FIETA: Few-shot Inductive Knowledge Graph Completion via Task-Aware Graph Neural Networks

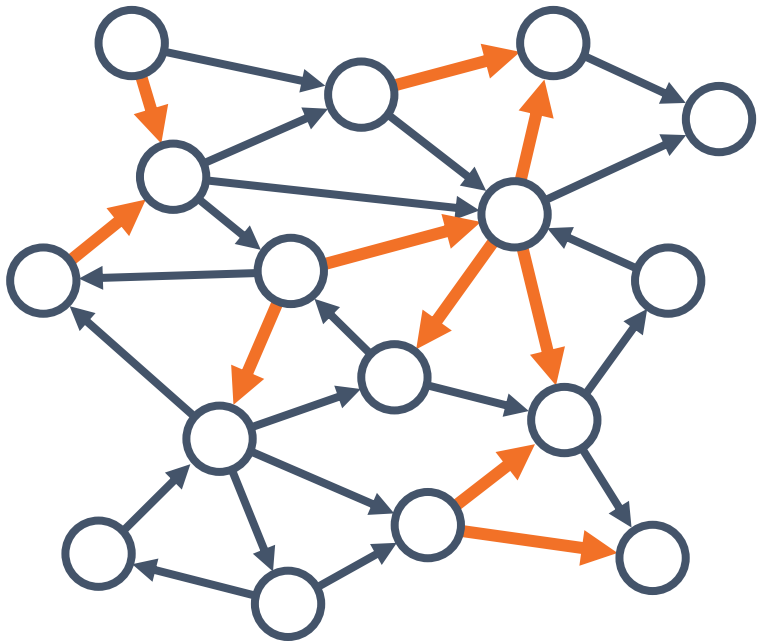
Chanyoung Chung[1], Minsung Hwang[1], and Joyce Jiyoung Whang[2*]
[1] School of Computing, KAIST, [2] Department of AI Computing, KAIST
[*] Corresponding Author

The 35th ACM International Conference on Information and Knowledge Management (CIKM 2026)

KAIST BDILab
BIG DATA INTELLIGENCE

01 Knowledge Graph Completion (KGC)

- Knowledge graph completion aims to seek missing relationships between entities.
- Most existing KGC methods assume that **sufficient training examples are available for all relations**.
  - They often struggle with **few-shot relations**, where **only a limited number of examples are provided**.



Knowledge Graph

(Rome, located in, Italy)
(Paris, located in, France)
(Berlin, located in, Germany)
(Seoul, located in, Korea)
(Tokyo, located in, Japan)
(Sydney, located in, Australia)
(Madrid, located in, Spain)
⋮

Training Examples

(New York, located in, ?)
→ United States

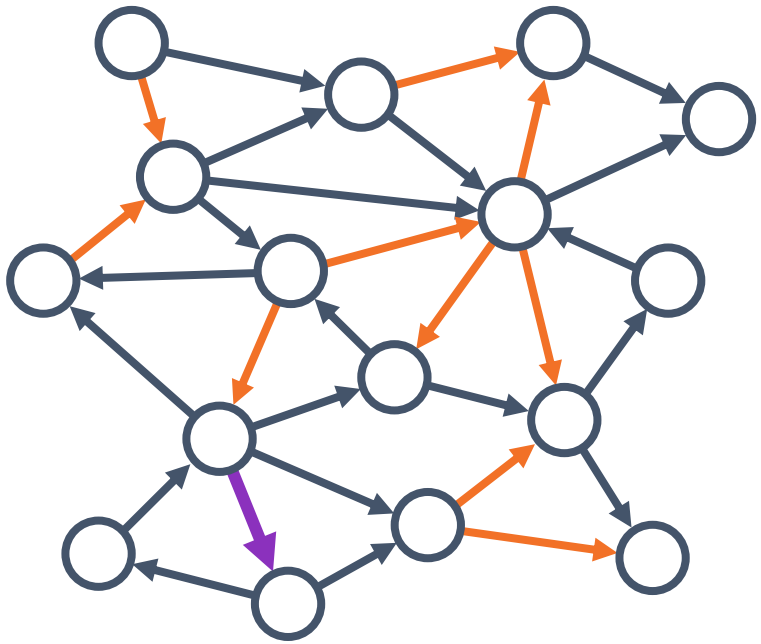
(Beijing, located in, ?)
→ China

(Hawaii, located in, ?)
→ United States

Queries

01 Few-shot KGC

- Few-shot KGC aims to predict missing triplets involving few-shot relations.
  - Each few-shot relation is associated with only a few examples of the relation, called support triplets.
  - Each few-shot task corresponds to predicting triplets for a few-shot relation.



Knowledge Graph

(Colosseum, landmark of, Italy)

Training Examples
(= Support Triplets)

(Taj Mahal, landmark of, ?)
→ India

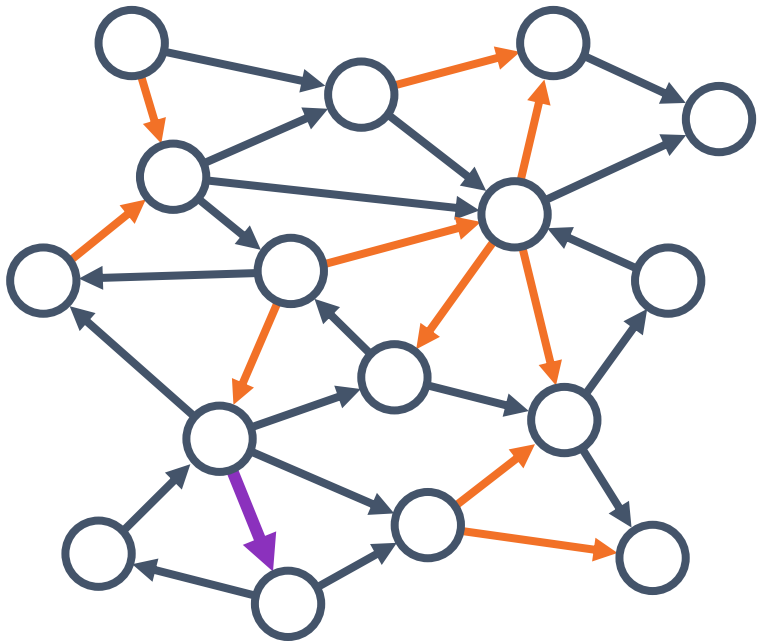
(Big Ben, landmark of, ?)
→ UK

(Sphinx, landmark of, ?)
→ Egypt

Queries

01 Few-shot KGC

- To deal with few-shot relations, most few-shot KGC methods use the learned entity and relation embeddings to obtain representations specialized for each few-shot task.
  - They require all entities and relations in KGs to be observed, and are thus limited to transductive scenarios.



