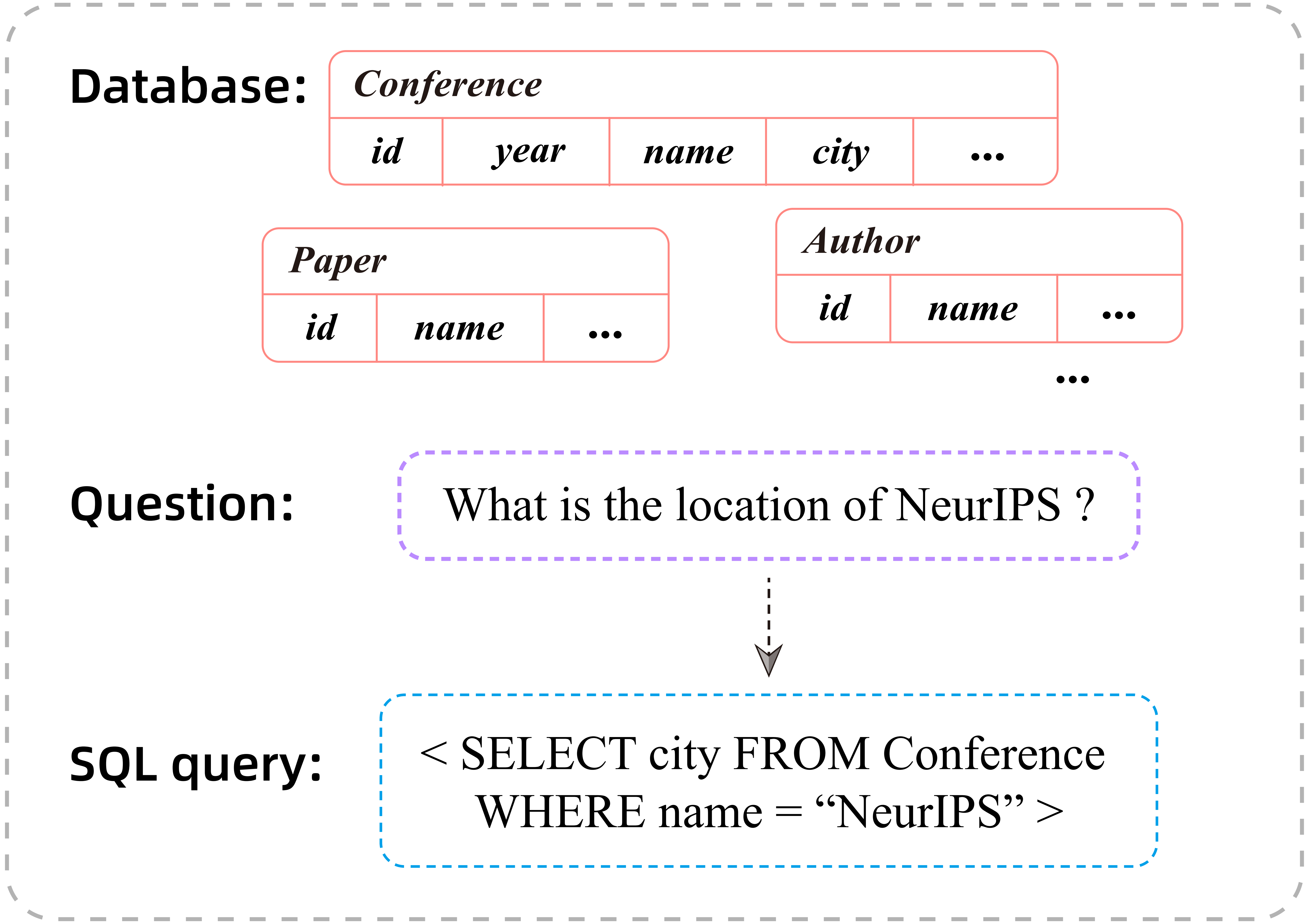
SADGA: Structure-Aware Dual Graph Aggregation Network for Text-to-SQL

Ruichu Cai[1], Jinjie Yuan[1], Boyan Xu[1], Zhifeng Hao[1,2]
[1] School of Computer Science, Guangdong University of Technology, Guangzhou, China
[2] College of Science, Shantou University, Shantou, China
cairuichu@gdut.edu.cn, yuanjinjie0320@gmail.com
hpakyim@gmail.com, zfhao@gdut.edu.cn

NeurIPS 2021

Introduction: Text-to-SQL

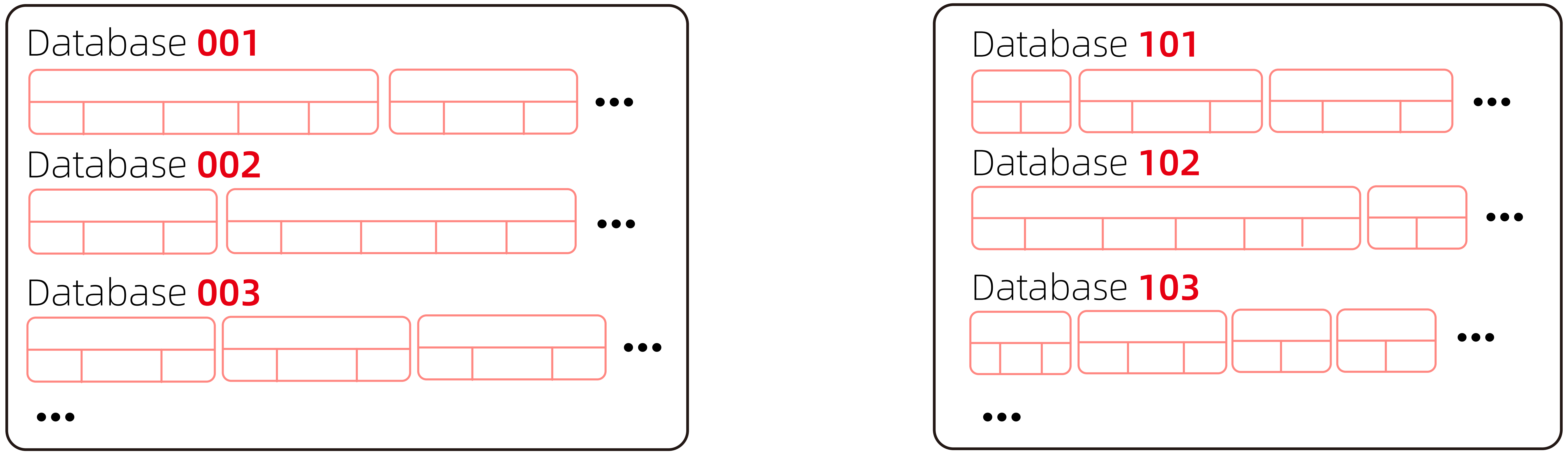
- Given a **question** and a **database**, automatically generate a **SQL query**.



Introduction: Cross-Domain Text-to-SQL

- **Cross-Domain** Text-to-SQL: Generalize the model to **unseen** database schema.

The **Train** Set Databases | The **Test** Set Databases



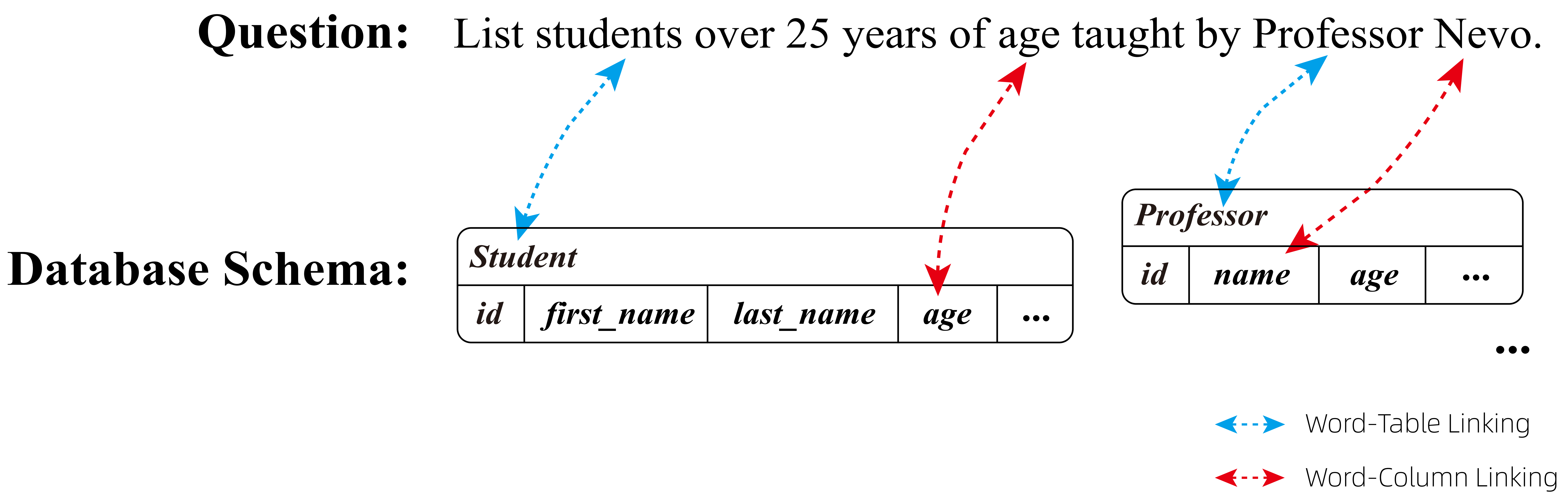
The databases do **not overlap** between the train and test sets.

Core Issue: Question-Schema Linking

How to build the linking between the natural language question and database schema?

Core Issue: Question-Schema Linking

How to build the linking between the natural language question and database schema?



Existing Works

Matching-based, e.g., IRNet [ACL 2019]:

- Use a simple **string matching** approach to link the question words and tables/columns.

Existing Works

Matching-based, e.g., IRNet [ACL 2019]:

- Use a simple **string matching** approach to link the question words and tables/columns.

Learning-based, e.g., RATSQL [ACL 2020]:

- Apply a **Relation-Aware Transformer** to globally learn the linking over the question and schema with **pre-defined** relations.

Limitations

a. The **structural gap** between the encoding process of the question and database schema;

b. Highly relying on **pre-defined** string-match linking maybe result in:

(i) unsuitable linking,

(ii) the latent association between question words and tables/columns to be undetectable.

