8 Details for FOL

First-Order Logic (FOL) Queries. In a knowledge graph, FOL queries are essential for extracting relevant information. An FOL query $\mathcal{Q}$ is composed of constants $\mathcal{C}$ (a subset of entities $\mathcal{E}$), variables $\mathcal{V}$ (existentially quantified and also a subset of $\mathcal{E}$), relation projections $\mathcal{R}(a,b)$ that represent binary functions over constants or variables (a and b), and logical symbols ($\exists, \wedge, \vee, \neg$). The goal of answering an FOL query $\mathcal{Q}$ is to identify the target answers $\mathcal{A}_{\mathcal{G}}(\mathcal{Q})$ by determining variable assignments within a subgraph of the knowledge graph. This process requires performing logical operations to ascertain the entities that fulfill the query conditions.

Referencing [34], we identify nine types of complex query structures that involve various geometric operations such as projection, intersection, and union. These structures are divided into six single-operation query types and three mixed-operation query types, each represented by a specific formula.

$$
\begin{aligned}
&[1p]q = v_? : r_1(e_1, v_?)\\
&[2p]q = v_? : \exists u : r_1(e_1, u) \wedge r_2(u, v_?)\\
&[3p]q = v_? : \exists u_1, u_2 : r_1(e_1, u_1) \wedge r_2(u_1, u_2) \wedge r_3(u_2, v_?)\\
&[2i]q = v_? : r_1(e_1, v_?) \wedge r_2(e_2, v_?)\\
&[3i]q = v? : r_1(e_1, v_?) \wedge r_2(e_2, v_?) \wedge r_3(e_3, v_?)\\
&[pi]q = v? : \exists u : r_1(e_1, u) \wedge r_2(u, v_?) \wedge r_3(e_2, v_?)\\
&[ip]q = v_? : \exists u : r_1(e_1, u) \wedge r_2(e_2, u) \wedge r_3(u, v_?)\\
&[2u]q = v_? : r_1(e_1, v_?) \vee r_2(e_2, v_?)\\
&[up]q = v_? : \exists u : (r_1(e_1, u) \vee r_2(e_2, u)) \wedge r_3(u, v_?)
\end{aligned}
$$

where the variables e_i and v_i represent the anchor entities and quantified bound variables entities, respectively, while $v_?$ signifies the target answer entities to the query. Due to the intricate nature of these complex queries, which encompass a wealth of logical and structural information, it becomes imperative to incorporate structural knowledge into the process of textual encoding. This integration enhances the ability to achieve inductive generalization within the confines of the same logical structure.

9 Details for Theoretical Analysis

9.1 Proofs for Theorem 1

Proof *We prove this by induction. Recall that $\mathbf{1}_q(u = v) = \boldsymbol{q}$ if $u = v$ and 0 otherwise. Note that paths of length 0 only contain self-loops, which we define as important paths because they should be prefixes of some longer important paths.*

Suppose $\boldsymbol{w}_q(e_i)$ is the representation of an edge e_i conditional on the relation q. For the base case, at layer $t = 0$, we have $\boldsymbol{h}_q^{(0)}(u,u) = \bigoplus_{P\in\mathcal{P}_{u\to u|q:|P|\le 0}} \bigotimes_{i=1}^{|P|} \boldsymbol{w}_q(e_i)$ because the only path from u to u is the self-loop, which is represented by $\boldsymbol{q}$. For $u \neq v$, $h_q^{(0)}(u,v) = 0$ because there are no relational paths from u to v at path length 0.