Knowledge Graph

(Eiffel Tower, landmark of, France)
(Colosseum, landmark of, Italy)
(Tokyo Tower, landmark of, Japan)

Training Examples
(= Support Triplets)

(Taj Mahal, landmark of, ?)
→ India

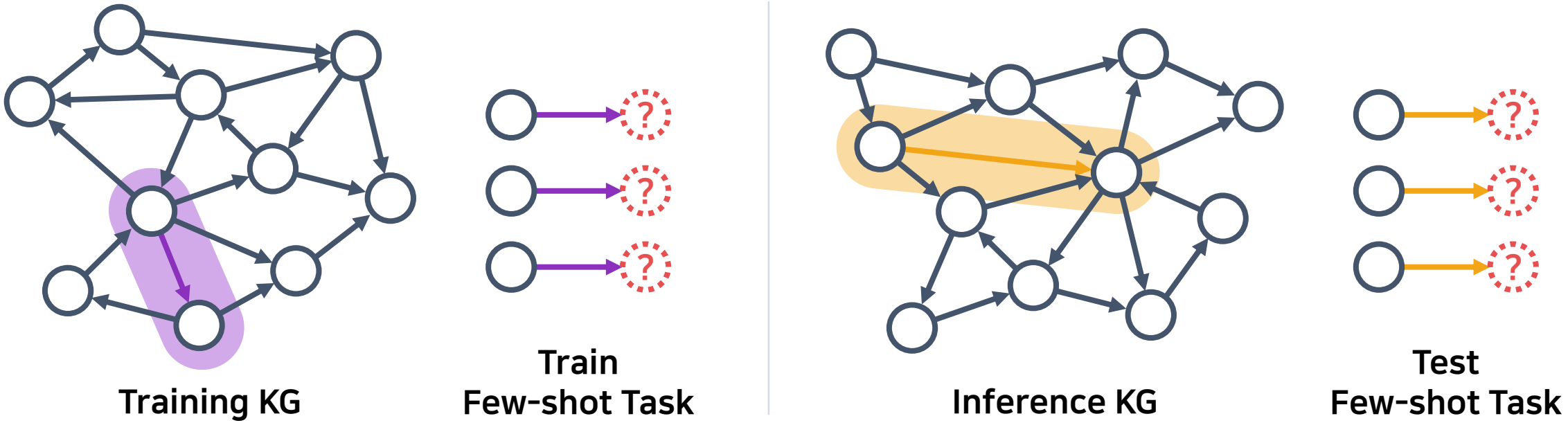
(Big Ben, landmark of, ?)
→ UK

(Sphinx, landmark of, ?)
→ Egypt

Queries

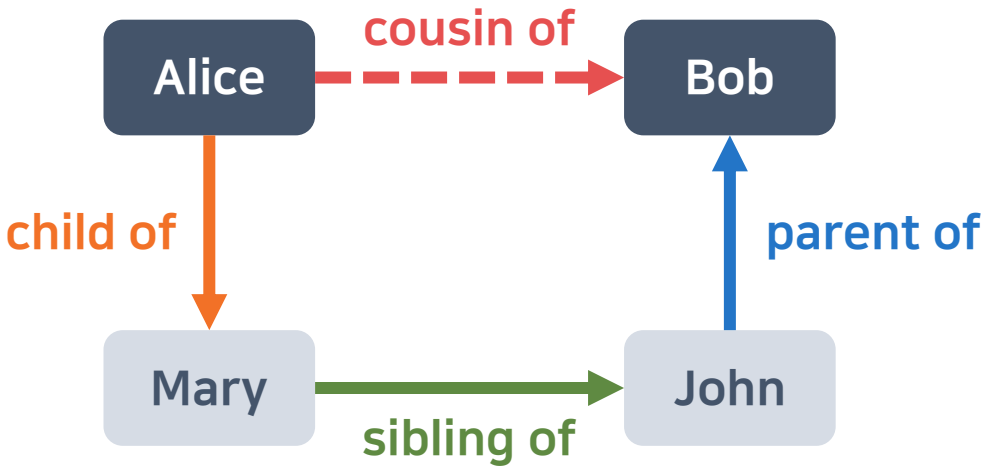
01 Few-shot Inductive KGC

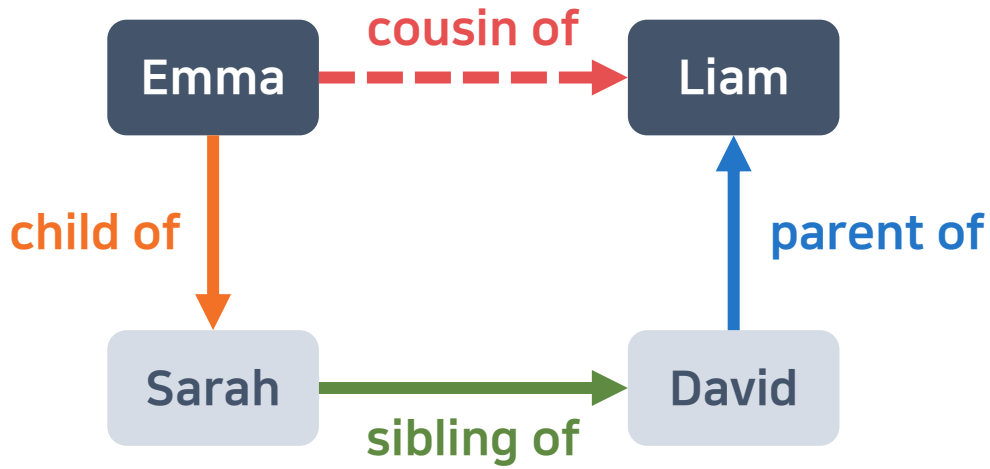
- **Few-shot inductive KGC** further extends few-shot KGC to inductive settings.
  - A KG provided at inference time contains **entities and relations unobserved during training**.
- Few-shot inductive KGC aims to predict missing triplets involving few-shot relations in KGs with new entities and relations appearing at inference time.



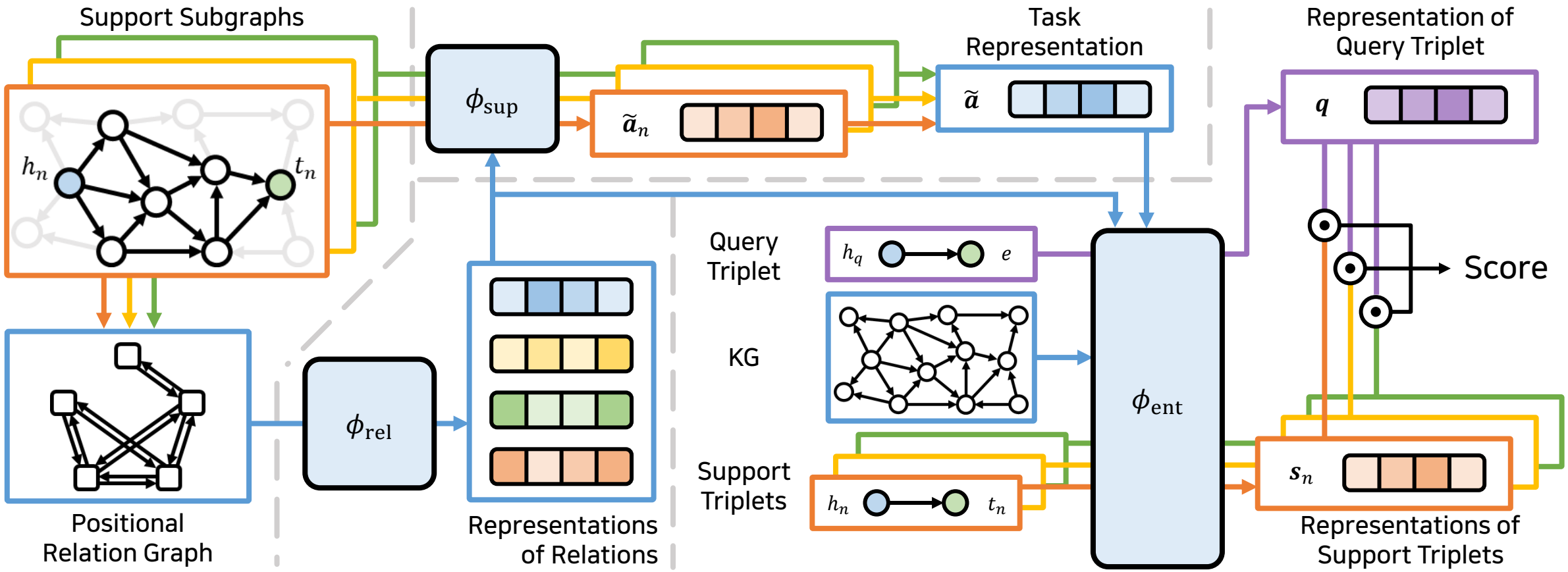
01 Motivation

- In few-shot inductive KGC, embeddings of unseen entities and relations cannot be learned.
  - Thus, support triplets alone provide limited information about few-shot relations.
- However, **local structures surrounding support triplets** can provide meaningful information.
  - For example, two support triplets may share the same relational patterns connecting their head and tail entities.
- Our Approach: Capture **structural patterns surrounding support triplets without relying on learned entity and relation embeddings**, and leverage them to predict missing entities in few-shot queries.