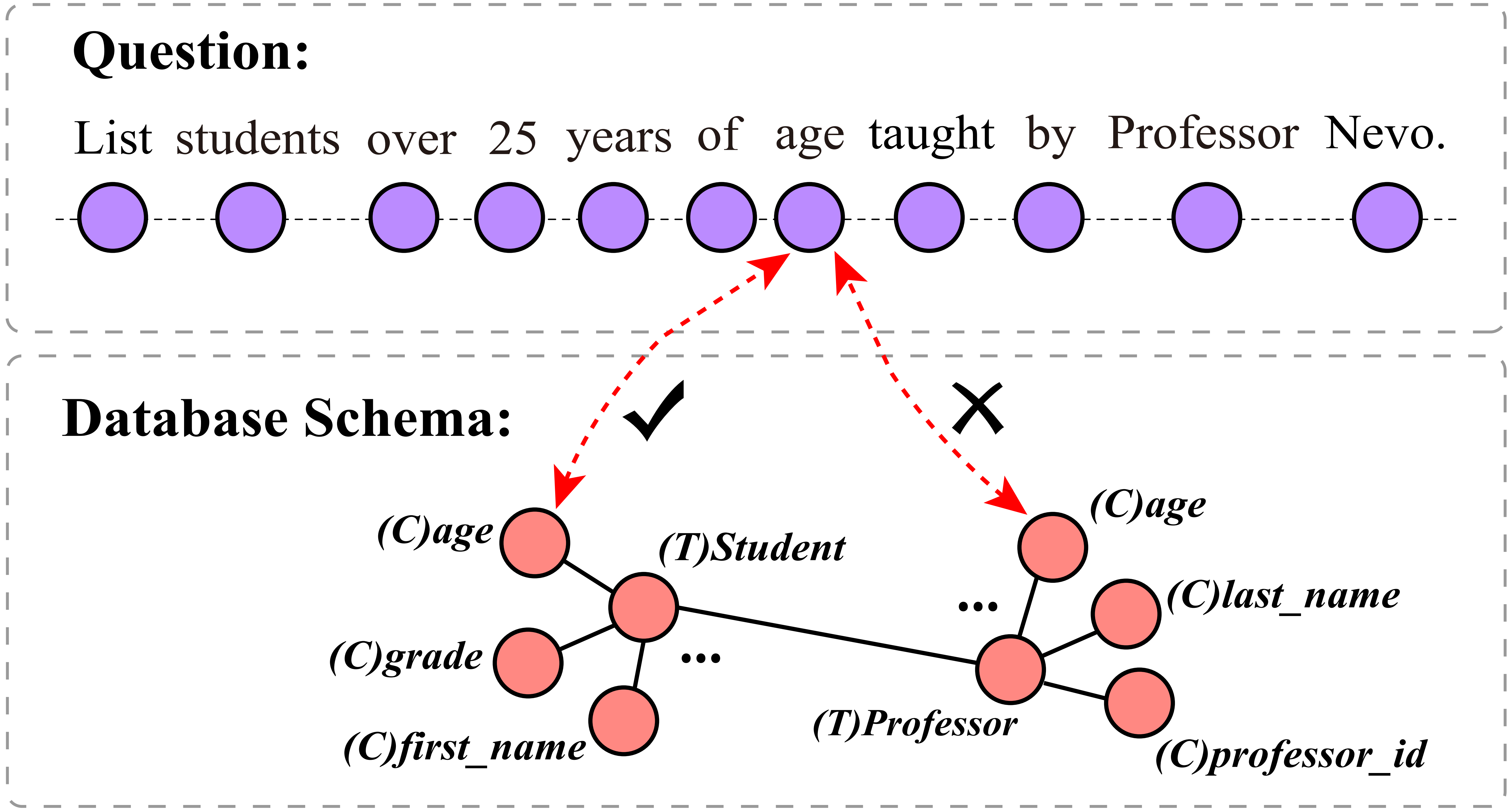
Limitations

a. The **structural gap** between the encoding process of the question and database schema;

b. Highly relying on **pre-defined** string-match linking maybe result in:

(i) unsuitable linking,

(ii) the latent association between question words and tables/columns to be undetectable.

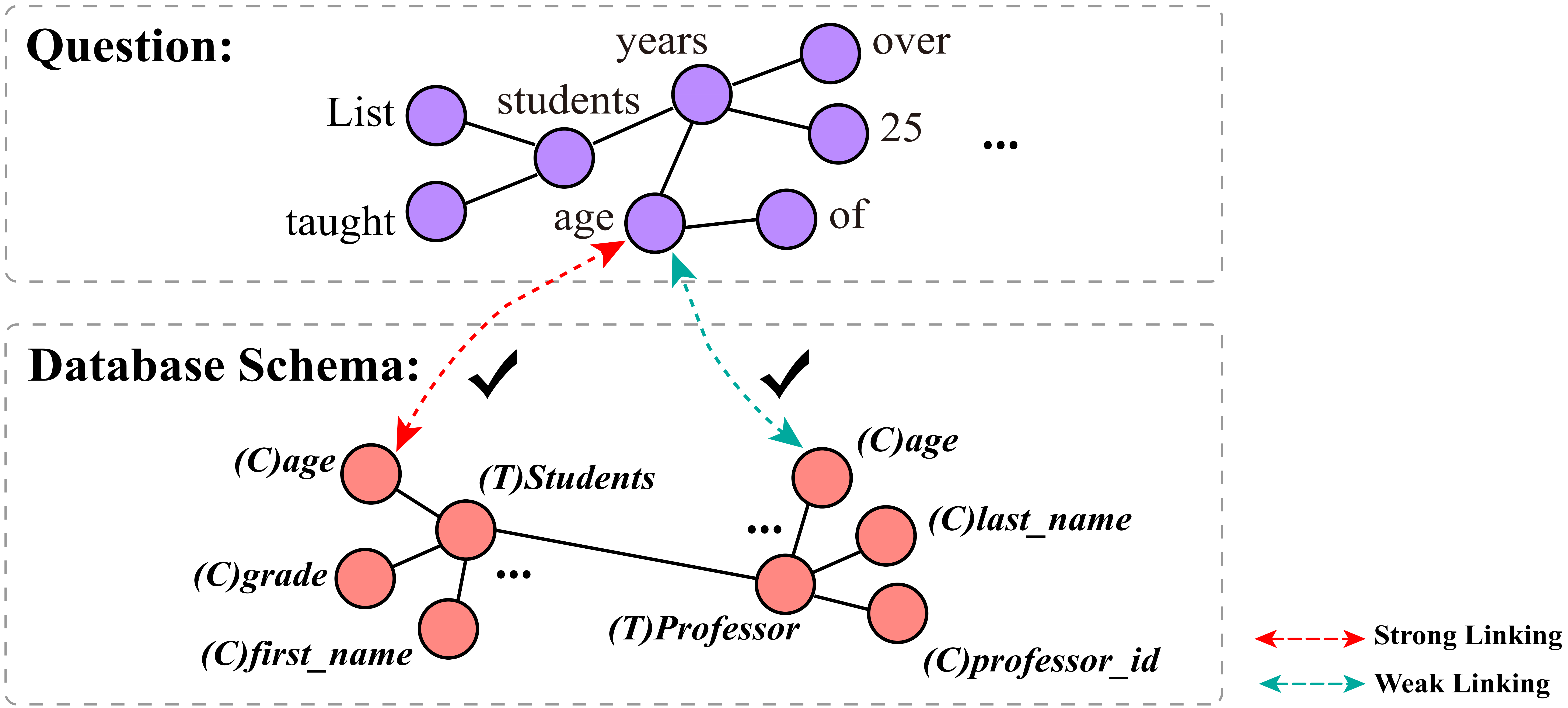


Our Solution

We propose a **Structure-Aware Dual Graph Aggregation Network (SADGA)** to perform Question-Schema Linking fully taking advantage of the **global** and **local** structural information.

Our Solution

We propose a **Structure-Aware Dual Graph Aggregation Network (SADGA)** to perform Question-Schema Linking fully taking advantage of the **global** and **local** structural information.



