c) Final epoch



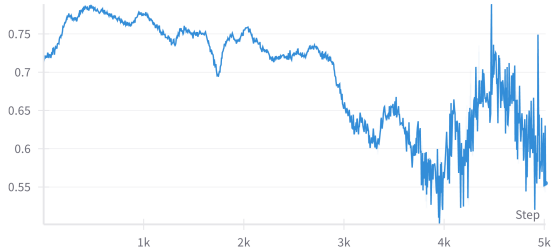
(d) Cosine similarity



(e) Distance within text

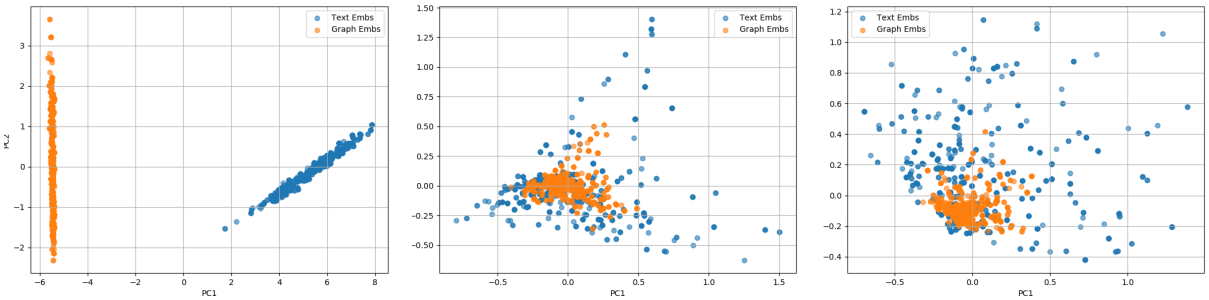


(f) Distance within graph

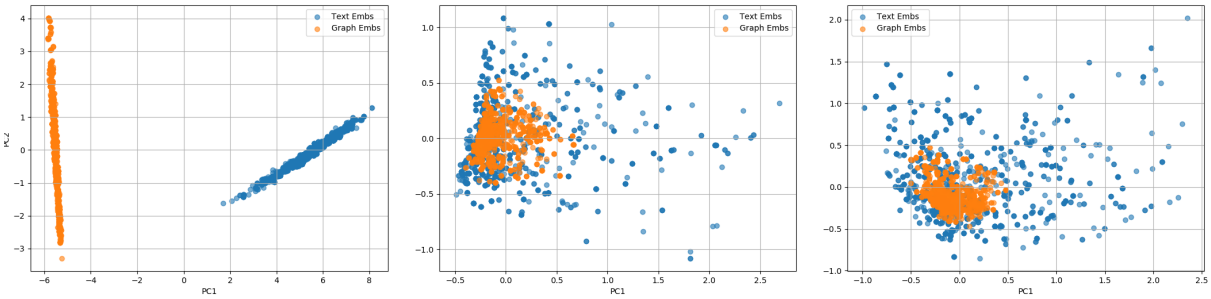


(g) Distance between text and graph

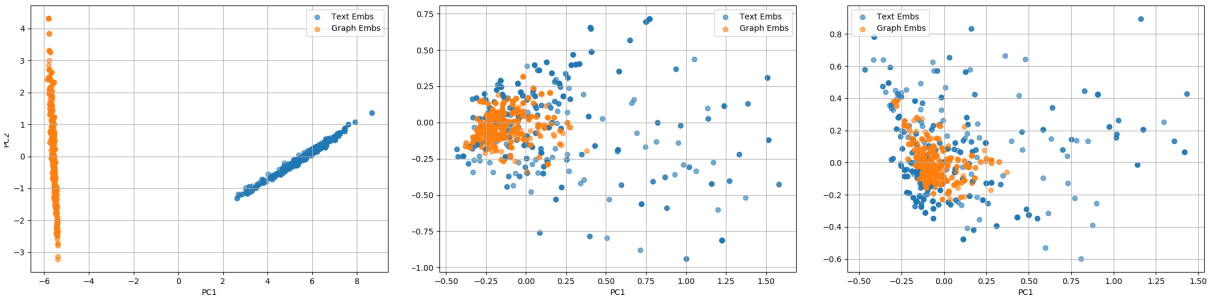
Figure 10: Results for ETRE on the TDDMan dataset when no CoD is applied.