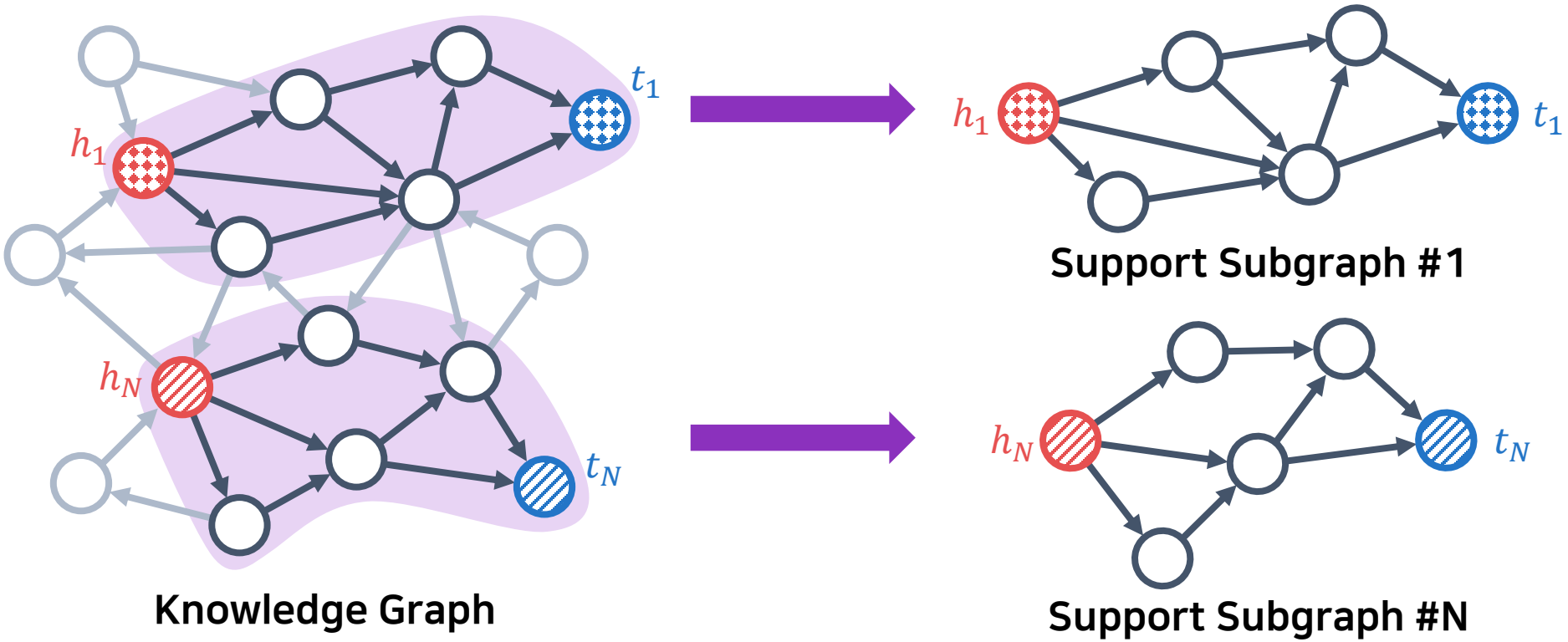


02 Overview of FIETA



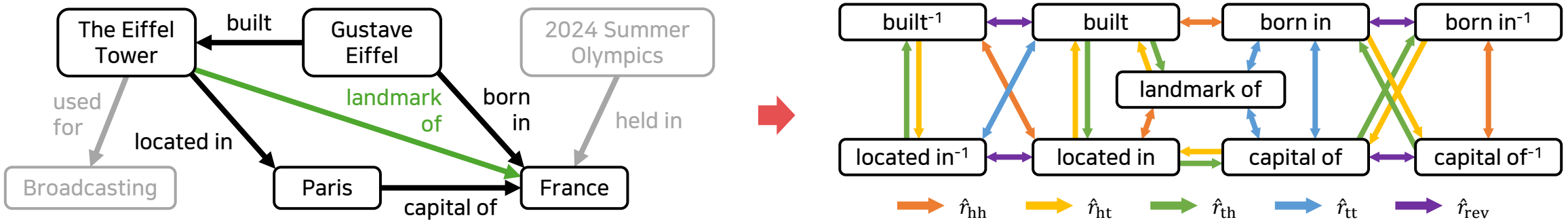
02 Extracting Support Subgraphs

- For each support triplet (h_n, r, t_n), extract a **support subgraph** $\tilde{G}_n$ capturing its local structure.
  - **Collect all triplets along paths connecting h_n and t_n**, with a maximum path length of L_{sup}.
  - Add reverse relations and triplets to capture bidirectional connectivity.



02 Constructing Positional Relation Graphs

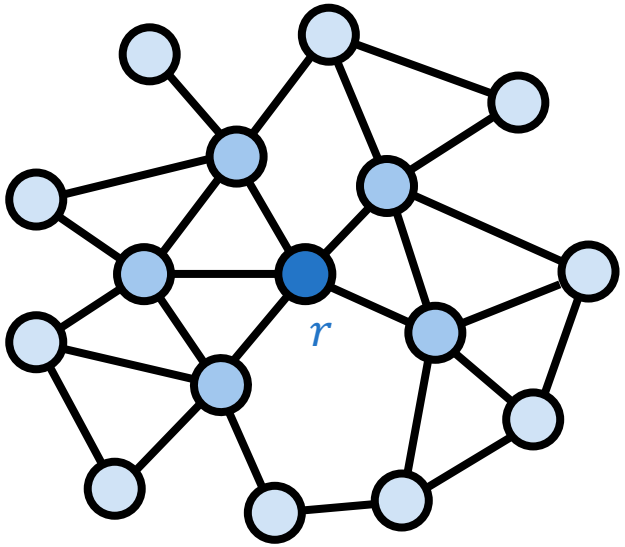
- Construct a **positional relation graph** $\hat{G}_r$ reflecting **task-centric connectivity patterns among relations**.
- Represent relations as nodes and define edges based on their **connectivity in support subgraphs**.
  - $(r_1, \hat{r}_{\mathrm{xy}}, r_2)$: An entity in position x of r_1 and an entity in position y of r_2 are the same.
    - Both x and y can be either h (head) or t (tail).
  - $(r_1, \hat{r}_{\mathrm{rev}}, r_2)$: r_1 and r_2 connect the same pair of entities in opposite directions, i.e., (a, r_1, b) and (b, r_2, a).



Support Subgraph **Positional Relation Graph**

02 Task-centric Modeling of Relations

- First, obtain **representations of relations optimized for each few-shot task**.
- Define a **task-aware GNN ϕ_{rel} on $\hat{G}_r$**, which iteratively propagates messages originating from r.
  - Distinguish messages from r by setting the initial representation of r with $\mathbf{1} \in \mathbb{R}^d$ and $\mathbf{0} \in \mathbb{R}^d$ otherwise.
- The final representations reflect **how relations are connected in the context of the target task**.