Structure-Aware Dual Graph Aggregation Network

A. Dual-Graph Construction

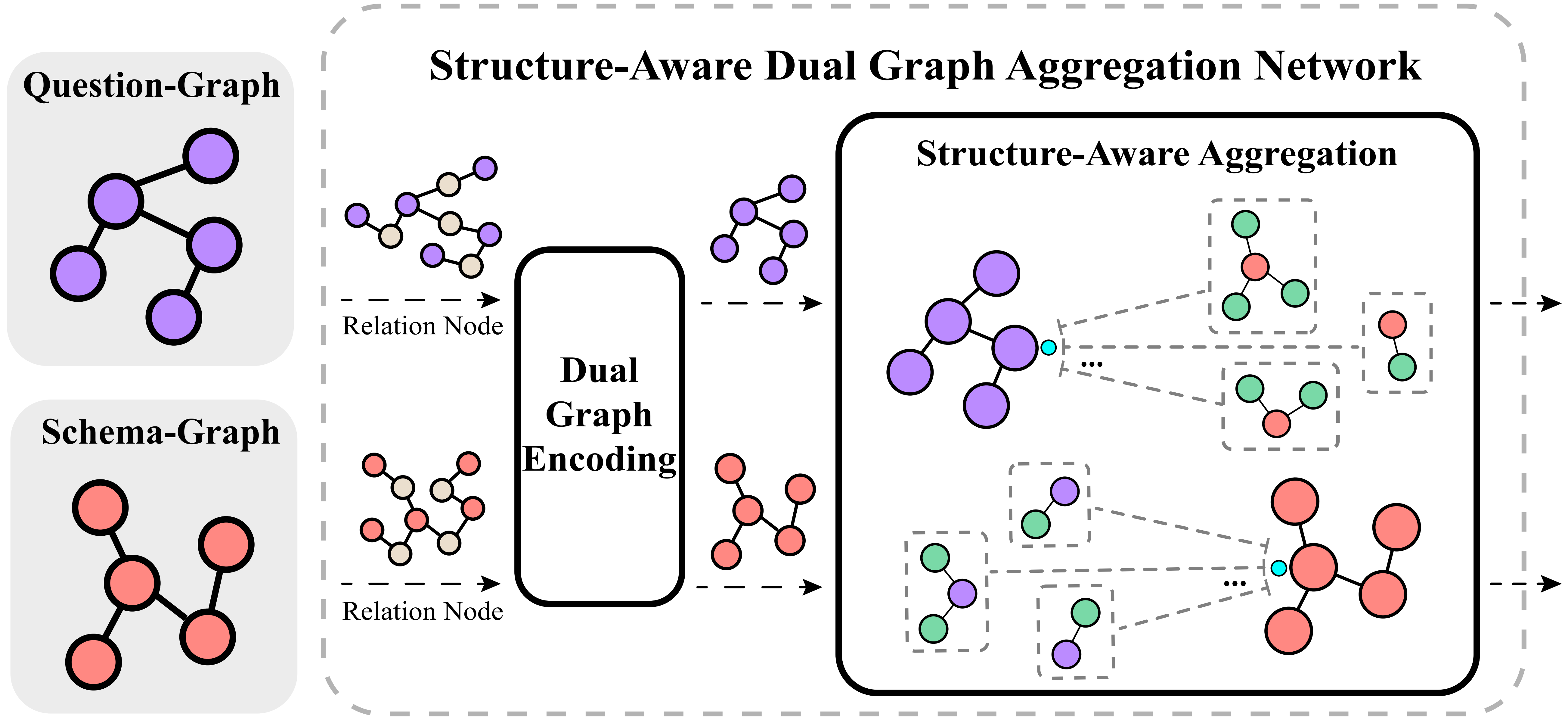
B. Dual-Graph Encoding

C. Structure-Aware Aggregation

C.1 Global Graph Linking

C.2 Local Graph Linking

C.3 Dual-Graph Aggregation Mechanism



Structure-Aware Dual Graph Aggregation Network

A. Dual-Graph Construction

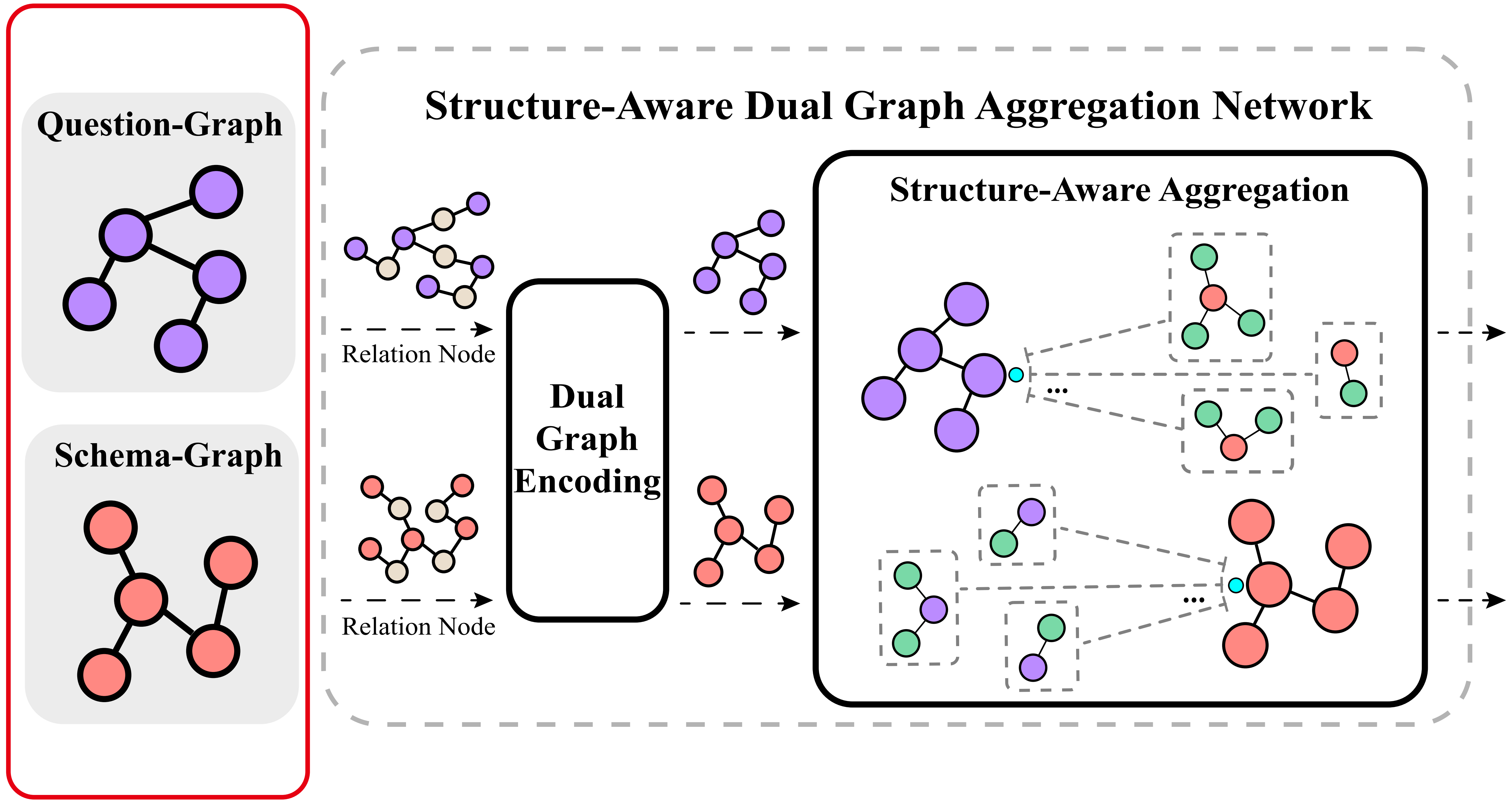
B. Dual-Graph Encoding

C. Structure-Aware Aggregation

C.1 Global Graph Linking

C.2 Local Graph Linking

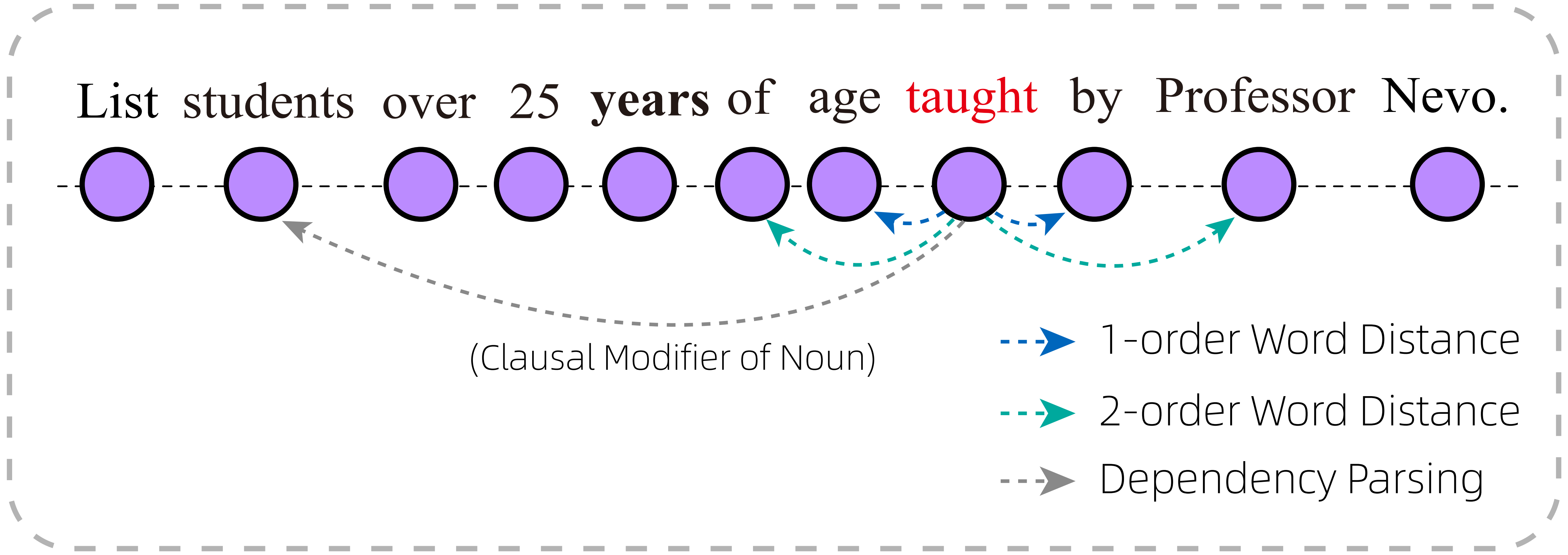
C.3 Dual-Graph Aggregation Mechanism



A. Dual-Graph Construction

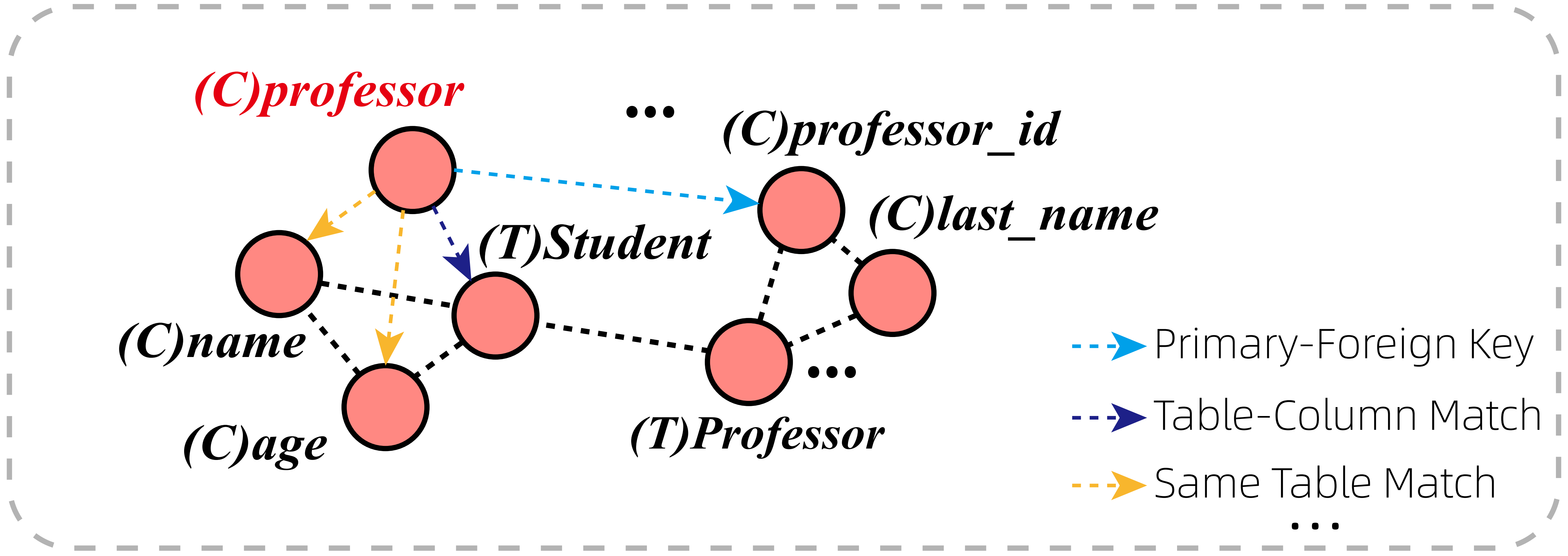
- Question-Graph

Take the word **taught** as a example:



- Schema-Graph

Take the column **professor** as a example:



