FastLog: Efficient End-to-end Rule Learning Over Large-scale Knowledge Graphs by Reduction to Vector Operations (Supplementary Material)

A PROOFS

In this section, we provide detailed proofs for all propositions in this work.

A.1 Proof of Proposition 1

Proof. (I) We first prove that the time complexity of a forward computation step for TensorLog is $O(NL(|\mathcal{K}|+|\mathcal{R}||\mathcal{E}|))$. Let $\mathrm{nnz}(M_{r_i})$ be the number of non-zero elements in the sparse matrix M_{r_i}. From Equations (1-2), we know that the complexity for each step in TensorLog comes from $\sum_{i=1}^{2n+1}\phi_{r,x}^{(k,l-1)}(w_i^{(r,k,l)}M_{r_i})$. Since the time complexity of $\sum_{i=1}^{2n+1}\phi_{r,x}^{(k,l-1)}(w_i^{(r,k,l)}M_{r_i})$ is $\sum_{i=1}^{2n+1}(\mathrm{nnz}(M_{r_i})+|\mathcal{E}|)$, where $n=|\mathcal{R}|$. By $\sum_{i=1}^{2n+1}\mathrm{nnz}(M_{r_i})=|\mathcal{K}|$, we can infer that the time complexity of a forward computation step for TensorLog is $O(NL(|\mathcal{K}|+|\mathcal{R}||\mathcal{E}|))$.

(II) We then prove that the time complexity of a backward propagation step for TensorLog is $O(NL(|\mathcal{K}|+|\mathcal{R}||\mathcal{E}|))$. For a backward propagation step, we know that only $w^{(r,k,l)}$ is trainable. The time complexity for calculating $\frac{\partial\mathcal{L}}{\partial\phi_{r,x}^{(k,l)}}M_{r_i}^{\top}$ is $\mathrm{nnz}(M_{r_i})+|\mathcal{E}|$. Therefore, the time complexity of a backward propagation step for TensorLog is $O(NL\sum_{i=1}^{2n+1}(\mathrm{nnz}(M_{r_i})+|\mathcal{E}|)) = O(NL(|\mathcal{K}|+|\mathcal{R}||\mathcal{E}|))$. □

A.2 Proof of Proposition 2

Proof. (I) From Proposition 1 we know that the time complexity of a forward computation step for TensorLog is $O(L(|\mathcal{K}|+|\mathcal{R}||\mathcal{E}|))$. From Equations (3-6), we know that the time complexity of a forward computation step for NeuralLP is

$$\underbrace{L(|\mathcal{K}|+|\mathcal{R}||\mathcal{E}|)}_{\text{TensorLog}}+\underbrace{\frac{L(L-1)}{2}|\mathcal{E}|}_{\text{Aggregation}}+\underbrace{(L+1)(8d^2)}_{\text{LSTM network}}+\underbrace{(d^2+d)}_{\text{MLP}}$$

where d denotes the hidden size. In general, it holds that $L(|\mathcal{K}|+|\mathcal{R}||\mathcal{E}|)+\frac{L(L-1)}{2}|\mathcal{E}| \gg (L+1)(8d^2)+(d^2+d)$. Therefore, we can infer that the time complexity of a forward computation step for NeuralLP is $O(L(|\mathcal{K}|+|\mathcal{R}||\mathcal{E}|)+L^2|\mathcal{E}|)$.

(II) From Proposition 1 we know that the time complexity of a backward propagation step for TensorLog is $O(L(|\mathcal{K}|+|\mathcal{R}||\mathcal{E}|))$. From Equations (3-6), we know that the time complexity of a backward propagation step for NeuralLP is

$$\underbrace{2L(|\mathcal{K}|+|\mathcal{R}||\mathcal{E}|)}_{\text{TensorLog}}+\underbrace{L(L-1)|\mathcal{E}|}_{\text{Aggregation}}+\underbrace{2(L+1)(8d^2)}_{\text{LSTM network}}+\underbrace{2(d^2+d)}_{\text{MLP}}$$

In general, it holds that $L(|\mathcal{K}|+|\mathcal{R}||\mathcal{E}|)+L(L-1)|\mathcal{E}| \gg +2(L+1)(8d^2)+2(d^2+d)$. Therefore, we can infer that the time complexity of a forward computation step for NeuralLP is $O(L(|\mathcal{K}|+|\mathcal{R}||\mathcal{E}|)+L^2|\mathcal{E}|)$ □

A.3 Proof of Proposition 3

Proof. From Equations (7-13), we know that DRUM, smDRUMand mmDRUM has the same training time complexity.