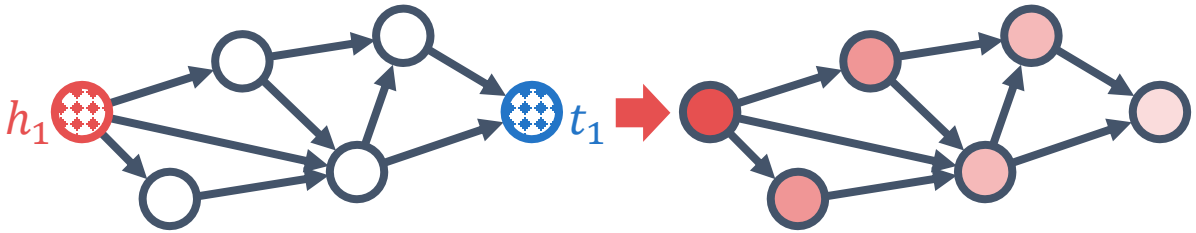
$$\boldsymbol{r}_i^{(l+1)} = \sigma\left(\sum_{(r_j,\hat{r}_k)\in\hat{\mathcal{N}}_r(r_i)} \hat{\alpha}_{ijk}^{(l)} \widehat{\boldsymbol{W}}_{\text{msg}}^{(l)} \left[\boldsymbol{r}_j^{(l)}; \hat{\boldsymbol{r}}_k\right]\right)$$

Attention weight of $(r_j, \hat{r}_k, r_i)$ in the l-th layer

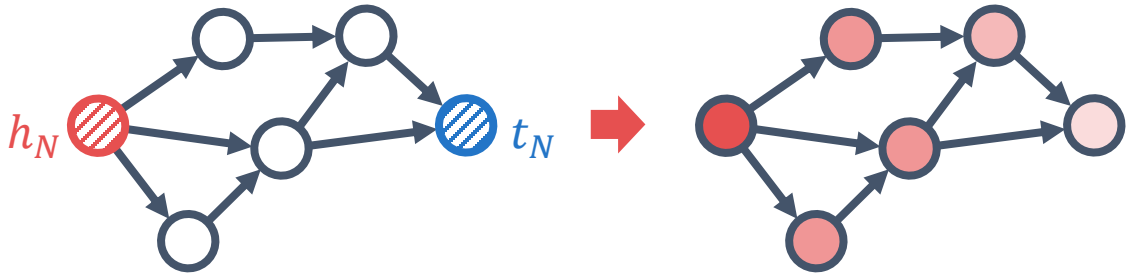
Message from $(r_j, \hat{r}_k, r_i)$ in the l-th layer

02 Computing Task Representations

- Then, obtain a **task representation** from the local structures of support triplets.
  - The task representation will later be **used to adjust the importance of triplets for each few-shot task**.
- Define a **task-aware GNN ϕ_{sup}** that computes entity representations on each support subgraph $\tilde{G}_n$.
  - Distinguish messages from h_n by setting the initial representation of h_n with $\mathbf{1} \in \mathbb{R}^d$ and $\mathbf{0} \in \mathbb{R}^d$ otherwise.



Support Subgraph #1



Support Subgraph #N

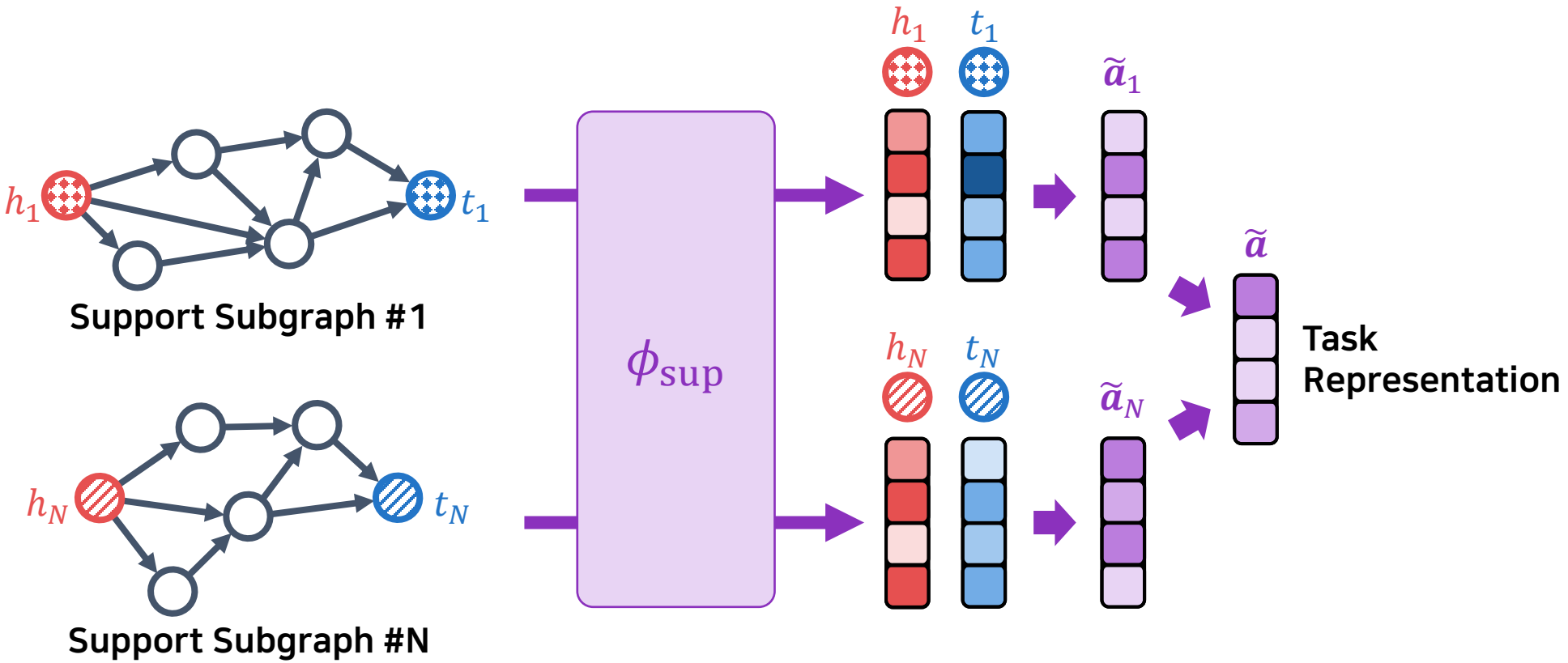
$$\tilde{\boldsymbol{v}}_i^{(l+1)} = \sigma\left(\sum_{(v_j,r_k)\in\tilde{\mathcal{N}}_n(v_i)} \tilde{\alpha}_{ijk}^{(l)} \tilde{\boldsymbol{W}}_{\text{msg}}^{(l)} \left[\tilde{\boldsymbol{v}}_j^{(l)}; \boldsymbol{r}_k\right]\right)$$

Attention weight of (v_j, r_k, v_i) in the l-th layer

Message from (v_j, r_k, v_i) in the l-th layer

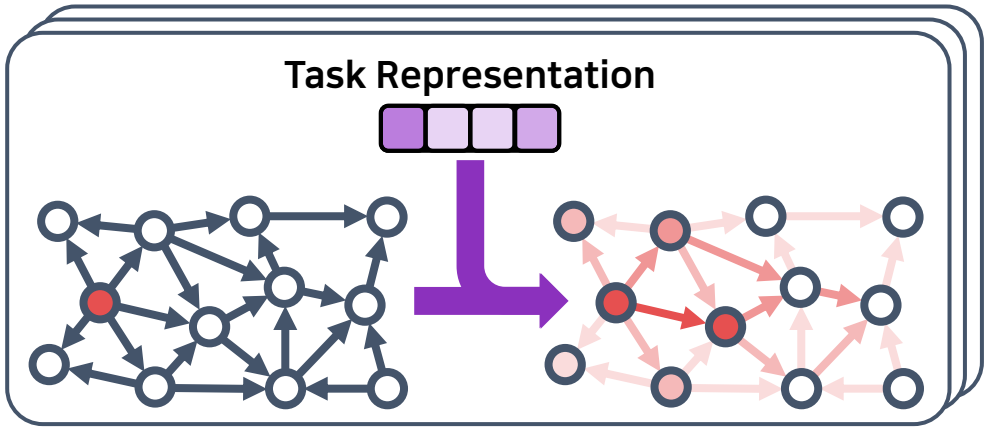
02 Computing Task Representations

- To compute a task representation, encode each support triplet (h_n, r, t_n).
- The task representation is obtained by **aggregating the representations of support triplets**.
  - The attention-based aggregation helps **mitigate the impact of potential noise in support triplets**.