- **Cross-Graph Relations**

Word-Table: Exact String Match, Partial String Match

Word-Column: Exact String Match, Partial String Match, Value Match

Structure-Aware Dual Graph Aggregation Network

A. Dual-Graph Construction

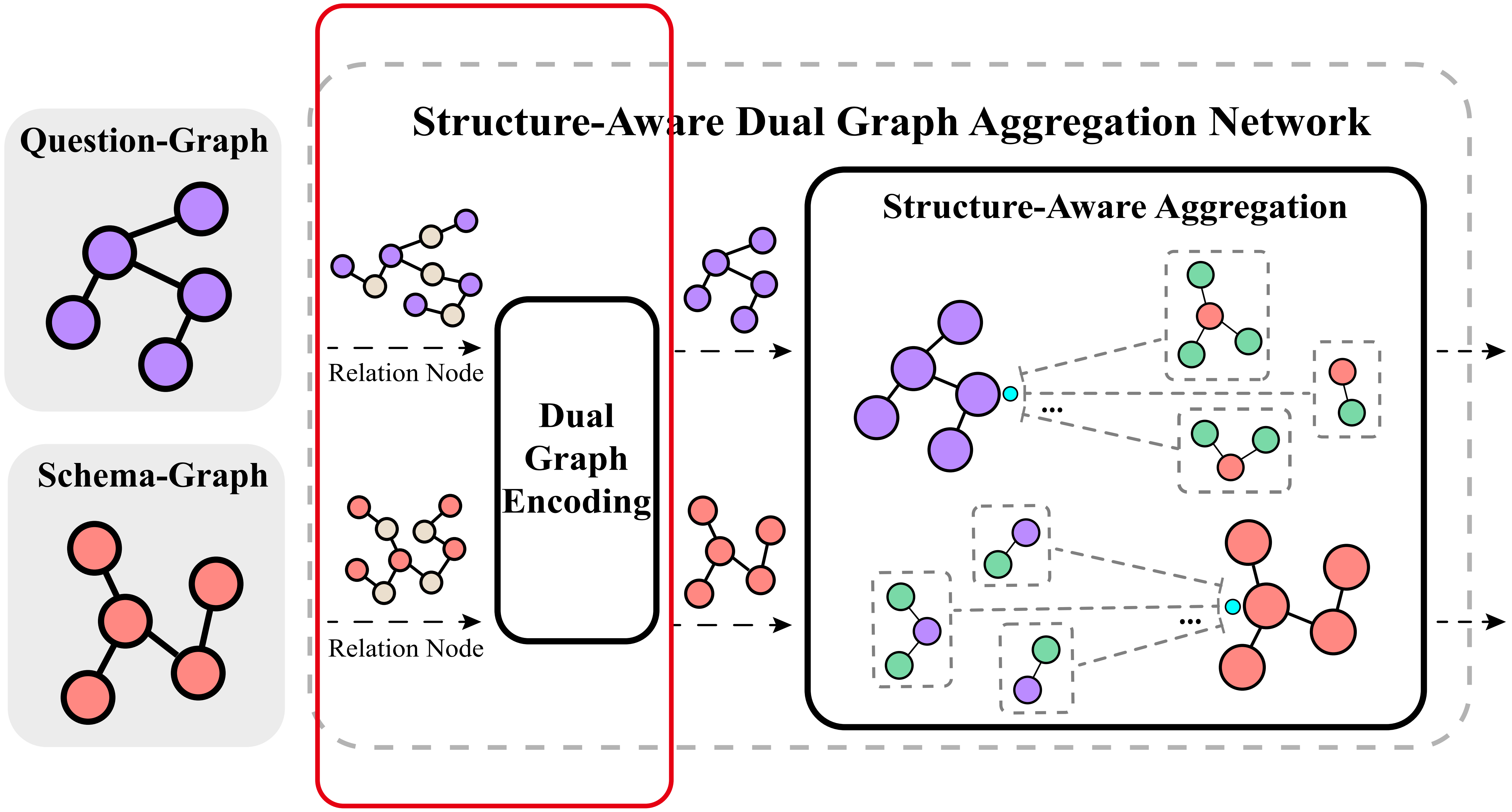
B. Dual-Graph Encoding

C. Structure-Aware Aggregation

C.1 Global Graph Linking

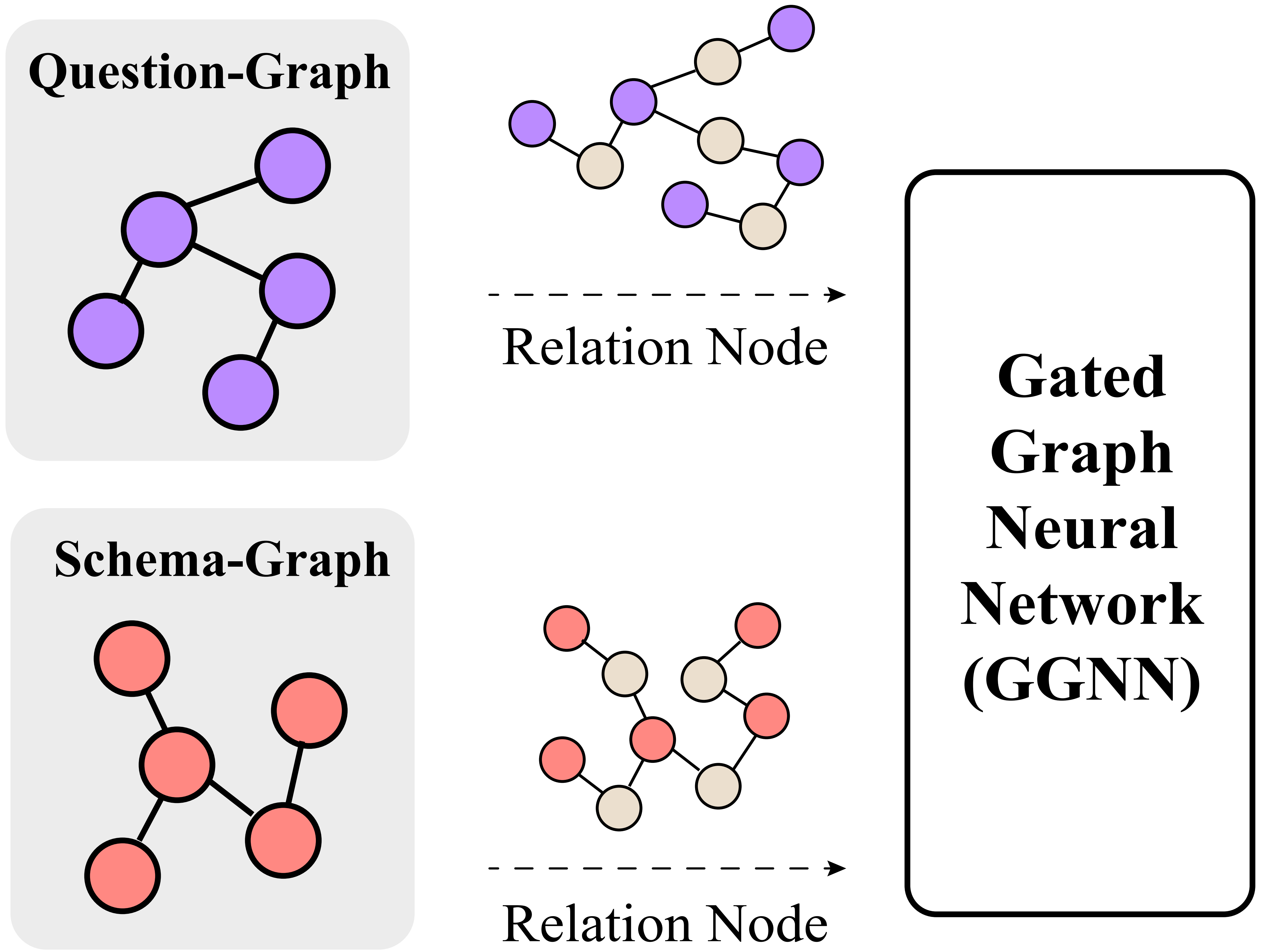
C.2 Local Graph Linking

C.3 Dual-Graph Aggregation Mechanism



B. Dual-Graph Encoding

- **Gated Graph Neural Network (GGNN)** is employed to encode the node representation of dual-graph by performing message propagation among the **self-structure**.



Structure-Aware Dual Graph Aggregation Network

A. Dual-Graph Construction

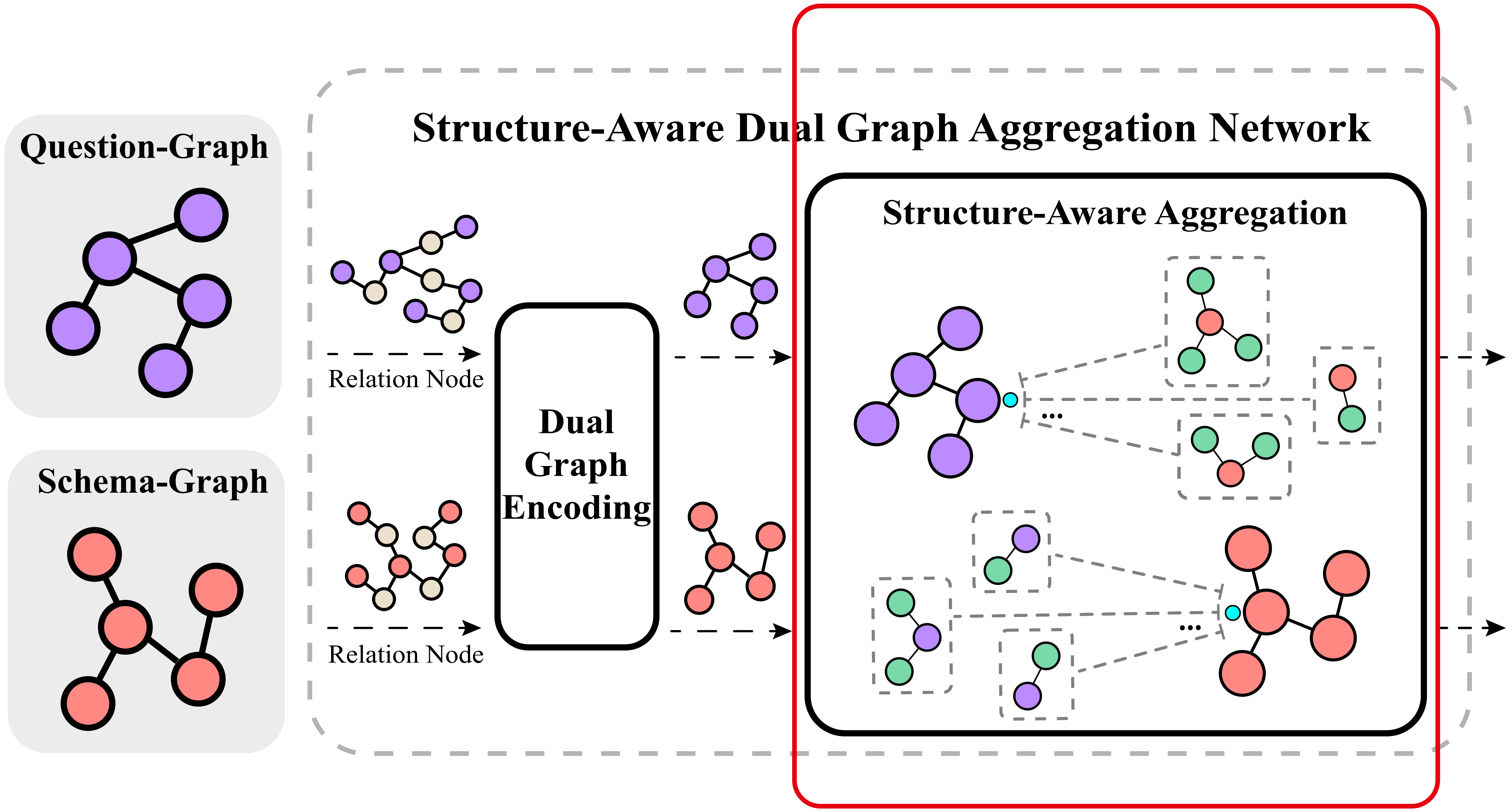
B. Dual-Graph Encoding

C. Structure-Aware Aggregation

C.1 Global Graph Linking

C.2 Local Graph Linking

C.3 Dual-Graph Aggregation Mechanism



C. Structure-Aware Aggregation

Given **Query-Graph** $\mathcal{G}_q$ and **Key-Graph** $\mathcal{G}_k$, we define the Structure-Aware Graph Aggregation to update **Query-Graph** $\mathcal{G}_q$:

$$\mathcal{G}_q^{Aggr} = \mathrm{GraphAggr}(\mathcal{G}_q, \mathcal{G}_k)$$

C. Structure-Aware Aggregation

Given **Query-Graph** $\mathcal{G}_q$ and **Key-Graph** $\mathcal{G}_k$, we define the Structure-Aware Graph Aggregation to update **Query-Graph** $\mathcal{G}_q$:

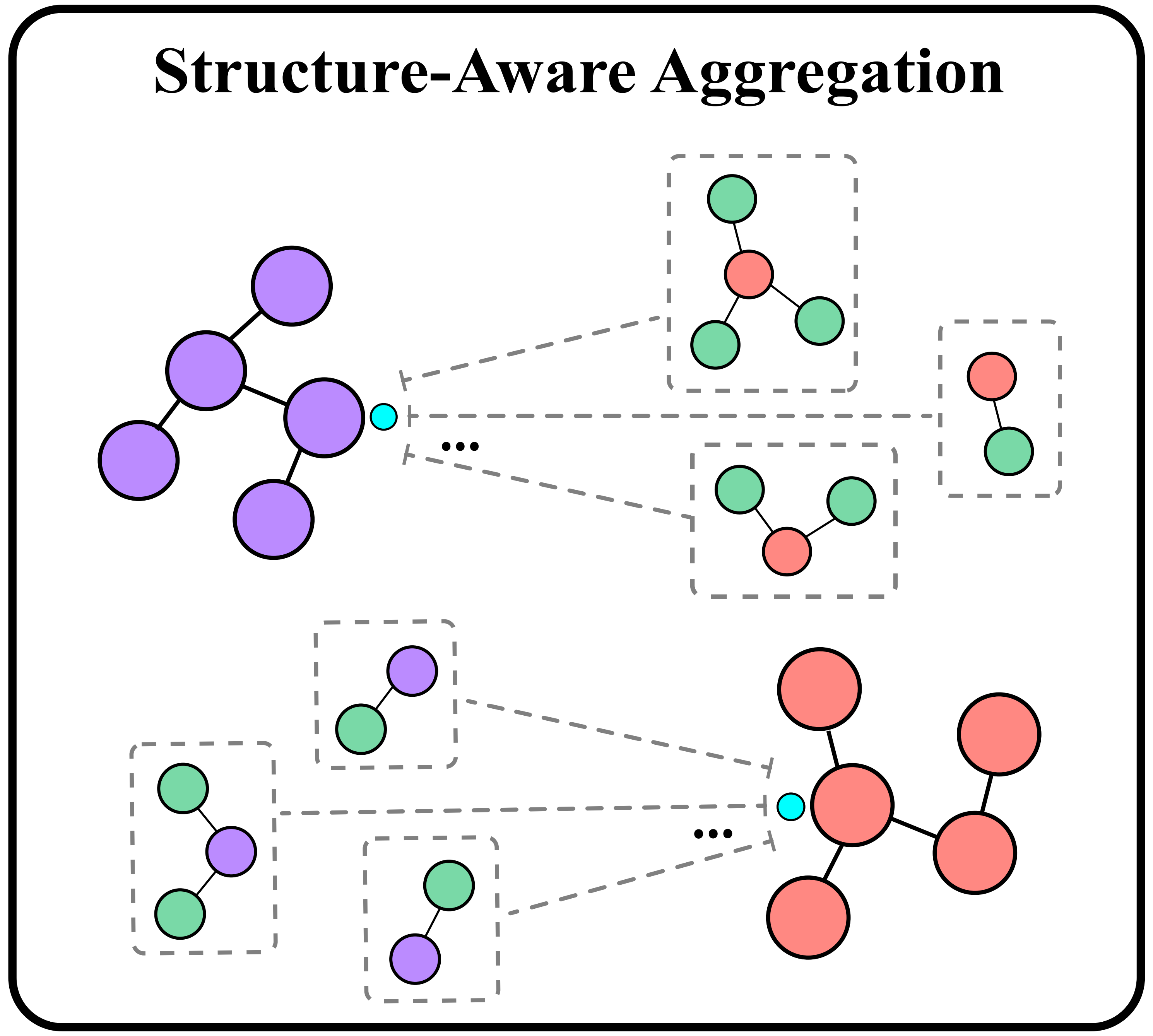
$$\mathcal{G}_q^{Aggr} = \mathrm{GraphAggr}(\mathcal{G}_q, \mathcal{G}_k)$$