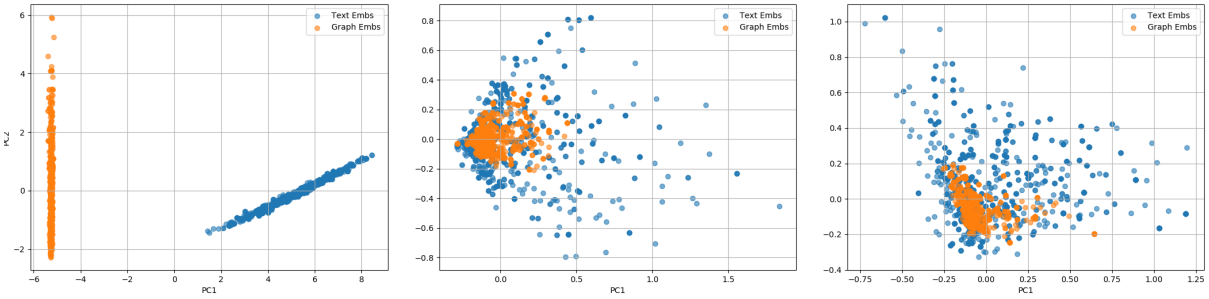
(a) Initial epoch (de) (b) Intermediate epoch (de) (c) Final epoch (de)



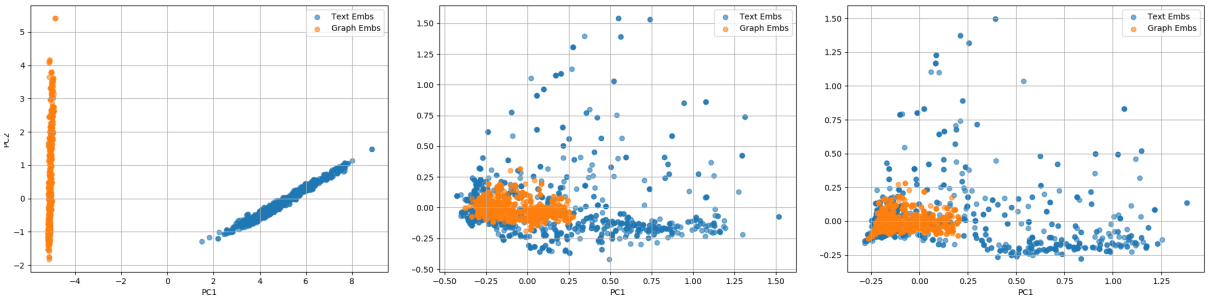
(d) Initial epoch (en) (e) Intermediate epoch (en) (f) Final epoch (en)



(g) Initial epoch (es) (h) Intermediate epoch (es) (i) Final epoch (es)