For the inductive case $t > 0$, we have

$$
\begin{aligned}
&\boldsymbol{h}_q^{(t)}(u,v)\\
=&\boldsymbol{h}_q^{(0)}(u,v) \oplus \bigoplus_{x\in\mathcal{N}_u^{(t-1)}(v)} \bigoplus_{(x,r,v)\in\mathcal{E}(v)} \boldsymbol{h}_q^{(t-1)}(u,x) \otimes \boldsymbol{w}_q(x,r,v)
\end{aligned}
$$

$$
\begin{aligned}
=&\boldsymbol{h}_q^{(0)}(u,v) \oplus \bigoplus_{x\in\mathcal{N}_u^{(t-1)}(v)} \left(\bigoplus_{(x,r,v)\in\mathcal{E}(v)} \left(\bigoplus_{P\in\hat{\mathcal{P}}_{u\to v|q}^{(\le t-1)}} \bigotimes_{i=1}^{|P|} \boldsymbol{w}_q(e_i) \right)\right.\\
&\left.\otimes \boldsymbol{w}_q(x,r,v)\right)\\
=&\boldsymbol{h}_q^{(0)}(u,v) \oplus \bigoplus_{x\in\mathcal{N}_u^{(t-1)}(v)} \left(\bigoplus_{P\in\hat{\mathcal{P}}_{u\to x|q}^{(\le t-1)}} \bigotimes_{i=1}^{|P|} \boldsymbol{w}_q(e_i) \otimes \boldsymbol{w}_q(x,r,v) \right)\\
=&\left(\bigoplus_{P\in\hat{\mathcal{P}}_{u\to v|q}^{(0)}} \bigotimes_{i=1}^{|P|} \boldsymbol{w}_q(e_i) \right) \oplus \left(\bigoplus_{P\in\hat{\mathcal{P}}_{u\to v|q}^{(\le t)}} \bigotimes_{i=1}^{|P|} \boldsymbol{w}_q(e_i) \right)\\
=&\bigoplus_{P\in\hat{\mathcal{P}}_{u\to v|q}^{(\le t)}} \bigotimes_{i=1}^{|P|} \boldsymbol{w}_q(e_i),
\end{aligned}
$$

As for the structural-semantic learning, we have $s_q^{(l)}(u,v) = \bigoplus_{P\in\hat{\mathcal{P}}_{u\to v|q}} \bigotimes_{i=1}^{|P|} \boldsymbol{w}_q(e_i)$, where P represents the one-hop paths. For simplicity of representation, we omit the weights of structural-semantic and relational-semantic knowledge. In this way, we can derive the formula as follows:

$$
e_q^{(l)}(u,v) = s_q^{(l)}(u,v) + h_q^{(l)}(u,v) = \bigoplus_{s,r} \bigoplus_{P\in\hat{\mathcal{P}}_{u\to v|q}} \bigotimes_{i=1}^{|P|} w_q(e_i)
$$

In this manner, we observe that the intermediate representation converges to our path formulation, with $\lim_{t\to\infty} h_q^{(t)}(u,v) = h_q(u,v)$. In practical applications, particularly for link prediction, we find that convergence requires only a very small number of iterations, typically around $T = 6$. Consequently, the theorem is completed.

9.2 Proofs for Proposition 1

Proof *We derive this proposition based on the Definition 4:*

$$
\begin{aligned}
r_1(a,x) &\in \Phi_{MTF}\\
\exists x.r_1(a,x) \wedge r_2(x,y) &\in \Phi_{MTF}\\
\neg\exists x.r_1(a,x) \wedge r_2(x,y) &\in \Phi_{MTF}\\
\forall x.\neg r_1(a,x) \vee \neg r_2(x,y) &\in \Phi_{MTF}
\end{aligned}
$$

We observe that in Definition 3, negation is applied solely to atomic formulas, whereas in Definition 4, negation extends to the entire formula. This distinction leads to the use of the universal quantifier in the latter. Additionally, Definition 3 is not applicable for multi-hop scenarios, while Definition 4 is specifically designed for such contexts. In summary, we have proved $\Phi_{EFO} \subsetneq \Phi_{MTF}$.