02 Task-guided Message Passing on KGs

- Finally, define a **task-aware GNN ϕ_{ent}** that **prioritizes triplets relevant to the few-shot task**.
  - Compute entity representations associated with the target triplet (h, r, t) by setting the initial representation of h with $\mathbf{1} \in \mathbb{R}^d$ and $\mathbf{0} \in \mathbb{R}^d$ otherwise.
  - Leverage task representations in computing attention weights to **assign higher weights for key triplets**.



Task-guided Message Passing

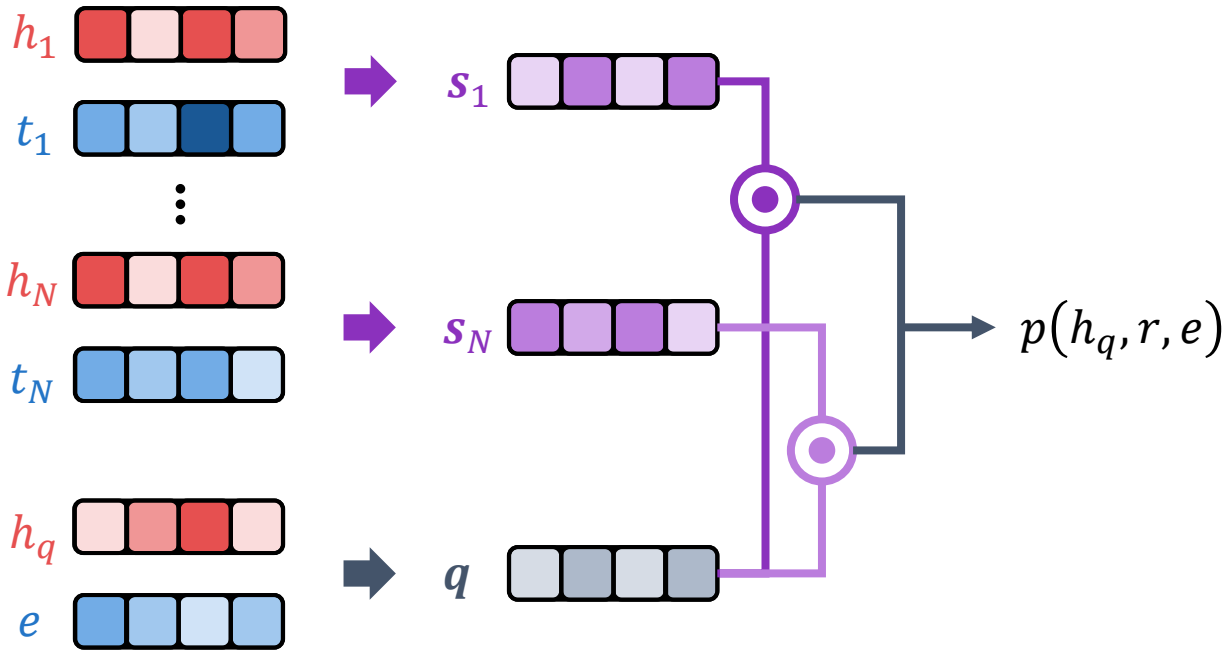
$$\boldsymbol{v}_i^{(l+1)} = \sigma\left(\sum_{(v_j, r_k) \in \mathcal{N}(v_i)} \alpha_{ijk}^{(l)} \boldsymbol{W}_{\text{msg}}^{(l)} \left[\boldsymbol{v}_j^{(l)}; \boldsymbol{r}_k\right]\right)$$

Attention weight of (v_j, r_k, v_i) in the l-th layer

Message from (v_j, r_k, v_i) in the l-th layer

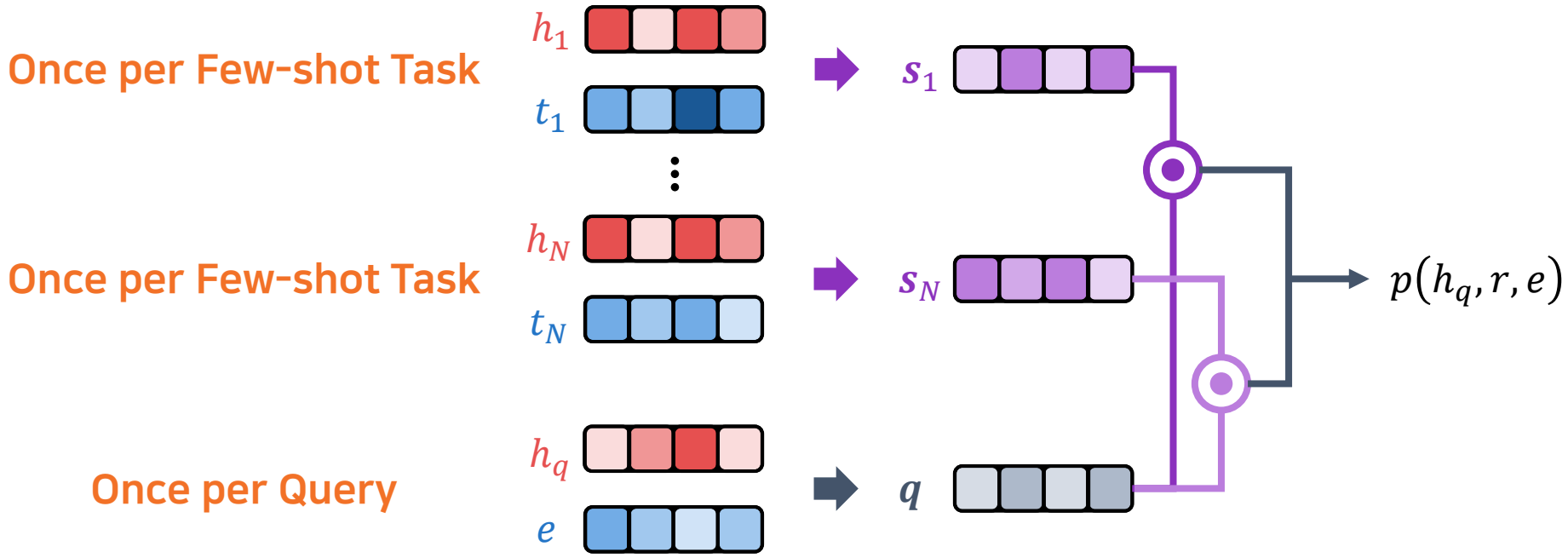
02 Matching-based Scoring

- With task-guided message passing, we obtain the following representations:
  - By running ϕ_{ent} for a support triplet (h_n, r, t_n), obtain **representations of h_n and t_n specific to (h_n, r, t_n)**.
  - By running ϕ_{ent} for a query $(h_q, r, ?)$, obtain **representations of h_q and each candidate entity e**.
- To compute the plausibility of a triplet (h_q, r, e), **measure its similarity to support triplets**.