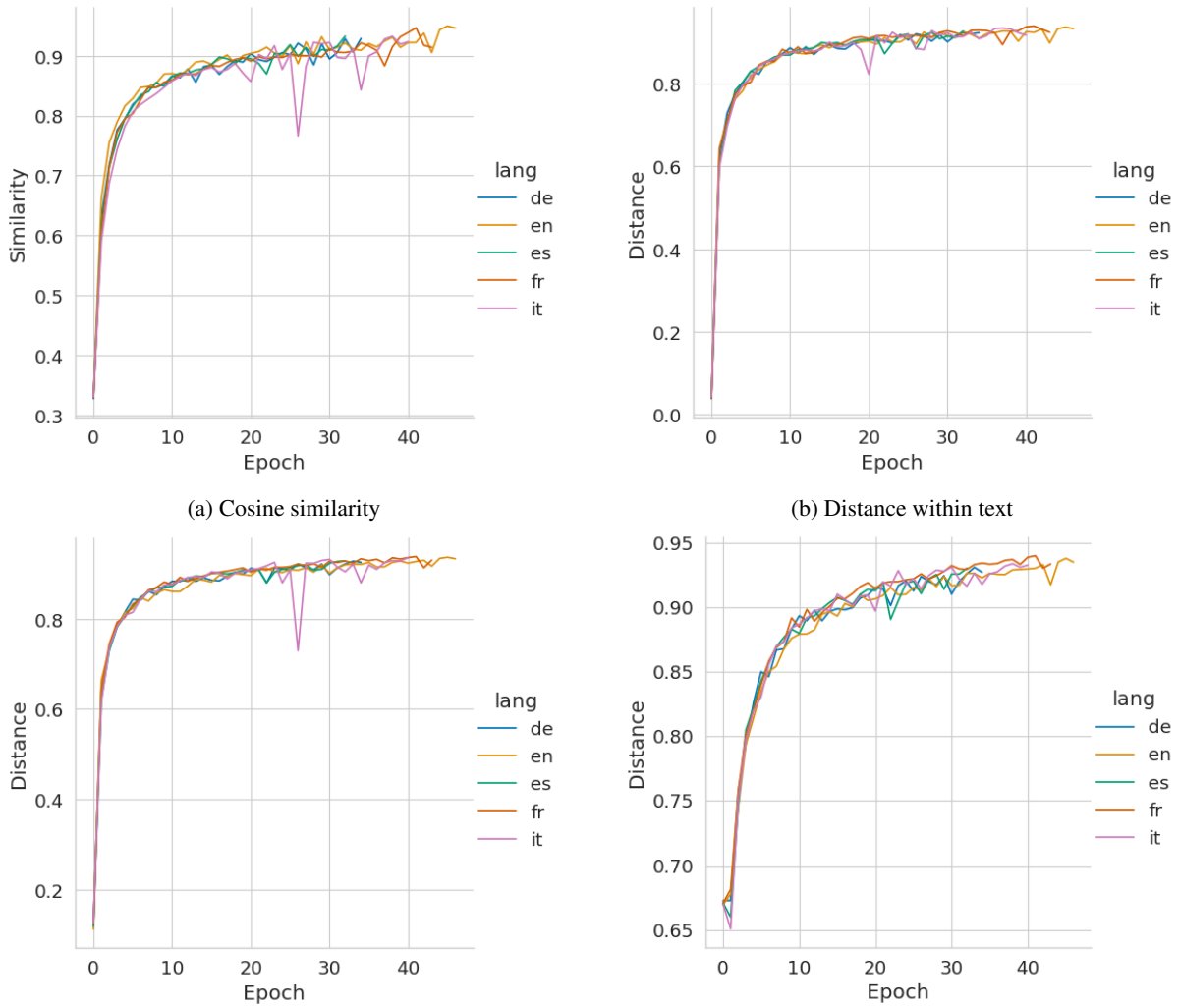


(j) Initial epoch (fr) (k) Intermediate epoch (fr) (l) Final epoch (fr)



(m) Initial epoch (it) (n) Intermediate epoch (it) (o) Final epoch (it)

Figure 11: PCA plots for MLRE across the different languages in the RED^{fm} dataset.



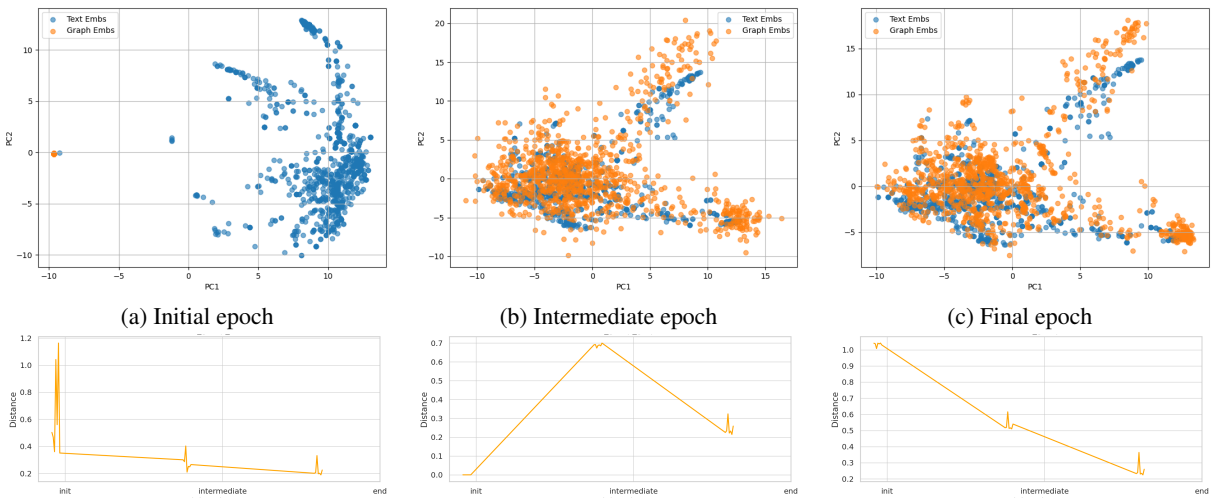
(a) Cosine similarity

(b) Distance within text

(c) Distance within graph

(d) Distance between text and graph

Figure 12: Cosine similarity and distance results for MLRE on the RED^{fm} dataset.