9.3 Proofs for Theorem 2

Proof *First, we demonstrate the optimality of the hierarchical reasoning optimization by proving that the derived $T^*(v)$ indeed represents the maximum truth value for a subquery rooted at v. We establish this by induction: assuming the proposition holds for subqueries rooted at child nodes of v, then according to the derivation process outlined in Equations (11-17), it also holds for v. The trivial case, where the child node of v is a constant entity, also supports this assertion by Eq.(11-17). Therefore, the maximum truth value for the query is obtainable through Eq.(17), consistent with the definition of $T^*(v)$.*

Consider the truth value of a set of assignments expressed as $T(v) = T^*(v = e)$. In analyzing the subquery rooted at v when v is assigned an arbitrary $e \in V$, the truth value $T(v)$ under the returned assignments for an intersection operation is:

$$
T(v) = \min_{1\le i\le K}(T^*(v_i)) = \min_{1\le i\le K}(T^*(v_i = e_i))
$$

while

$$
T^*(v = e) = \max_{T_{1\le i\le K}(T^*(v_i=e_i))} = T_{1\le i\le K}(T^*(v_i = e_i))
$$

Hence $T(v) = T^*(v = e)$. Similarly, it also holds for union:

$$
T(v) = \max_{1\le i\le K}(T^*(v_i)) = \max_{1\le i\le K}(T^*(v_i = e_i)) = T^*(v = e)
$$

For relational projection, by the induction hypothesis, we have

$$T(v) = T^*(v = e) \cap T(r(e, e)), e = \arg\max_{e' \in V}(T^*(v = e') \cap T(r(e, e')))$$

$$\Rightarrow T(v) = \max_{e' \in V}(T^*(v = e') \cap T(r(e, e')))$$

while by definition of T^*, it holds that

$$\begin{aligned} T^*(v = e) &= \max(T(v) \cap T(r(v, e))) \\ &= \max_{e' \in V}(T^*(v = e') \cap T(r(e, e'))) \end{aligned}$$

Thus $T(v) = T^*(v = e)$. Similarly, for anti-relational projection, it holds:

$$\begin{aligned} T(v) &= T^*(v = e) \cap T(1 - r(e, e')), \\ e &= \arg\max_{e' \in V}(T^*(v = e') \cap T(1 - r(e, e'))) \end{aligned}$$

$$\Rightarrow T(v) = \max(T^*(v = e) \cap T(1 - r(e, e'))) = T^*(v = e)$$

Hence, the induction hypothesis is confirmed for v. The trivial case occurs when $v = c$ is a constant entity, and its truth value directly corresponds to $T(c) = T^*(c)$. Consequently, the root v yields a set of assignments with the following truth value:

$$\begin{aligned} T(v) &= T^*(v = e), \\ e &= \arg\max_{e' \in V}(T^*(v = e')) = T'(v) = \max_{e' \in V}(T^*(v = e')) \end{aligned}$$

which is the optimal truth value.

10 THE PROOF FOR THEOREM 3

As a realistic scenario, $\mathcal{D}^{(new)}$ in the dataset for new entities follows different distributions from the training dataset, and such new distribution is unknown in advance. To improve the generalization ability, we adapt model parameters learned from the dataset in the training period $\phi^{(old)}$, resulting in updated parameters $\phi^{(new)}$. Formally, we view $p(\phi^{(old)})$ as prior parameter distribution in the training period and aim to learn the posterior parameter distribution $q(\phi^{(new)})$ conditioning on $\mathcal{D}^{(new)}$. The unseen distribution of $\mathcal{D}^{(new)}$ prevents us from estimating $q(\phi^{(new)})$ by either using unbiased empirical loss or Bayes rules. Therefore, for convenience of discussion, we first define the difference of real predictive loss and its empirical estimation as follows

$$\Delta\mathcal{L} = \mathcal{L}\left(f_{\phi^{(old)}}, \mathcal{D}^{(new)}\right) - \hat{\mathcal{L}}\left(f_{\phi^{(old)}}, \mathcal{D}^{(new)}\right) \tag{22}$$

We are interested at the relation of ΔL and the distribution discrepancy between $p(\phi^{(old)})$ and $p(\phi^{(new)})$. Towards this goal, following [28], [38], we construct the following function:

$$\begin{aligned} f\left(\mathcal{D}^{(new)}\right) =& 2\left(\left|\mathcal{D}^{(new)}\right| - 1\right)\mathbb{E}_{\phi\sim q\left(\phi^{(new)}\right)}\left[(\Delta\mathcal{L})^2\right] \\ &- \mathbb{KL}\left(q\left(\phi^{(new)}\right)\|p\left(\phi^{(old)}\right)\right) \end{aligned} \tag{23}$$