Update **Question-Graph** $\mathcal{G}_Q$:

$$\mathcal{G}_Q^{Aggr} = \mathrm{GraphAggr}(\mathcal{G}_Q, \mathcal{G}_S)$$

Update **Schema-Graph** $\mathcal{G}_S$:

$$\mathcal{G}_S^{Aggr} = \mathrm{GraphAggr}(\mathcal{G}_S, \mathcal{G}_Q)$$



C. Structure-Aware Aggregation

Given **Query-Graph** $\mathcal{G}_q$ and **Key-Graph** $\mathcal{G}_k$:

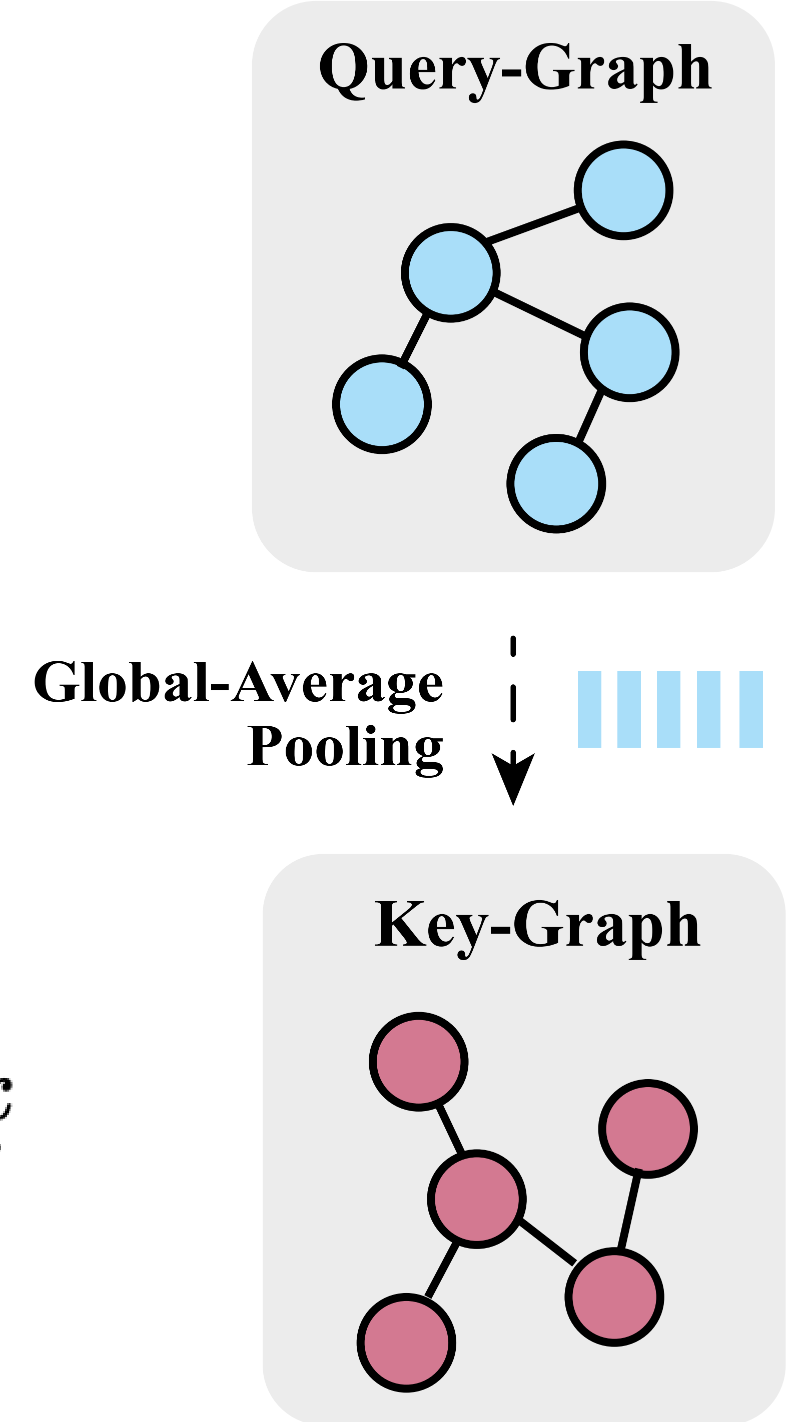
$$\mathcal{G}_q = \{\boldsymbol{h}_i^q\}_{i=1}^m \qquad \mathcal{G}_k = \{\boldsymbol{h}_j^k\}_{j=1}^n$$

In the beginning,

Global **Query-Graph** Vector $\leftarrow$ $$\boldsymbol{h}_{glob}^q = \frac{1}{m}\sum\nolimits_{i=1}^m \boldsymbol{h}_i^q$$

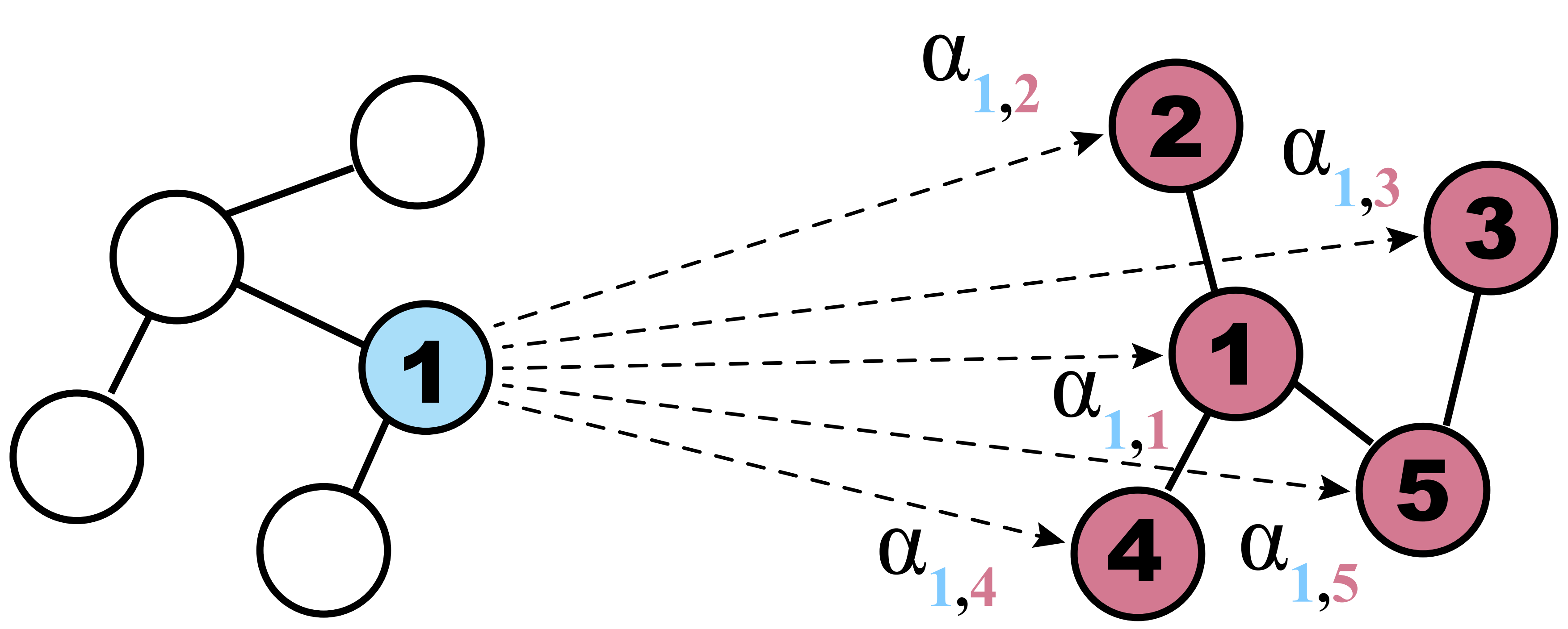
Relevant Score $\leftarrow$ $$e_j = \theta\left({\boldsymbol{h}_{glob}^q}^T \boldsymbol{W}_g \boldsymbol{h}_j^k\right)$$

Query-aware Representation $\leftarrow$ $$\boldsymbol{h}_j^k = (1 - e_j)\boldsymbol{W}_{qg}\boldsymbol{h}_{glob}^q + e_j \boldsymbol{W}_{kg}\boldsymbol{h}_j^k$$



C.1 Global Graph Linking

To learn the linking between each **query** node and the global structure of the **Key-Graph**, we calculate the **global attention score**:



(e.g., 1st node in **Query-Graph**)