(I) From Proposition 1 we know that the time complexity of a forward computation step for TensorLog is $O(L(|\mathcal{K}|+|\mathcal{R}||\mathcal{E}|))$. From Equations (7-9), we know that the time complexity of a forward computation step for DRUM is

$$\underbrace{NL(|\mathcal{K}|+|\mathcal{R}||\mathcal{E}|)}_{\text{TensorLog}}+\underbrace{2N(L+1)(8d^2)}_{\text{BiLSTM networks}}+\underbrace{N((2d)^2+2d)}_{\text{MLPs}}$$

where d denotes the hidden size. In general, it holds that $NL(|\mathcal{K}|+|\mathcal{R}||\mathcal{E}|) \gg N(L+1)(8d^2)+N((2d)^2+2d)$. Therefore, we can infer that the time complexity of a forward computation step for DRUM, smDRUM, and mmDRUM is $O(NL(|\mathcal{K}|+|\mathcal{R}||\mathcal{E}|))$.

(II) From Proposition 1 we know that the time complexity of a backward propagation step for TensorLog is $O(L(|\mathcal{K}|+|\mathcal{R}||\mathcal{E}|))$. From Equations (7-9), we know that the time complexity of a backward propagation step for DRUM, smDRUM, and mmDRUM is

$$\underbrace{2NL(|\mathcal{K}|+|\mathcal{R}||\mathcal{E}|)}_{\text{TensorLog}}+\underbrace{4N(L+1)(8d^2)}_{\text{BiLSTM networks}}+\underbrace{2N((2d)^2+2d)}_{\text{MLPs}}$$

In general, it holds that $2NL(|\mathcal{K}|+|\mathcal{R}||\mathcal{E}|) \gg 4N(L+1)(8d^2)+2N((2d)^2+2d)$. Therefore, we can infer that the time complexity of a forward computation step for DRUM, smDRUM, and mmDRUM is $O(NL(|\mathcal{K}|+|\mathcal{R}||\mathcal{E}|))$ □

A.4 Proof of Proposition 4

Proof. (I) From Equation (15), we know that the time complexity of the function $\mathcal{F}_{\mathrm{e2f}}$ is $|\mathcal{K}|$. From Equation (16), we know that the time complexity of the function $\mathcal{F}_{\mathrm{r2f}}$ is $|\mathcal{K}|$. From Equation (17), we know that the time complexity of the function $\mathcal{F}_{\mathrm{f2e}}$ is $2|\mathcal{K}|$. From Equation (18), we know that the time complexity of a forward computation step for FastLog is

$$NL(\underbrace{|\mathcal{K}|}_{\mathcal{F}_{\mathrm{e2f}}}+\underbrace{|\mathcal{K}|}_{\mathcal{F}_{\mathrm{r2f}}}+\underbrace{|\mathcal{K}|}_{\odot}\underbrace{2|\mathcal{K}|}_{\mathcal{F}_{\mathrm{f2e}}})$$

Therefore, we can infer that the time complexity of a forward computation step for FastLogis $O(NL|\mathcal{K}|)$.

(II) For a backward propagation step of FastLog, we know that only $w^{(r,k,l)}$ is trainable. Let $z=\mathcal{F}_{\mathrm{e2f}}(\phi_{r,x}^{(k,l-1)})\odot\mathcal{F}_{\mathrm{r2f}}(w^{(r,k,l)})$. The time complexity for calculating $\frac{\partial\mathcal{F}_{\mathrm{f2e}}(z)}{\partial z}$ is $O(|\mathcal{K}|)$. The time complexity for calculating $\frac{\partial z}{\partial\mathcal{F}_{\mathrm{e2f}}(\phi_{r,x}^{(k,l-1)})}$ is $O(|\mathcal{K}|)$. The time complexity for calculating $\frac{\partial z}{\partial\mathcal{F}_{\mathrm{r2f}}(w^{(r,k,l)})}$ is $O(|\mathcal{K}|)$. The time complexity for calculating $\frac{\partial\mathcal{F}_{\mathrm{r2f}}(w^{(r,k,l)})}{\partial w^{(r,k,l)}}$ is $O(|\mathcal{K}|)$. Therefore, the time complexity of a backward propagation step for FastLogis $O(NL|\mathcal{K}|)$. □