Next, using Markov's inequality, we have:

$$p\left(f\left(\mathcal{D}^{(new)}\right) > \epsilon\right) = p\left(e^{f\left(\mathcal{D}^{(new)}\right)} > e^{\epsilon}\right) \leq \frac{\mathbb{E}_{\tilde{e}}\left[e^{f\left(\mathcal{D}^{(new)}\right)}\right]}{e^{\epsilon}}, \tag{24}$$

where $\mathcal{E}_{\tilde{e}}\left[e^{f\left(\mathcal{D}^{(new)}\right)}\right]$ denotes the expectation of $e^{f\left(\mathcal{D}^{(new)}\right)}$ *w.r.t.* new entity distribution $\tilde{e} \in \mathcal{D}^{()}$. To upper bound the expectation, we have the following inequality:

$$\begin{aligned} f\left(\mathcal{D}^{(new)}\right) =& 2\left(\left|\mathcal{D}^{(new)}\right| - 1\right)\mathbb{E}_{\phi\sim q\left(\phi^{(new)}\right)}\left[(\Delta\mathcal{L})^2\right] \\ &- \mathbb{KL}\left(q\left(\phi^{(new)}\right)\|p\left(\phi^{(m)}\right)\right) \\ =& \mathbb{E}_{\phi\sim q\left(\phi^{(new)}\right)}\left[\log\left(e^{2\left(\left|\mathcal{D}^{(new)}\right|-1\right)(\Delta\mathcal{L})^2}\frac{q\left(\phi^{(new)}\right)}{p\left(\phi^{(old)}\right)}\right)\right] \\ \leq& \log\left(\mathbb{E}_{\phi\sim q\left(\phi^{(new)}\right)}\left[e^{2\left(\left|\mathcal{D}^{(new)}\right|-1\right)(\Delta\mathcal{L})^2}\frac{q\left(\phi^{(new)}\right)}{p\left(\phi^{(old)}\right)}\right]\right) \\ =& \log\left(\mathbb{E}_{\phi\sim p\left(\phi^{(old)}\right)}\left[e^{2\left(\left|\mathcal{D}^{(new)}\right|-1\right)(\Delta\mathcal{L})^2}\right]\right) \end{aligned} \tag{25}$$

where Jensen's inequality is utilized to derive the inequality. Therefore, we have

$$\begin{aligned} \mathbb{E}_{\tilde{e}}\left[e^{f\left(\mathcal{D}^{(new)}\right)}\right] &\leq \mathbb{E}_{\tilde{e}}\mathbb{E}_{\phi\sim p\left(\phi^{(old)}\right)}\left[e^{2\left(\left|\mathcal{D}^{(new)}\right|-1\right)(\Delta\mathcal{L})^2}\right] \\ &= \mathbb{E}_{\phi\sim p\left(\phi^{(old)}\right)}\mathbb{E}_{\tilde{e}}\left[e^{2\left(\left|\mathcal{D}^{(new)}\right|-1\right)(\Delta\mathcal{L})^2}\right] \end{aligned} \tag{26}$$

We switch the order of expectations because $p(\phi)$ is independent to $\mathcal{D}^{(new)}$. Next, based on Hoeffding's inequality, we have:

$$p(\Delta\mathcal{L} > \epsilon) \leq e^{-2\left|\mathcal{D}^{(new)}\right|\epsilon^2} \tag{27}$$

and we can further derive the following inequality:

$$\begin{aligned} \mathbb{E}_{\tilde{e}}\left[e^{f\left(\mathcal{D}^{(new)}\right)}\right] \leq& \mathbb{E}_{\phi\sim p\left(\phi^{(old)}\right)}\mathbb{E}_{\tilde{e}}\left[e^{2\left(\left|\mathcal{D}^{(new)}\right|-1\right)}(\Delta\mathcal{L})^2\right] \\ \leq& \left|\mathcal{D}^{(new)}\right| \end{aligned} \tag{28}$$