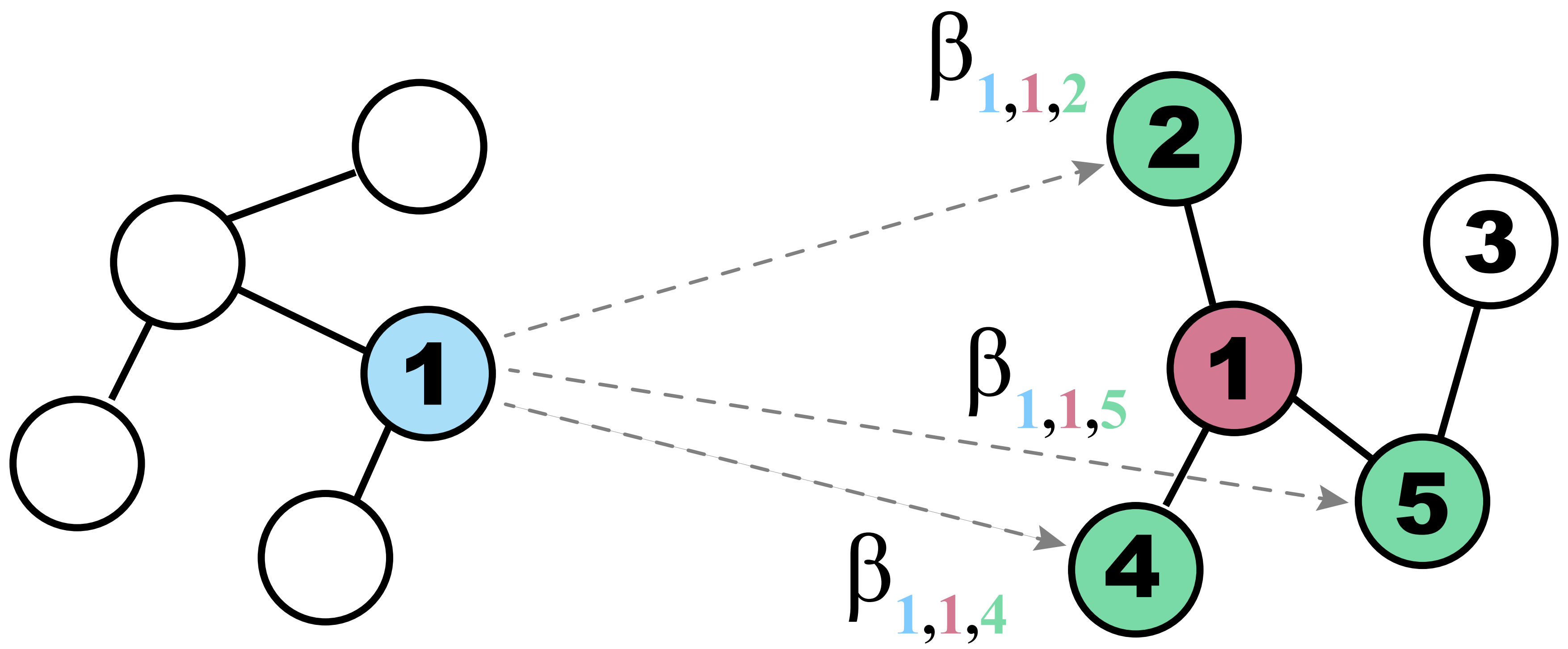
Learned Relation Feature

$$s_{i,j} = \sigma\left(\boldsymbol{h}_i^q \boldsymbol{W}_q \left(\boldsymbol{h}_j^k + \boldsymbol{R}_{ij}^E\right)^T\right)$$

$$\alpha_{i,j} = \mathrm{softmax}_j \left\{s_{i,j}\right\}$$

C.2 Local Graph Linking

In this phase, the **query** node will calculate the **local attention score** with the **neighbor** nodes of the **key** node cross dual-graph:



(e.g., 1st node in **Query-Graph** and 1st node in **Key-Graph**)

Learned Relation Feature

$$o_{i,j,t} = \sigma\left(\boldsymbol{h}_i^q \boldsymbol{W}_{nq}\left(\boldsymbol{h}_t^k + \boldsymbol{R}_{it}^E\right)^T\right)$$

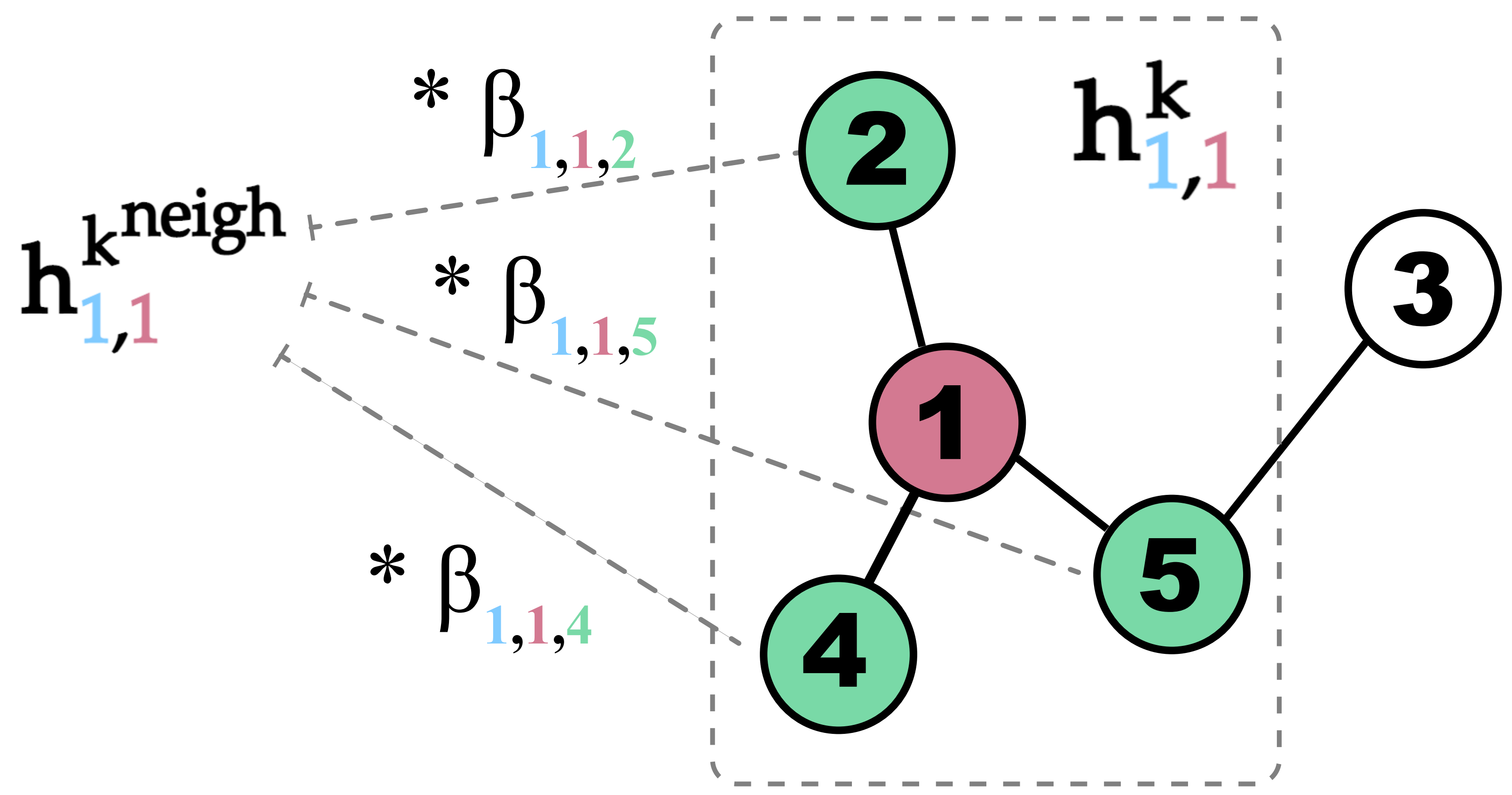
$$\beta_{i,j,t} = \operatorname{softmax}_t\left\{o_{i,j,t}\right\}\left(t \in \mathcal{N}_j\right)$$

Neighbors of j-th **Key** Node

C.3 Dual-Graph Aggregation Mechanism

Aggregate the **neighbor** information with the **local attention score**:

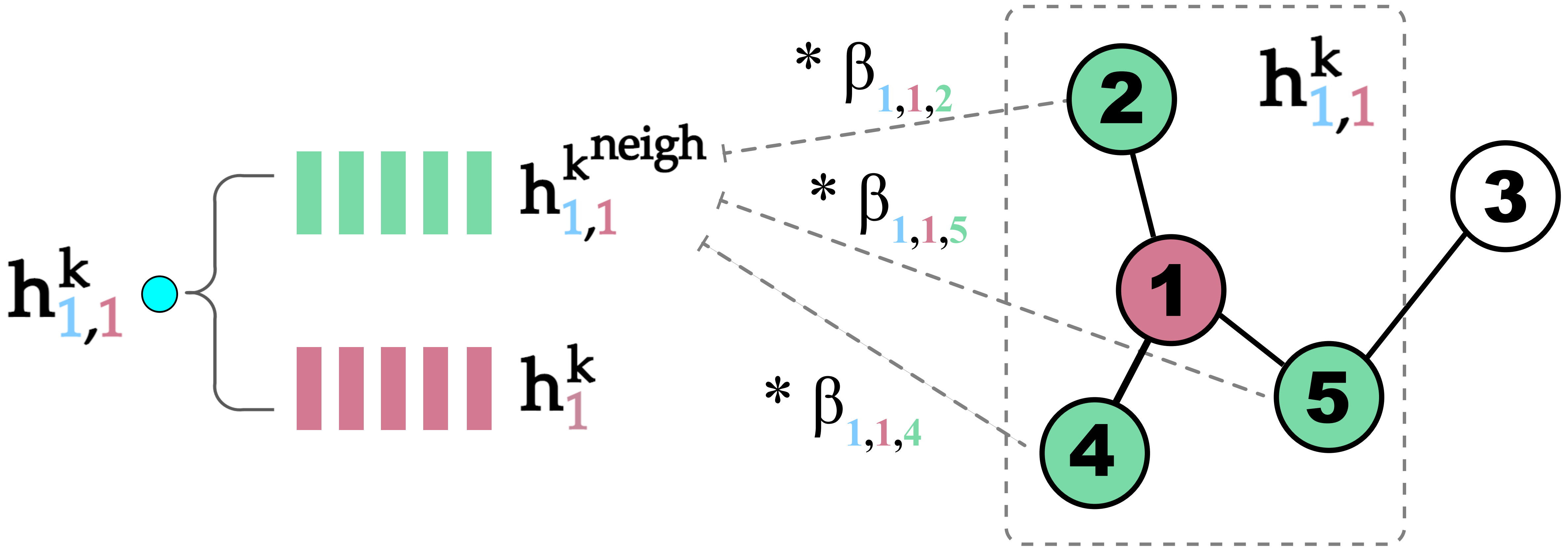
Neighbor Context Vector ⬅ $$\boldsymbol{h}_{i,j}^{k^{\text{neigh}}} = \sum_{t=1}^{T} \beta_{i,j,t} \boldsymbol{h}_t^k,$$



C.3 Dual-Graph Aggregation Mechanism

Aggregate the **neighbor** information with the **local attention score**:

Neighbor Context Vector ← $\boldsymbol{h}_{i,j}^{k^{\text{neigh}}} = \sum_{t=1}^{T} \beta_{i,j,t} \boldsymbol{h}_t^k,$



Apply a **gate** function to extract essential features among the **key** node self and the **neighbor** information:

$$\boldsymbol{h}_{i,j}^{k^{\text{self}}} = \boldsymbol{h}_j^k$$

$$\text{gate}_{i,j} = \theta\left(\boldsymbol{W}_{ng}\left[\boldsymbol{h}_{i,j}^{k^{\text{self}}}; \boldsymbol{h}_{i,j}^{k^{\text{neigh}}}\right]\right)$$

j-th **Key** Node Neighbor-aware Feature toward i-th **Query** Node ← $\boldsymbol{h}_{i,j}^{k} = (1 - \text{gate}_{i,j}) * \boldsymbol{h}_{i,j}^{k^{\text{self}}} + \text{gate}_{i,j} * \boldsymbol{h}_{i,j}^{k^{\text{neigh}}}$