A.5 Proof of Proposition 5

Proof. From Proposition 4, we know that the time complexity of a forward computation step for FastLogis $O(NL|\mathcal{K}|)$. From Equations (20-21), we know that the time complexity of a forward computation step for NeuralLP-FL is

$$\underbrace{L|\mathcal{K}|}_{\texttt{FastLog}} + \underbrace{\frac{L(L-1)}{2}|\mathcal{E}|}_{\text{Aggregation}} + \underbrace{(L+1)(8d^2)}_{\text{LSTM network}} + \underbrace{(d^2+d)}_{\text{MLP}}$$

where d denotes the hidden size. In general, it holds that $L|\mathcal{K}| + \frac{L(L-1)}{2}|\mathcal{E}| \gg (L+1)(8d^2)+(d^2+d)$. Therefore, we can infer that the time complexity of a forward computation step for NeuralLP-FL is $O(L|\mathcal{K}| + L^2|\mathcal{E}|)$.

(II) For a backward propagation step for NeuralLP-FL, we know that both $w^{(r,1,l)}$ and $\alpha^{(r,1,l)}$ are trainable. Let $z = \mathcal{F}_{\text{e2f}}(\sum_{j=0}^{l-1}\alpha_j^{(r,1,l)}\phi_{r,x}^{(1,j)}) \odot \mathcal{F}_{\text{r2f}}(w^{(r,1,l)})$. The time complexity for calculating $\frac{\partial \mathcal{F}_{\text{f2e}}(z)}{\partial z}$ is $O(|\mathcal{K}|)$. The time complexity for calculating $\frac{\partial z}{\partial \mathcal{F}_{\text{e2f}}(\sum_{j=0}^{l-1}\alpha_j^{(r,1,l)}\phi_{r,x}^{(1,j)})}$ is $O(|\mathcal{K}|)$. The time complexity for calculating $\frac{\partial \mathcal{F}_{\text{e2f}}(\sum_{j=0}^{l-1}\alpha_j^{(r,1,l)}\phi_{r,x}^{(1,j)})}{\partial \alpha^{(r,1,l)}}$ is $O(L^2|\mathcal{E}|)$. The time complexity for calculating $\frac{\partial z}{\partial \mathcal{F}_{\text{r2f}}(w^{(r,1,l)})}$ is $O(|\mathcal{K}|)$. The time complexity for calculating $\frac{\partial \mathcal{F}_{\text{r2f}}(w^{(r,1,l)})}{\partial w^{(r,1,l)}}$ is $O(|\mathcal{K}|)$. Therefore, the time complexity of a backward propagation step for NeuralLP-FL is $O(L|\mathcal{K}|+L^2|\mathcal{E}|)$. □

A.6 Proof of Proposition 6

To prove Proposition 6, we first introduce three sparse matrices $M_{\text{e2f}}, M_{\text{r2f}}$, and M_{f2e}, where $M_{\text{e2f}} \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{K}|}$ (resp. $M_{\text{r2f}} \in \mathbb{R}^{2n\times|\mathcal{K}|}$ or $M_{\text{f2e}} \in \mathbb{R}^{|\mathcal{K}|\times|\mathcal{E}|}$) stores the mapping between a head entity (resp. relation or fact) and its corresponding fact (resp. fact or tail entity).