Combining Eq.(28) and Eq.(24), we get:

$$p\left(f\left(\mathcal{D}^{(new)}\right) > \epsilon\right) \leq \frac{\left|\mathcal{D}^{(new)}\right|}{e^{\epsilon}} = \delta \tag{29}$$

where $\delta = |\mathcal{D}^{(new)}|/e^{\epsilon}$. Therefore, with the probability of at least $1 - \delta$, we have that for all $\phi^{(new)}$:

$$\begin{aligned} f\left(\mathcal{D}^{(new)}\right) =& 2\left(\left|\mathcal{D}^{(new)}\right| - 1\right)\mathbb{E}_{\phi\sim q\left(\phi^{(new)}\right)}\left[(\Delta\mathcal{L})^2\right] \\ &- \mathbb{KL}\left(q\left(\phi^{(new)}\right)\|p\left(\phi^{(old)}\right)\right) \\ \leq& \frac{\log\left|\mathcal{D}^{(new)}\right|}{\delta} \end{aligned} \tag{30}$$

Further, by utilizing Jensen's inequality again, we have:

$$\begin{aligned} &\left(\mathbb{E}_{\phi\sim q\left(\phi^{(new)}\right)}[(\Delta\mathcal{L})]\right)^2 \\ \leq& \mathbb{E}_{\phi\sim q\left(\phi^{(new)}\right)}\left[(\Delta\mathcal{L})^2\right] \\ \leq& \frac{\mathbb{KL}\left(q\left(\phi^{(new)}\right)\|p\left(\phi^{(old)}\right)\right) + \frac{\log\left|\mathcal{D}^{(new)}\right|}{\delta}}{2\left(|\mathcal{D}^{(new)}| - 1\right.} \end{aligned} \tag{31}$$

In this way, the generalization bound is affected by $\mathbb{KL}(q(\phi^{(new)})\|p(\phi^{(old)}))$, which is due to the distributional difference of $\phi^{(new)}$ and $\phi^{(old)}$. One can observe that $\mathbb{KL}(q(\phi^{(new)})\|p(\phi^{(old)})) = C(p(\phi^{(new)}), p(\phi^{(old)})) - I(p(\phi^{(new)}))$, where $C(\cdot,\cdot)$ is the cross entropy and the $I(\cdot)$ is the entropy. Substituting definition of Δ in Eq.(31), we can have that:

$$\begin{aligned} &\mathcal{L}\left(f_{\phi^{(old)}}, \mathcal{D}^{(new)}\right) \\ \leq& \hat{\mathcal{L}}\left(f_{\phi^{(old)}}, \mathcal{D}^{(new)}\right) \\ &+ \sqrt{\frac{C\left(q\left(\phi^{(new)}\right), p\left(\phi^{(old)}\right)\right) - I^{(new)} + \log\frac{\left|\mathcal{D}^{(new)}\right|}{\delta}}{2\left|\mathcal{D}^{(old)}\right| - 1}} \end{aligned} \tag{32}$$

where $I^{(new)}$ is the $I(p\left(\phi^{(new)}\right))$. In this way, when the distributions of $p(\phi^{(new)})$ is equal (*i.e.*, $\mathcal{E}_{train} = \mathcal{E}_{inf}$) to that of $p(\phi^{(old)})$, then generation bound can degrade to

$$\mathcal{L}\left(f_{\phi^{(old)}}, \mathcal{D}^{(new)}\right) \leq \hat{\mathcal{L}}\left(f_{\phi^{(old)}}, \mathcal{D}^{(new)}\right) + \sqrt{\frac{\log \frac{\left|\mathcal{D}^{(new)}\right|}{\delta}}{2\left|\mathcal{D}^{(old)}\right| - 1}} \tag{33}$$

Then the proof is complete.

11 Details for SAQE

11.1 Conversion Between FOL Expression and Multi-hop Query Computation Tree

Transforming a disjunctive normal form FOL expression into its corresponding query computation tree is a three-phase process, encompassing the creation of a dependency graph, the consolidation of union branches, and the isolation of variables.