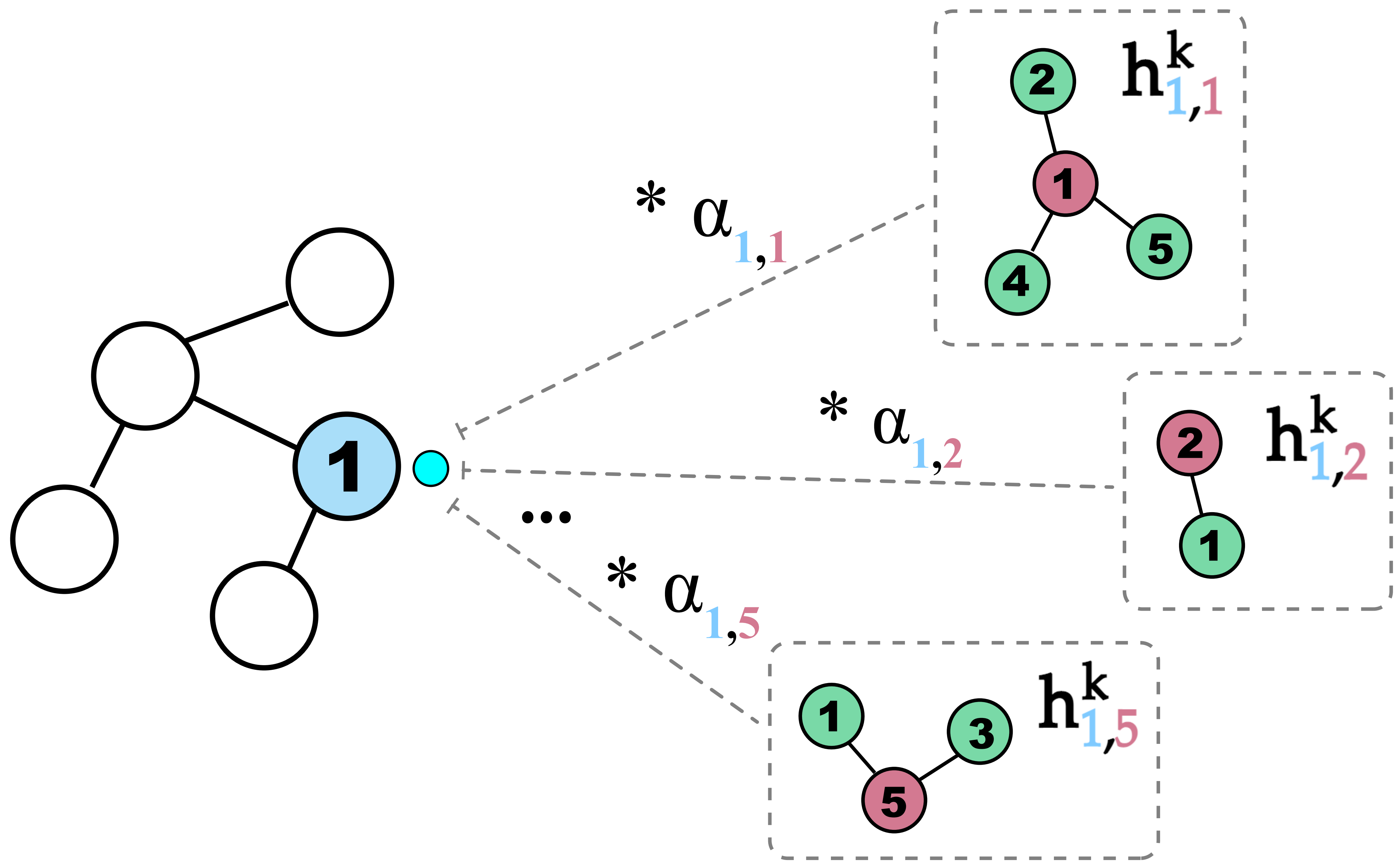
C.3 Dual-Graph Aggregation Mechanism

Finally, each query node aggregates the structure-aware information from all key nodes with the **global attention score** (Step 1):

$$\boldsymbol{h}_i^{q^{\text{new}}} = \sum_{j=1}^{n} \alpha_{i,j} \left(\boldsymbol{h}_{i,j}^{k} + \boldsymbol{R}_{ij}^{E}\right)$$

$$\text{gate}_i = \theta\left(\boldsymbol{W}_{\text{gate}}\left[\boldsymbol{h}_i^{q}; \boldsymbol{h}_i^{q^{\text{new}}}\right]\right)$$

$$\boldsymbol{h}_i^{q^{\text{Aggr}}} = (1 - \text{gate}_i) * \boldsymbol{h}_i^{q} + \text{gate}_i * \boldsymbol{h}_i^{q^{\text{new}}}$$



We can obtain the final query node representation with the structure-aware information of the Key-Graph.

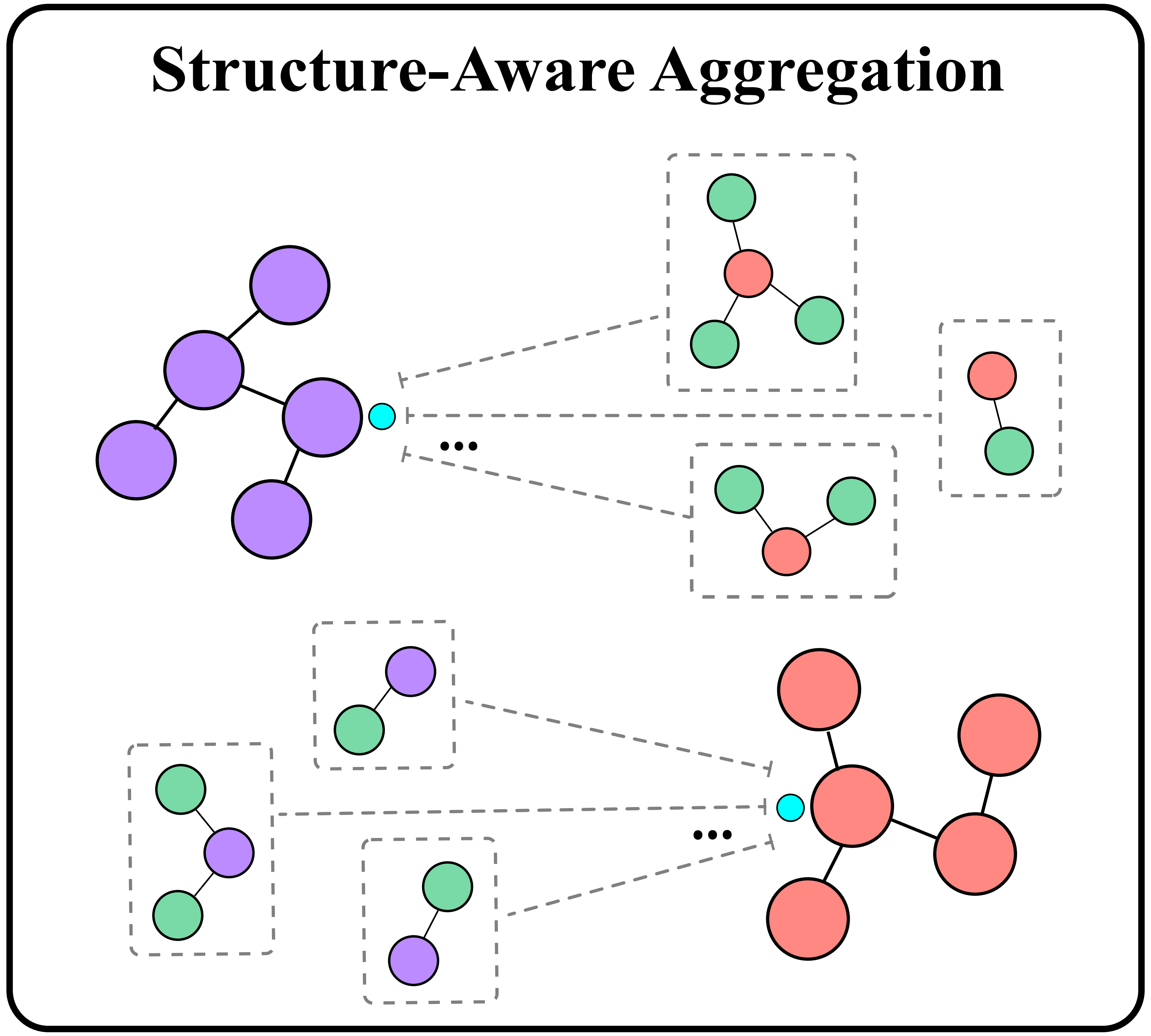
C. Structure-Aware Aggregation

Update **Question-Graph** $\mathcal{G}_Q$:

$$\mathcal{G}_Q^{Aggr} = \mathrm{GraphAggr}(\mathcal{G}_Q, \mathcal{G}_S)$$

Update **Schema-Graph** $\mathcal{G}_S$:

$$\mathcal{G}_S^{Aggr} = \mathrm{GraphAggr}(\mathcal{G}_S, \mathcal{G}_Q)$$



Learn the question-schema linking on **Local Structure Level**, instead of on node-level.

Experiments: Spider Datasets

- The most challenging benchmark on **cross-domain** Text-to-SQL.
- Contain 9 traditional specific-domain dataset, e.g., ATIS, GeoQuery.
- **Unseen** databases in the test set.
- Participants must submit the models (only two) to obtain the test accuracy for the official **non-released** test set.

Experiments: Results

At the time of writing, our best model has achieved the **3rd** on the overall leaderboard.

Approach	Dev	Test	Approach	Dev	Test
GNN [3]	40.7	39.4	RATSQL-HPFT + BERT-large	69.3	64.4
Global-GNN [2]	52.7	47.4	YCSQL + BERT-large	-	65.3
IRNet v2 [11]	55.4	48.5	DuoRAT + BERT-large [24]	69.4	65.4
RATSQL [27]	62.7	57.2	RATSQL + BERT-large [27]	69.7	65.6
SADGA	**65.6**	-	**SADGA + BERT-large**	**71.6**	**66.7**
EditSQL + BERT-base [36]	57.6	53.4	ShadowGNN + RoBERTa [5]	72.3	66.1
GNN + Bertrand-DR [15]	57.9	54.6	RATSQL + STRUG [9]	72.6	68.4
IRNet v2 + BERT-base [11]	63.9	55.0	RATSQL + GraPPa [34]	**73.4**	69.6
RATSQL + BERT-base [27]	65.8	-	RATSQL + GAP [25]	71.8	69.7
SADGA + BERT-base	**69.0**	-	**SADGA + GAP**	73.1	**70.1**