Proof. For all $1 \le l \le L$, it holds that

$$\begin{aligned}
\phi_{r,a}^{(l)} &= \mathcal{F}_{\text{f2e}}(\mathcal{F}_{\text{e2f}}(\sum_{j=0}^{l-1}\alpha_j^{(r,l)}\phi_{r,a}^{(j)}) \odot \mathcal{F}_{\text{r2f}}(w^{(r,l)})) \\
&= ((\sum_{j=0}^{l-1}\alpha_j^{(r,l)}\phi_{r,a}^{(j)})M_{\text{e2f}} \odot (w^{(r,l)}M_{\text{r2f}}))M_{\text{f2e}} \\
&= (\sum_{j=0}^{l-1}\alpha_j^{(r,l)}\phi_{r,a}^{(j)})((M_{\text{e2f}} \odot (w^{(r,l)}M_{\text{r2f}}))M_{\text{f2e}}) \\
&= (\sum_{j=0}^{l-1}\alpha_j^{(r,l)}\phi_{r,a}^{(j)})(\sum_{i=1}^{2n}w_i^{(r,l)}M_{r_i}) \\
&= \sum_{i=1}^{2n}(\sum_{j=0}^{l-1}\alpha_j^{(r,l)}\phi_{r,a}^{(j)})(w_i^{(r,l)}M_{r_i})
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
\texttt{NeuralLP-FL}(\theta_r^L, a, b) &= \phi_{r,a}^{(L+1)}v_b \\
&= \sum_{j=0}^{L}\alpha_j^{(r,L+1)}\phi_{r,x}^{(j)} \\
&= \sum_{j=0}^{L}\alpha_j^{(r,L+1)}(\\
&\quad \sum_{i=1}^{2n}(\sum_{k=0}^{j-1}\alpha_k^{(r,j)}\phi_{r,a}^{(k)})(w_i^{(r,j)}M_{r_i})) \\
&= \texttt{NeuralLP}(\theta_r^L, a, b)
\end{aligned}$$

□

A.7 Proof of Proposition 7

Proof. (I) From Proposition 4, we know that the time complexity of a forward computation step for FastLogis $O(NL|\mathcal{K}|)$. From Equations (22-23), we know that the time complexity of a forward computation step for DRUM-FL is

$$\underbrace{NL|\mathcal{K}|}_{\texttt{FastLog}} + \underbrace{2N(L+1)(8d^2)}_{\text{BiLSTM networks}} + \underbrace{N((2d)^2+2d)}_{\text{MLPs}}$$

where d denotes the hidden size. In general, it holds that $NL|\mathcal{K}| \gg +2N(L+1)(8d^2) + ((2d)^2+2d)$. Therefore, we can infer that the time complexity of a forward computation step for DRUM-FL is $O(NL|\mathcal{K}|)$.

(II) For a backward propagation step of DRUM-FL, we know that only $w^{(r,k,l)}$ is trainable. Let $z = \mathcal{F}_{\text{e2f}}(\phi_{r,x}^{(k,l-1)}) \odot \mathcal{F}_{\text{r2f}}(w^{(r,k,l)})$. The time complexity for calculating $\frac{\partial \mathcal{F}_{\text{f2e}}(z)}{\partial z}$ is $O(|\mathcal{K}|)$. The time complexity for calculating $\frac{\partial z}{\partial \mathcal{F}_{\text{e2f}}(\phi_{r,x}^{(k,l-1)})}$ is $O(|\mathcal{K}|)$. The time complexity for calculating $\frac{\partial z}{\partial \mathcal{F}_{\text{r2f}}(w^{(r,k,l)})}$ is $O(|\mathcal{K}|)$. The time complexity for calculating $\frac{\partial \mathcal{F}_{\text{r2f}}(w^{(r,k,l)})}{\partial w^{(r,l,l)}}$ is $O(|\mathcal{K}|)$. Therefore, the time complexity of a backward propagation step for DRUM-FL is $O(NL|\mathcal{K}|)$. □

A.8 Proof of Proposition 8

To prove Proposition 8, we first introduce three sparse matrices $M_{\text{e2f}}, M_{\text{r2f}}$, and M_{f2e}, where $M_{\text{e2f}} \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{K}|}$ (resp. $M_{\text{r2f}} \in \mathbb{R}^{(2n+1)\times|\mathcal{K}|}$ or $M_{\text{f2e}} \in \mathbb{R}^{|\mathcal{K}|\times|\mathcal{E}|}$) stores the mapping between a head entity (resp. relation or fact) and its corresponding fact (resp. fact or tail entity).