(a) Initial epoch

(b) Intermediate epoch

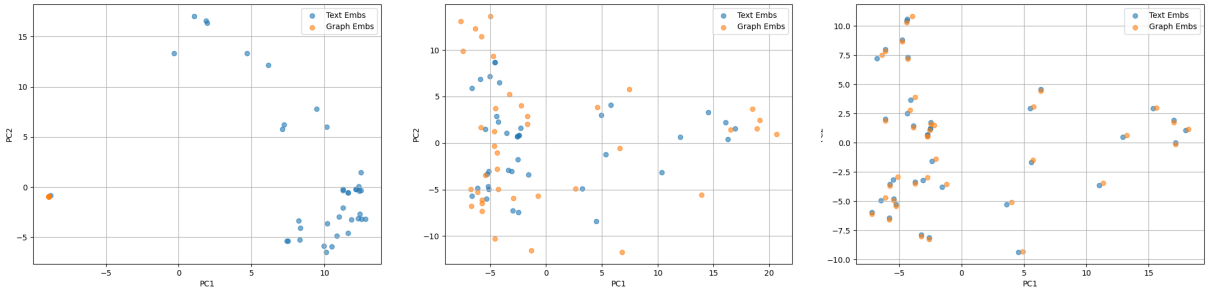
(c) Final epoch

(d) Distance within text

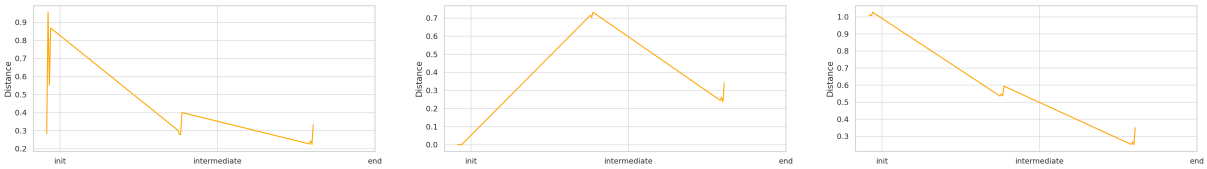
(e) Distance within graph

(f) Distance between text and graph

Figure 13: Results for form understanding on the SROIE dataset.

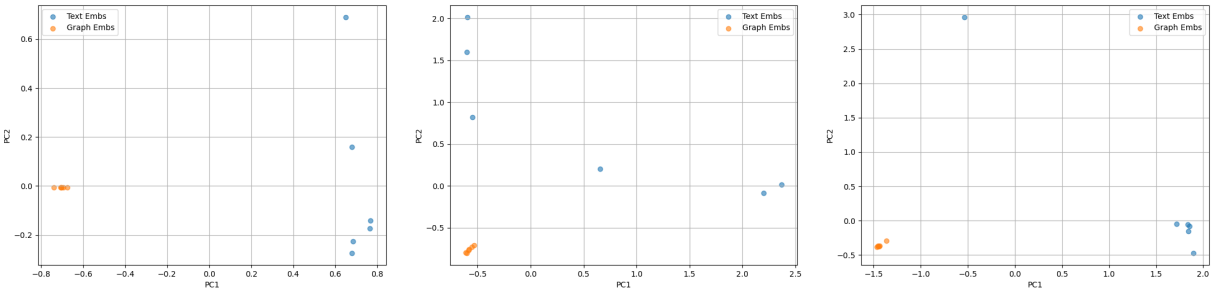


(a) Initial epoch (b) Intermediate epoch (c) Final epoch

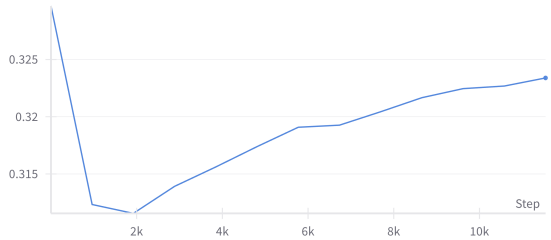


(d) Distance within text (e) Distance within graph (f) Distance between text and graph