02 Computational Cost of FIETA

- Given a few-shot task with N support triplets and Q queries, FIETA
  - Performs task-centric modeling of relations with **a single GNN call**.
  - Computes a task representation with N **GNN calls, one for each support triplet**.
  - Performs task-guided message passing with $N + Q$ **GNN calls, one for each support and query triplet**.
- **Relation and task representations are shared across all queries** within the same few-shot task.

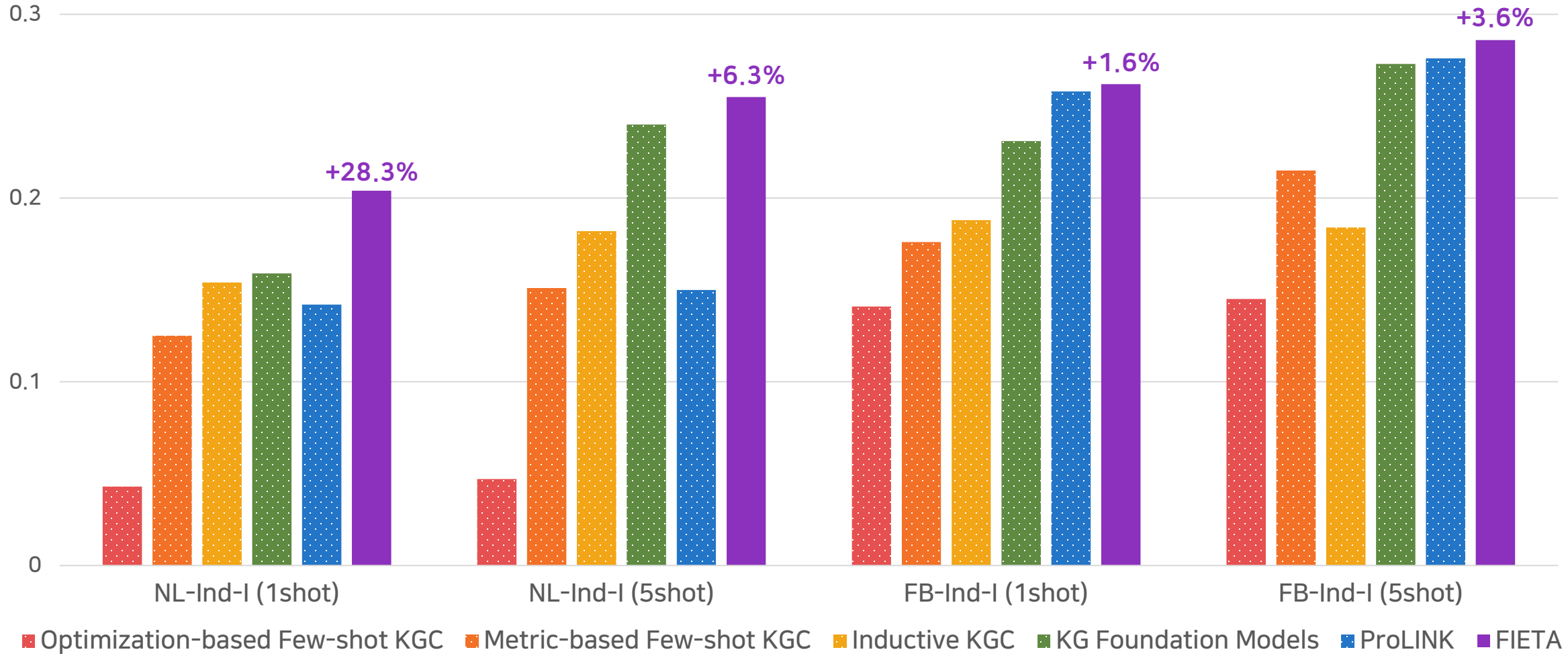


03 Experimental Settings

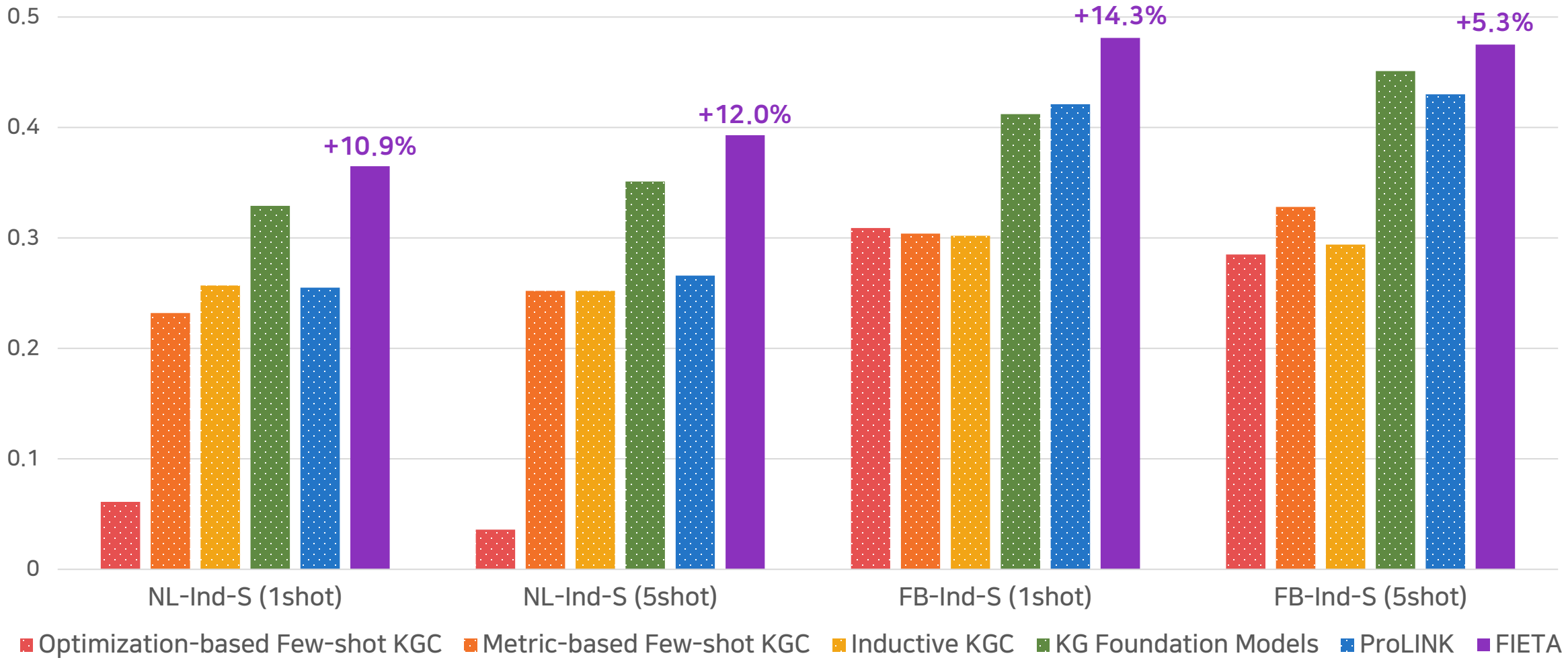
- Datasets
  - Existing few-shot KGC datasets are **not applicable to inductive settings**.
  - Construct six datasets for few-shot inductive KGC using NELL-One (NL) and FB15k237 (FB).
    - **NL/FB-Ind-T (Transductive)**: All relations in the inference KG are seen in the training KG.
    - **NL/FB-Ind-S (Semi-Inductive)**: Some relations in the inference KG are unseen in the training KG.
    - **NL/FB-Ind-I (Inductive)**: All relations in the inference KG are unseen in the training KG.
  - For all datasets, **entities of training and inference KGs are disjoint**.