Proof. For all $1 \le k \le N, 1 \le l \le L$, it holds that

$$\begin{aligned}
\phi_{r,a}^{(k,l)} &= \mathcal{F}_{\text{f2e}}(\mathcal{F}_{\text{e2f}}(\phi_{r,a}^{(k,l-1)}) \odot \mathcal{F}_{\text{r2f}}(w^{(r,k,l)})) \\
&= ((\phi_{r,a}^{(k,l-1)}M_{\text{e2f}}) \odot (w^{(r,k,l)}M_{\text{r2f}}))M_{\text{f2e}} \\
&= \phi_{r,a}^{(k,l-1)}((M_{\text{e2f}} \odot (w^{(r,k,l)}M_{\text{r2f}}))M_{\text{f2e}}) \\
&= \phi_{r,a}^{(k,l-1)}(\sum_{i=1}^{2n+1}w_i^{(r,k,l)}M_{r_i})
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
\texttt{DRUM-FL}(\theta_r^{N,L}, a, b) &= (\sum_{k=1}^{N} \phi_{r,a}^{(k,L)}) v_b \\
&= (\sum_{k=1}^{N} ((\cdots (v_a^\top (\sum_{i=1}^{2n+1} w_i^{(r,k,1)} M_{r_i})) \\
&\quad (\sum_{i=1}^{2n+1} w_i^{(r,k,2)} M_{r_i})) \\
&\quad \cdots \\
&\quad (\sum_{i=1}^{2n+1} w_i^{(r,k,L)} M_{r_i}))) v_b \\
&= v_a^\top (\sum_{k=1}^{N} \prod_{l=1}^{L} \sum_{i=1}^{2n+1} w_i^{(r,k,l)} M_{r_i}) v_b \\
&= \texttt{DRUM}(\theta_r^{N,L}, a, b)
\end{aligned}$$

□

A.9 Proof of Proposition 9

To prove Proposition 9, we first introduce three sparse matrices $M_{\text{e2f}}, M_{\text{r2f}}$, and M_{f2e}, where $M_{\text{e2f}} \in \mathbb{R}^{|\mathcal{E}| \times |\mathcal{K}|}$ (resp. $M_{\text{r2f}} \in \mathbb{R}^{(2n+1) \times |\mathcal{K}|}$ or $M_{\text{f2e}} \in \mathbb{R}^{|\mathcal{K}| \times |\mathcal{E}|}$) stores the mapping between a head entity (resp. relation or fact) and its corresponding fact (resp. fact or tail entity).

Proof. For all $1 \le k \le N, 1 \le l \le L$, it holds that

$$\begin{aligned}
\phi_{r,a}^{(k,l)} &= \mathcal{F}_{\text{f2e}}^{\max}(\mathcal{F}_{\text{e2f}}(\phi_{r,a}^{(k,l-1)}) \odot \mathcal{F}_{\text{r2f}}(w^{(r,k,l)})) \\
&= ((\phi_{r,a}^{(k,l-1)} M_{\text{e2f}}) \odot (w^{(r,k,l)} M_{\text{r2f}})) \otimes M_{\text{f2e}} \\
&= \phi_{r,a}^{(k,l-1)} ((M_{\text{e2f}} \odot (w^{(r,k,l)} M_{\text{r2f}})) \otimes M_{\text{f2e}}) \\
&= \phi_{r,a}^{(k,l-1)} \otimes ((M_{\text{e2f}} \odot (w^{(r,k,l)} M_{\text{r2f}})) M_{\text{f2e}}) \\
&= \phi_{r,a}^{(k,l-1)} \otimes (\sum_{i=1}^{2n+1} w_i^{(r,k,l)} M_{r_i})
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
\texttt{smDRUM-FL}(\theta_r^{N,L}, a, b) &= (\sum_{k=1}^{N} \phi_{r,a}^{(k,L)}) v_b \\
&= (\sum_{k=1}^{N} ((\cdots (v_a^\top \otimes (\sum_{i=1}^{2n+1} w_i^{(r,k,1)} M_{r_i})) \\
&\quad \otimes (\sum_{i=1}^{2n+1} w_i^{(r,k,2)} M_{r_i})) \\
&\quad \cdots \\
&\quad \otimes (\sum_{i=1}^{2n+1} w_i^{(r,k,L)} M_{r_i}))) v_b \\
&= v_a^\top (\sum_{k=1}^{N} \bigotimes_{l=1}^{L} \sum_{i=1}^{2n+1} w_i^{(r,k,l)} M_{r_i}) v_b \\
&= \texttt{smDRUM}(\theta_r^{N,L}, a, b)
\end{aligned}$$