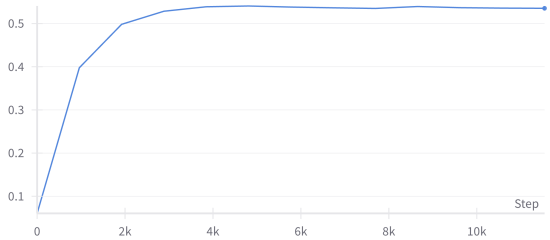
Figure 14: Results for form understanding on the FUNSD dataset.



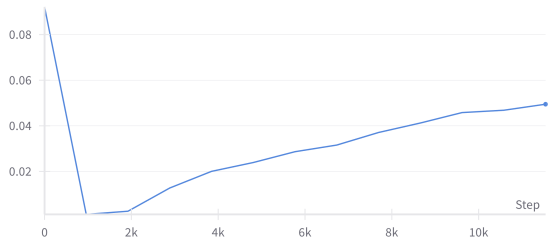
(a) Initial epoch (b) Intermediate epoch (c) Final epoch



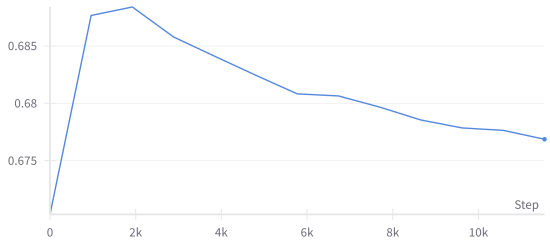
(d) Cosine similarity



(e) Distance within text

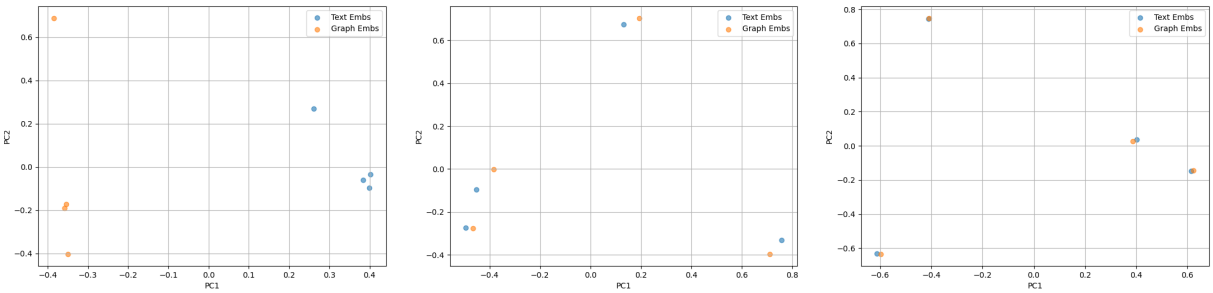


(f) Distance within graph

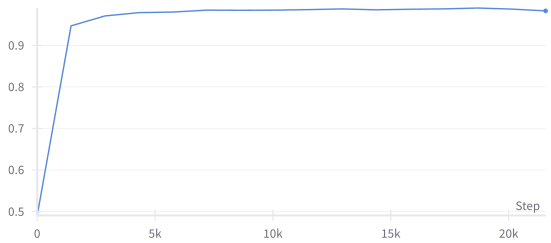


(g) Distance between text and graph

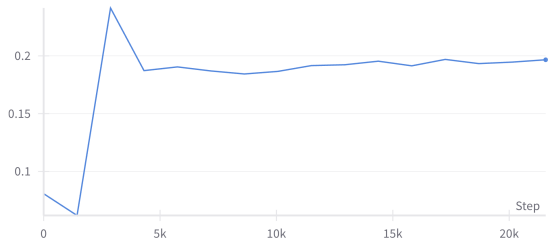
Figure 15: Results for reasoning pattern prediction on the WebQSP dataset when no CoD is applied.