- Baselines
  - **Few-shot KGC**: MetaR, GANA, HiRe, GMatching, FAAN, MetaP, P-INT, CIAN
  - **Inductive KGC**: Neural LP, DRUM, NBFNet, RED-GNN, A*Net, AdaProp, MaKEr, InGram
  - **KG Foundation Models**: ULTRA, KG-ICL
  - **Few-shot Inductive KGC**: ProLINK

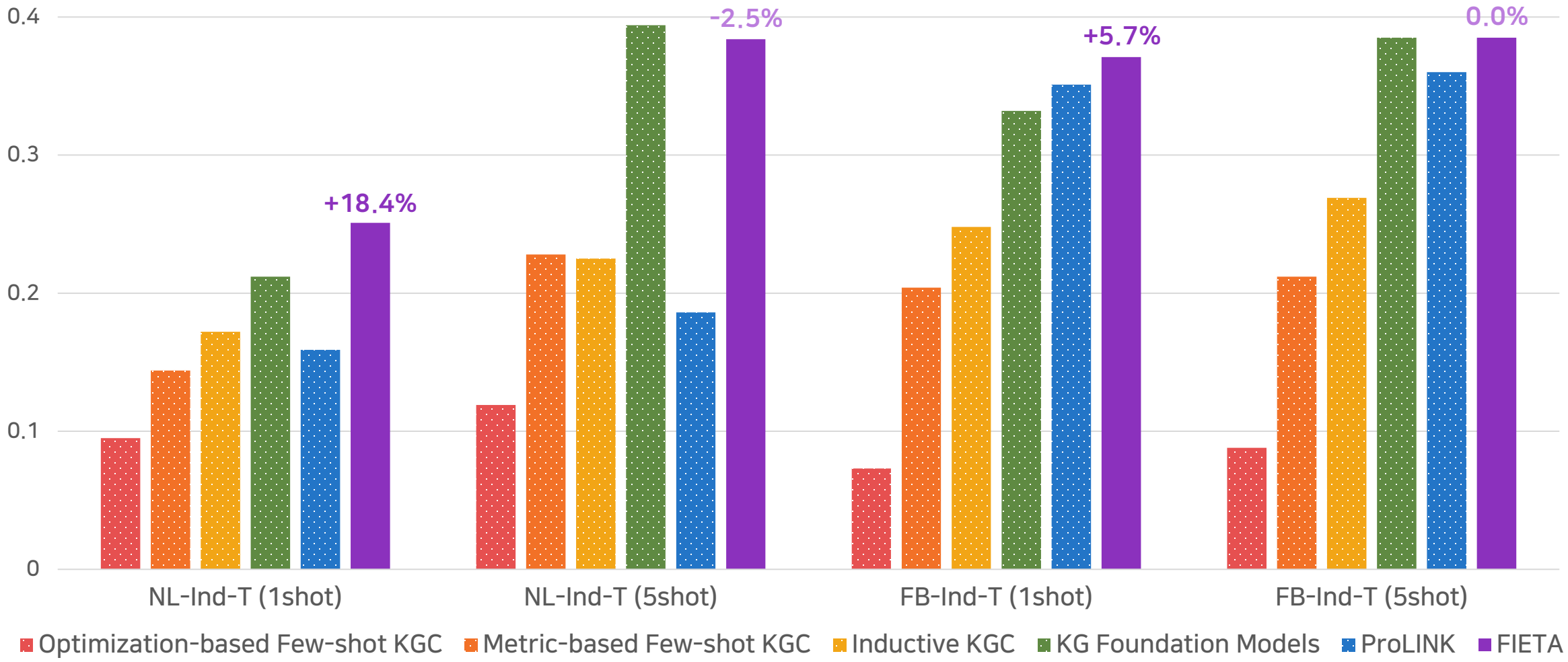
03 Inductive Inference for Relations



03 Semi-Inductive Inference for Relations

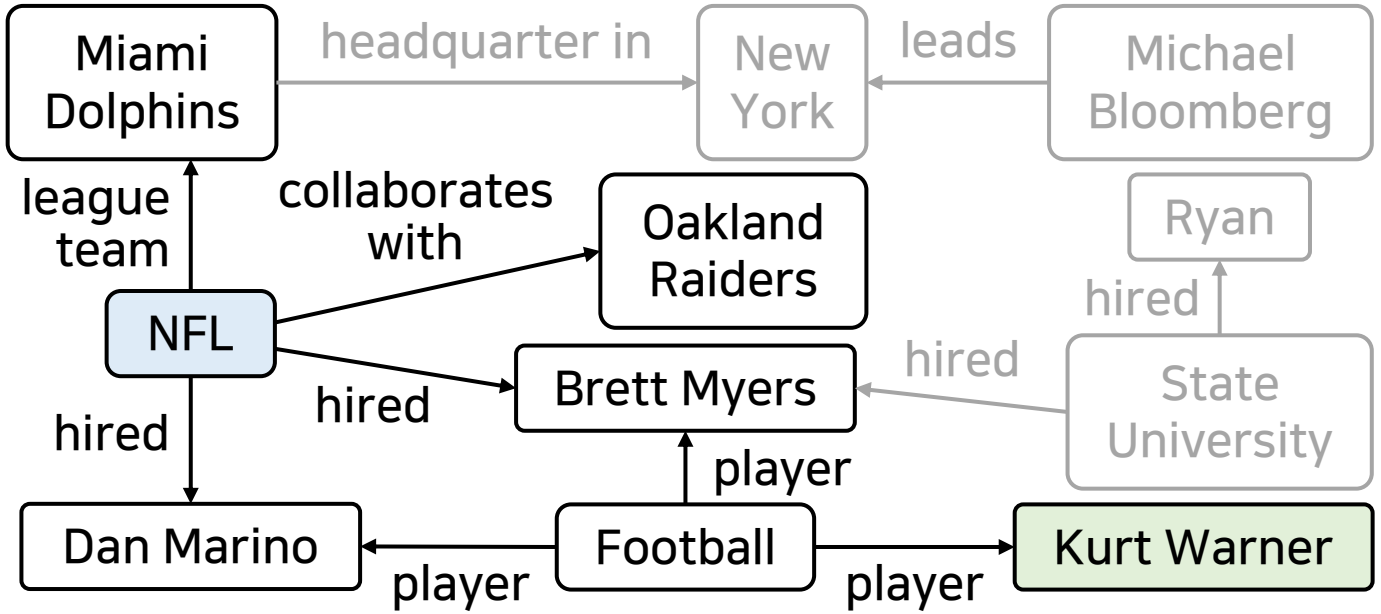


03 Transductive Inference for Relations



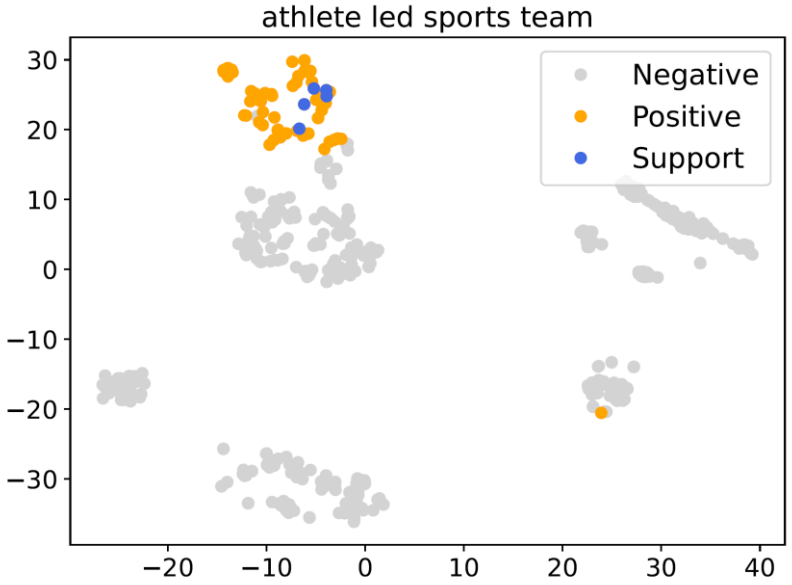
03 Attention Weights of Triplets

- Attention weight visualization of triplets for the query (NFL, league player, ?).
  - Triplets that are **crucial to the prediction gain high attention weights**.
  - Triplets that are **less important in prediction receive low attention weights**.
- Task-aware representations successfully guide the entity-level message passing.