Dependency Graph Generation. For each FOL expression, a distinct node is created for every variable and constant entity found within single-hop atoms. It is noteworthy that the same constant entity might be depicted by various nodes when featured in multiple single-hop atoms. Connections between nodes are established through undirected edges that reflect their respective relationships as described in the single-hop atoms. For instance, if $e_j^i = r(v', v)$ (or $r(c, v)$), a linking edge labeled r_i is drawn between the node of v' (or c) and v. In cases where $e_j^i = \neg r(v', v)$ (or $\neg r(c, v)$), the link is marked with an edge labeled $\neg r_i$, with i serving to differentiate edges derived from distinct conjunctions. This undirected dependency multigraph needs to be a tree—a connected, acyclic graph. The node labeled $v?$ is assigned as the root, with all edges reoriented to point from the child nodes toward this parent node, thus reversing the relational directions. Nodes representing constant entities serve as leaf nodes, each connecting to a single variable node within the structure.

Union Branches Merging. In the query computation tree, union structures are addressed by analyzing the path τ from the root to each leaf node. We pinpoint the node v_i, where each connecting edge to its descendant v_j indicates the same relation albeit across various conjunctions: $r_{t_1}, r_{t_2}, \ldots, r_{t_p}$. These edges are consolidated into a single edge $r_{t_1,t_2,\ldots,t_p}$ by employing the distributive law:

$$(P \wedge Q) \vee (P \wedge R) \leftrightarrow P \wedge (Q \vee R)$$

wher P, Q and R are terms. This adjustment guarantees a subpath from v_i to some node v_k along τ that consists only of the edge $r_{t_1,t_2,\ldots,t_p}$, with v_k branching to different child nodes via relations from conjunctions $t_1, t_2, \ldots, t_p$. Edges are classified as union, whereas one-to-many relations are categorized as intersections, thus simplifying the multigraph into a graph without redundant edges.

Variable Separation. In managing one-to-many relations defined by intersection/union structures, we subdivide the parent node v_k into several nodes $v_k^1, v_k^2, \ldots$, linking each to its respective child node through the original relational edge. Each segmented node reconnects to v_k via intersection or union edges. The process of merging union branches often results in complex one-to-many relationships, which may feature both intersection and union edges. To address this complexity, v_k is first divided to form a union structure (*e.g.*, v_k^3 and v_k^4), and then the child node is separated into an intersection structure (*e.g.*, v_k^1 and v_k^2). It is demonstrated that the logic expression represented by the query computation tree faithfully corresponds to the original FOL expression:

$$(r_1(c_1, v_1) \wedge r_2(c_2, v_1)) \vee (r_4(v_1, v_2)) \vee (r_3(c_3, v_1) \wedge r_4(v_1, v_2))$$

Furthermore, we argue that the query computation multi-hop tree provides a more suitable representation for complex logical queries on knowledge graphs. Although FOL expressions offer a more generalized form, such expressions are seldom employed in practical scenarios due to their complexity and abstraction. In contrast, the query computation tree aligns more closely with the human approach to multi-hop questioning. Typically, an intermediate variable in such queries is characterized by its relationships with other entities or their logical combinations. This method of structuring information is effectively mirrored in the query computation tree, where each subtree rooted at a variable encapsulates these relationships, providing a clear and navigable structure that enhances both understanding and application.

12 Details for Experiments

We conduct extensive in Table 6, where SAQE$_s$ is a single result and SAQE$_a$ is the average performance. For extensive comparision, you can choose SAQE$_a$ as baseline. We also provide more results on inductive FOL query answering when $\mathcal{E}_{inf}/\mathcal{E}_{train} = 550\%, 300\%, 217\%, 175\%, 150\%, 106\%$ in Table 7, where SAQE$_v$ is the valid setting and SAQE$_t$ is the test setting performance. For extensive comparision, you can choose SAQE$_t$ as baseline.

TABLE 6
Test MRR results (%) on answering FOL queries On NELL995 datasets. SAQE$_s$ is a single result and SAQE$_a$ is the average performance. avg$_n$ is the average MRR on queries with negation.