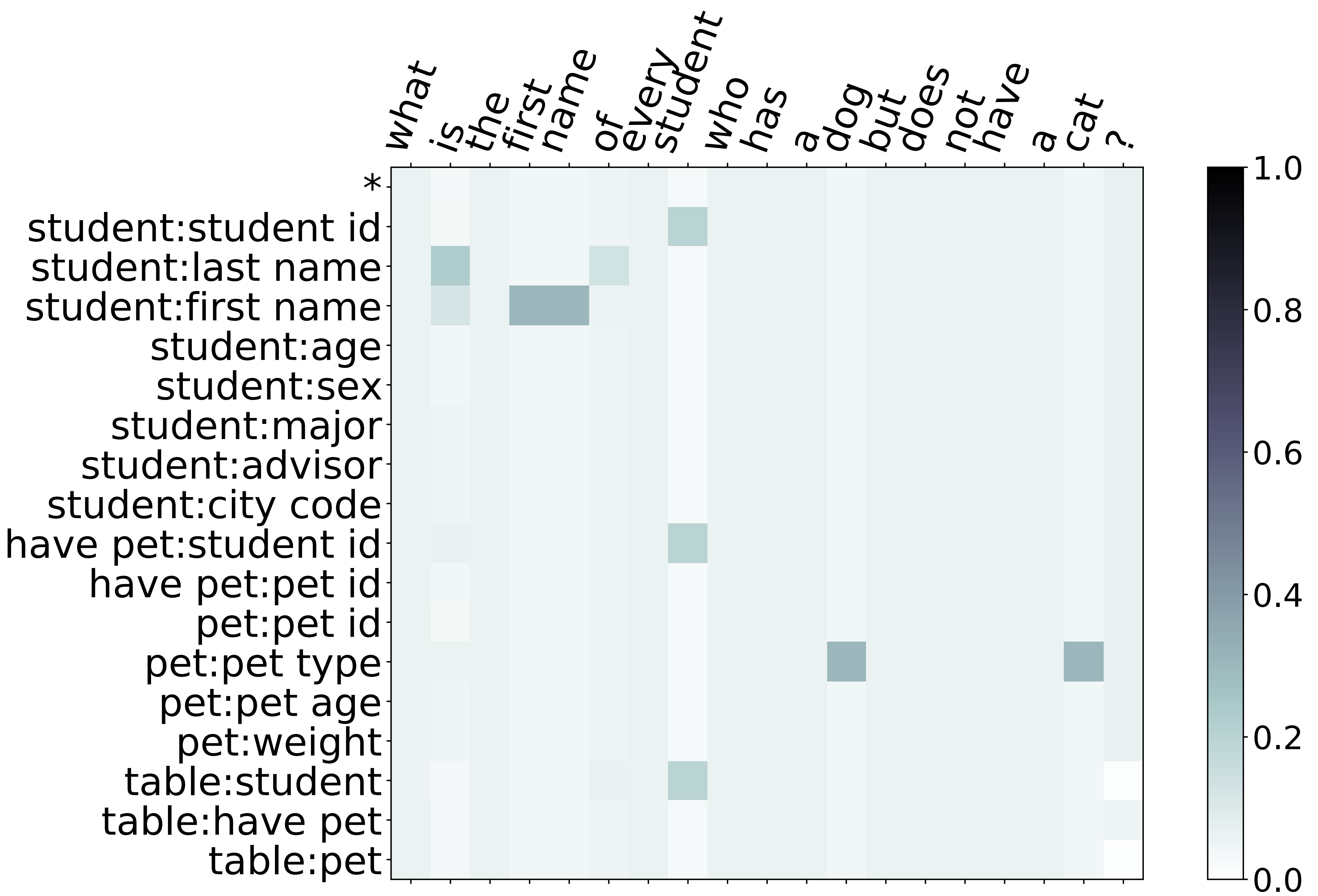
Rank	Model	Dev	Test
1 Nov 19, 2020	DT-Fixup SQL-SP + RoBERTa (DB content used) *Borealis AI* (Xu et al., ACL'21)	75.0	**70.9**
2 Nov 19, 2020	RAT-SQL + GraPPa + Adv (DB content used) *Anonymous*	75.5	70.5
3 Nov 19, 2020	SADGA + GAP (DB content used) *Anonymous*	73.1	70.1
4 Dec 25, 2020	RATSQL + GraPPa + GP (DB content used) *OCFT Gamma Big Data Lab* (Zhao et al.,'21)	72.8	69.8
4 Sep 08, 2020	RATSQL + GAP (DB content used) *University of Waterloo & AWS AI Labs* (Shi et al., AAAI'21) code	71.8	69.7
4 Aug 18, 2020	RATSQL + GraPPa (DB content used) *Yale & Salesforce Research* (Yu et al., ICLR'21) code	73.4	69.6
4 Mar 10, 2021	SmBoP + GraPPa (DB content used) *Tel-Aviv University & Allen Institute for AI* (Rubin and Berant, NAACL'21) code	74.7	69.5
7 Nov 20, 2020	RAT-SQL + STRUG (DB content used) *Microsoft Research & OSU* (Deng et al., NAACL '21)	72.6	68.4

Experiments: Ablation Study

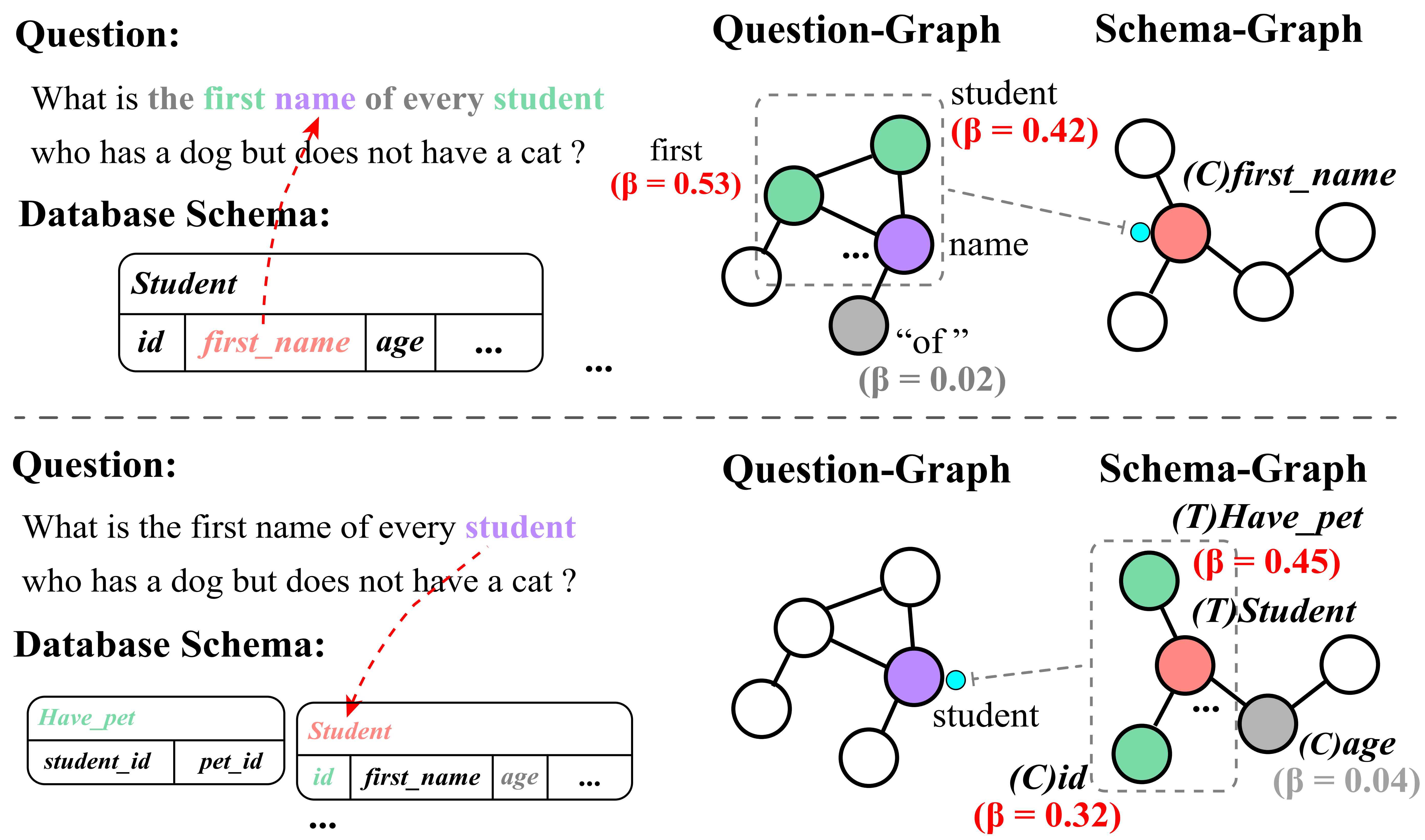
Model	Easy	Medium	Hard	Extra Hard	All
SADGA	**80.6**	**67.7**	**57.6**	**45.8**	**65.6**
w/o Local Graph Linking	83.5(+2.9)	64.8(-2.9)	53.4(-4.2)	38.6(-7.2)	63.2(-2.4)
w/o Structure-Aware Aggregation	83.5(+2.9)	62.1(-5.6)	55.2(-2.4)	42.2(-3.6)	62.9(-2.7)
w/o GraphAggr$(\mathcal{G}_S, \mathcal{G}_Q)$	83.1(+2.5)	64.1(-3.6)	52.3(-5.3)	40.4(-5.4)	62.9(-2.7)
w/o GraphAggr$(\mathcal{G}_Q, \mathcal{G}_S)$	79.0(-1.6)	63.7(-4.0)	50.0(-7.6)	41.6(-4.2)	61.5(-4.1)
Cross-Graph Linking in Dual-Graph Encoding	82.3(+1.7)	63.7(-4.0)	51.1(-6.5)	45.2(-0.6)	63.1(-2.5)
w/o Relation Node (replace with edge types)	79.4(-1.2)	63.5(-4.2)	54.6(-3.0)	40.4(-5.4)	62.1(-3.5)
w/o Global Pooling (Eq. 3 and Eq. 4)	82.7(+2.1)	64.3(-3.4)	54.0(-3.6)	41.6(-4.2)	63.5(-2.1)
w/o Aggregation Gate (Eq. 8, $\text{gate}_{i,j} = 0.5$)	81.9(+1.3)	60.1(-7.6)	54.6(-3.0)	40.4(-5.4)	61.2(-4.4)
w/o Relation Feature in Aggregation ($\boldsymbol{R}^E_{ij}$)	79.4(-1.2)	64.3(-3.4)	54.6(-3.0)	41.6(-4.2)	62.7(-2.9)
SADGA + BERT-base	**85.9**	**71.7**	**58.0**	**47.6**	**69.0**
w/o Local Graph Linking	85.5(-0.4)	69.5(-2.2)	54.0(-4.0)	42.8(-4.8)	66.4(-2.6)
w/o Structure-Aware Aggregation	85.9(-0)	68.8(-2.9)	57.5(-0.5)	41.0(-6.6)	66.5(-2.5)

Experiments: Case Study

"What is the first name of every student who has a dog but does not have a cat? "



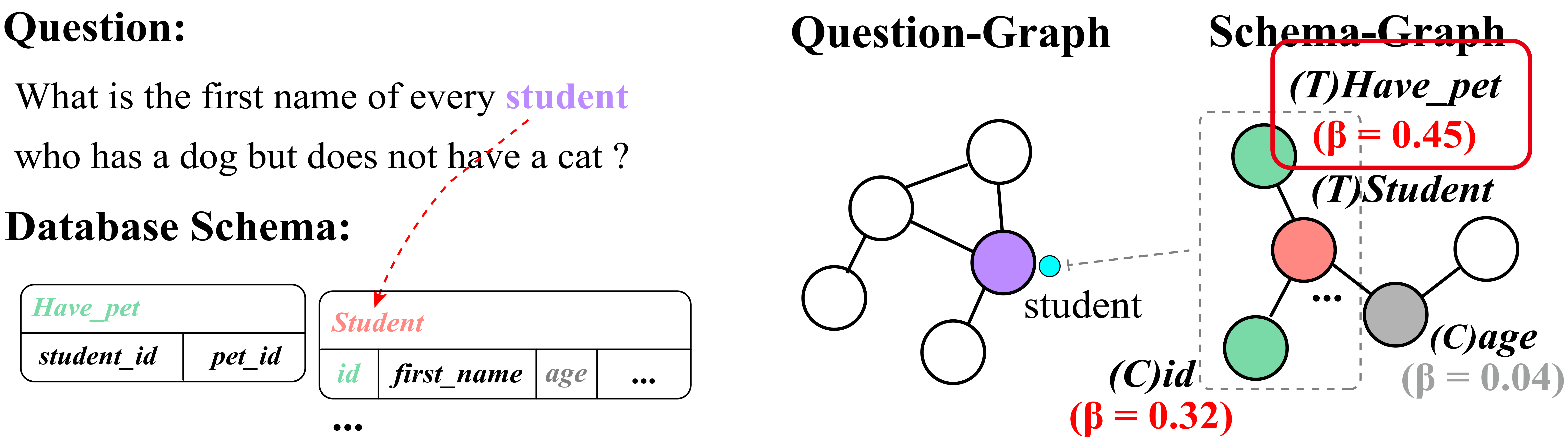
Alignment between question words
and tables/columns on **Global Graph Linking**



Analysis on **Local Graph Linking**

Experiments: Case Study

“What is the first name of every student who has a dog but does not have a cat? ”



Analysis on **Local Graph Linking**

Conclusion

- A **Structure-Aware Dual Graph Aggregation Network (SADGA)** for cross-domain Text-to-SQL task.

- **SADGA:** (i) a unified graph encoding for both question and schema,
  (ii) a graph aggregation approach to consider the **global** and **local** structure information of dual graph.

- Detailed experiments and case studies show the effectiveness of SADGA.

- We will extend SADGA to other **heterogeneous graph tasks**.

Thanks for Listening!

Welcome to QA for questions!

Our code is available at: https://github.com/DMIRLAB-Group/SADGA