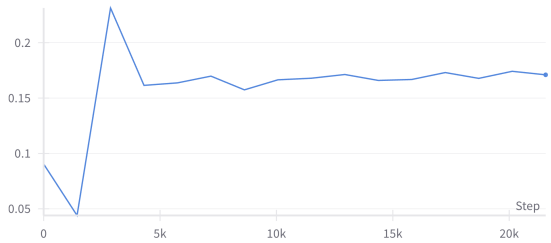
(a) Initial epoch (b) Intermediate epoch (c) Final epoch



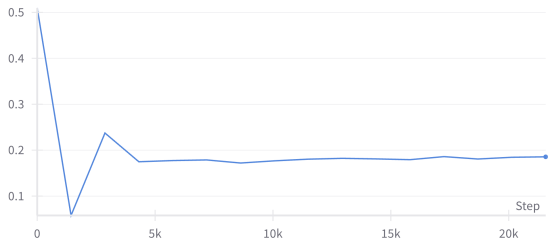
(d) Cosine similarity



(e) Distance within text

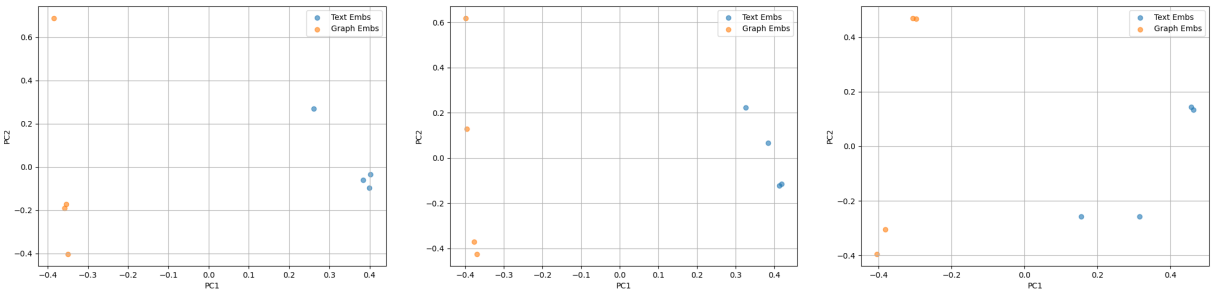


(f) Distance within graph

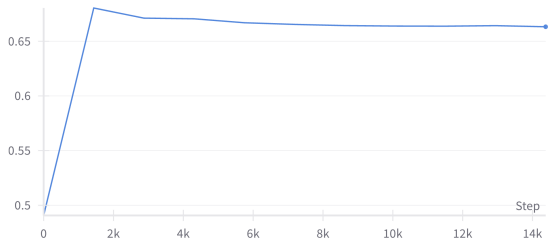


(g) Distance between text and graph

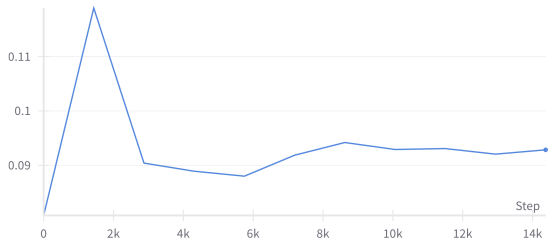
Figure 16: Results for KBQA entity-ranking on the WebQSP dataset.



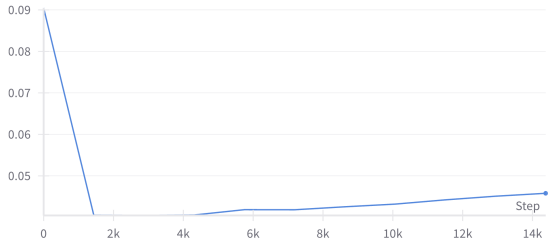
(a) Initial epoch (b) Intermediate epoch (c) Final epoch



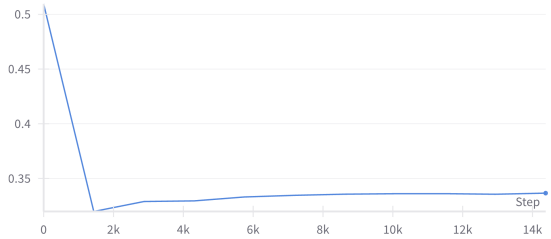
(d) Cosine similarity



(e) Distance within text



(f) Distance within graph



(g) Distance between text and graph

Figure 17: Results for KBQA entity-ranking on the WebQSP dataset when no CoD is applied.