Datasets	Model	avg$_p$	avg$_n$	1p	2p	3p	2i	3i	pi	ip	2u	up	2in	3in	inp	pin	pni
NELL995	SAQE$_s$	33.9	13.4	61.9	25.7	22.5	43.6	51.8	31.2	27.6	21.9	18.8	15.2	17.7	17.3	10.8	6.2
	SAQE$_a$	33.9	13.3	60.9	24.3	22.3	44.3	55.2	32.5	27.6	20.0	18.5	13.9	15.2	17.4	11.0	8.9
FB15K237	SAQE$_s$	34.6	16.7	50.1	22.8	23.7	42.9	56.4	39.6	29.6	24.1	22.3	17.2	27.6	16.8	14.1	7.6
	SAQE$_a$	33.6	17.7	49.0	21.5	21.4	43.1	57.8	38.3	27.7	22.6	21.1	17.3	27.3	15.9	14.8	13.6
FB15K	SAQE$_s$	75.3	50.7	90.6	70.2	60.1	81.4	85.2	77.1	73.8	78.1	61.2	62.7	61.0	48.5	48.7	32.6
	SAQE$_a$	77.2	59.5	89.5	72.1	66.0	82.2	86.6	78.2	75.5	76.8	68.0	63.0	66.3	56.5	54.2	57.5

TABLE 7
Test Hits@10 results (%) on answering inductive FOL queries when $\mathcal{E}_{inf}/\mathcal{E}_{train} = 550\%, 300\%, 217\%, 175\%, 150\%, 106\%$. avg$_p$ is the average on EPFO queries ($\wedge, \vee$). SAQE$_v$ is the valid setting and SAQE$_t$ is the test setting performance.

Ratio	Model	avg$_p$	avg$_n$	1p	2p	3p	2i	3i	pi	ip	2u	up	2in	3in	inp	pin	pni
106%	SAQE$_v$	51.3	38.4	73.6	37.9	32.0	68.4	73.2	48.8	42.6	49.4	32.5	40.2	44.1	25.2	29.7	49.8
	SAQE$_t$	49.9	37.8	71.6	37.0	32.1	67.0	73.4	47.5	41.1	46.4	33.2	39.8	42.8	23.9	33.6	48.8
150%	SAQE$_v$	50.8	31.5	71.8	36.7	29.5	72.6	81.0	53.8	40.1	35.6	29.7	29.5	41.6	28.9	29.2	27.3
	SAQE$_t$	49.1	30.5	67.2	36.8	31.0	72.1	80.6	52.1	40.0	34.0	27.9	29.4	39.1	27.8	27.4	28.7
175%	SAQE$_v$	51.6	34.3	67.9	37.3	32.3	72.5	82.9	57.6	43.3	39.8	30.4	33.6	46.5	28.6	29.4	33.3
	SAQE$_t$	51.8	34.8	65.6	37.9	32.4	73.7	83.9	58.7	43.9	40.3	29.9	34.7	47.2	30.1	28.7	33.4
217%	SAQE$_v$	50.1	31.5	63.5	36.0	29.6	73.1	88.9	59.7	43.6	30.7	24.8	27.6	48.9	31.5	22.5	23.7
	SAQE$_t$	49.4	29.5	63.2	36.7	29.4	73.2	86.7	58.4	42.8	28.3	25.9	26.4	46.9	29.2	22.9	22.3
300%	SAQE$_v$	43.7	22.5	54.3	32.0	25.4	68.2	79.4	51.0	37.0	23.8	21.8	19.4	34.4	25.1	18.9	13.8
	SAQE$_t$	44.7	21.6	60.3	32.0	25.9	68.6	79.8	51.1	37.4	24.0	22.8	18.5	33.8	24.7	18.0	12.9
550%	SAQE$_v$	38.6	17.5	44.5	25.9	18.9	64.7	82.6	49.7	28.1	16.5	16.7	14.3	31.9	16.9	15.0	9.4
	SAQE$_t$	37.9	17.7	42.7	25.2	17.8	63.7	82.0	49.2	26.5	18.3	16.4	15.7	32.1	17.6	13.8	11.0