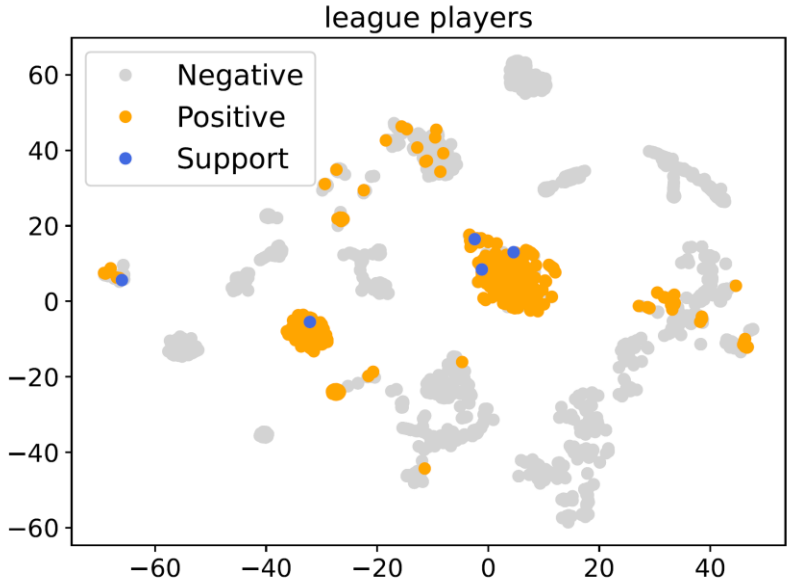


03 Representations of Triplets

- FIETA is designed to assign higher scores to positive triplets than negative ones.
  - We expect that the representations of positive triplets are closer to the representations of support triplets.
- Visualize the representations of support, positive, and negative triplets using t-SNE.
  - **The representations of positive triplets are positioned closer to support triplets than negative triplets.**



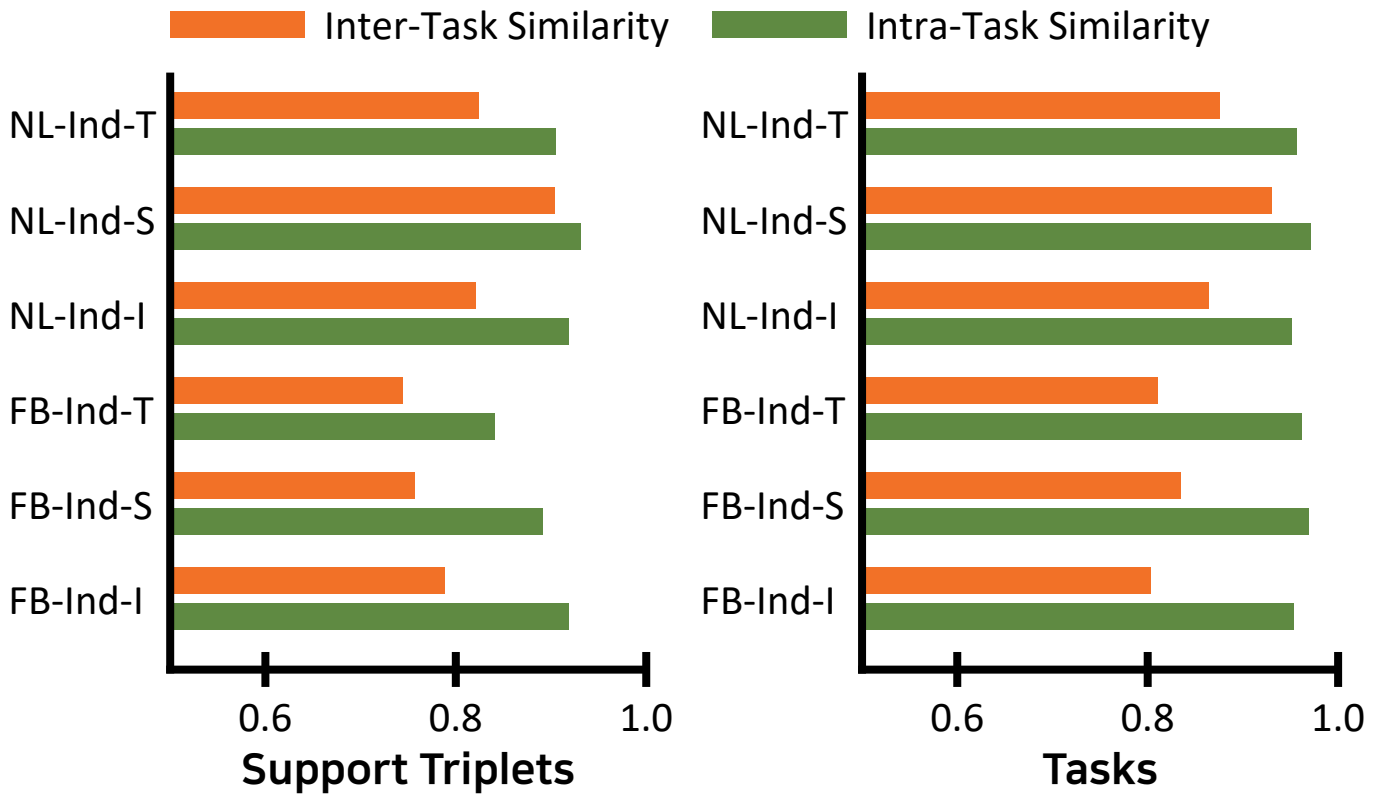
Task: athlete led sports team



Task: league players

03 Task Representations

- Expectations on task representations
  - **Support triplets within the same task are expected to be closer** to each other than to those from other tasks.
  - **Task representations of the same task are expected to be closer** to each other than to those of different tasks.



04 Conclusion

- Investigate **few-shot inductive KGC**, which aims to predict missing triplets involving **few-shot relations in KGs with new entities and relations** appearing at inference time.

- Propose **FIETA** that introduces **task-aware GNNs** to compute task-specific representations.
  - FIETA adaptively taking into account **different contributions of triplets across few-shot tasks**.

- By learning to compute representations **using the structures of support subgraphs and KGs**, FIETA can handle few-shot tasks on KGs with new entities and relations appearing after training.

- Provide six datasets for few-shot inductive KGC, and **FIETA outperforms 19 baselines** across various inductive scenarios.

Thank You!

▲ BDI Lab

▲ GitHub

Our datasets and codes are available at:
https://github.com/bdi-lab/FIETA

You can find us at:
{chanyoung.chung, hminsung, jjwhang}@kaist.ac.kr
https://bdi-lab.kaist.ac.kr

KAIST BDILab
BIG DATA INTELLIGENCE