□

A.10 Proof of Proposition 10

To prove Proposition 10, we first introduce three sparse matrices M_{e2f}, M_{r2f}, and M_{f2e}, where $M_{\text{e2f}} \in \mathbb{R}^{|\mathcal{E}| \times |\mathcal{K}|}$ (resp. $M_{\text{r2f}} \in \mathbb{R}^{(2n+1) \times |\mathcal{K}|}$ or $M_{\text{f2e}} \in \mathbb{R}^{|\mathcal{K}| \times |\mathcal{E}|}$) stores the mapping between a head entity (resp. relation or fact) and its corresponding fact (resp. fact or tail entity).

Proof. For all $1 \le k \le N, 1 \le l \le L$, it holds that

$$\begin{aligned}
\phi_{r,a}^{(k,l)} &= \mathcal{F}_{\text{f2e}}^{\max}(\mathcal{F}_{\text{e2f}}(\phi_{r,a}^{(k,l-1)}) \odot \mathcal{F}_{\text{r2f}}(w^{(r,k,l)})) \\
&= ((\phi_{r,a}^{(k,l-1)} M_{\text{e2f}}) \odot (w^{(r,k,l)} M_{\text{r2f}})) \otimes M_{\text{f2e}} \\
&= \phi_{r,a}^{(k,l-1)} ((M_{\text{e2f}} \odot (w^{(r,k,l)} M_{\text{r2f}})) \otimes M_{\text{f2e}}) \\
&= \phi_{r,a}^{(k,l-1)} \otimes ((M_{\text{e2f}} \odot (w^{(r,k,l)} M_{\text{r2f}})) M_{\text{f2e}}) \\
&= \phi_{r,a}^{(k,l-1)} \otimes (\sum_{i=1}^{2n+1} w_i^{(r,k,l)} M_{r_i})
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
\texttt{mmDRUM-FL}(\theta_r^{N,L}, a, b) &= (\max_{k=1}^{N} \phi_{r,a}^{(k,L)}) v_b \\
&= (\max_{k=1}^{N} ((\cdots (v_a^\top \otimes (\sum_{i=1}^{2n+1} w_i^{(r,k,1)} M_{r_i})) \\
&\quad \otimes (\sum_{i=1}^{2n+1} w_i^{(r,k,2)} M_{r_i})) \\
&\quad \cdots \\
&\quad \otimes (\sum_{i=1}^{2n+1} w_i^{(r,k,L)} M_{r_i}))) v_b \\
&= v_a^\top (\max_{k=1}^{N} \bigotimes_{l=1}^{L} \sum_{i=1}^{2n+1} w_i^{(r,k,l)} M_{r_i}) v_b \\
&= \texttt{mmDRUM}(\theta_r^{N,L}, a, b)
\end{aligned}$$

□

A.11 Proof of Proposition 11

Proof. (I) We first prove that the space complexity of a forward computation step for `TensorLog` is $O(m|\mathcal{E}|)$. For all $1 \le k \le N, 1 \le l \le L$, `TensorLog` requires a space of $m|\mathcal{E}|$ to store the intermediate estimated truth degrees. Since the summation of predicate selection is serial, this process does not require additional space. Therefore, a forward computation step for `TensorLog` is $O(m|\mathcal{E}|)$.

(II) We then prove that the space complexity of a backward propagation step for `TensorLog` is $O(mNL(2|\mathcal{R}|+1)|\mathcal{E}|)$. For a backward propagation step, `TensorLog` requires to store all intermediate estimated truth degrees for all L steps for all N rules to calculate the gradient. Therefore, the space complexity of a backward propagation step for `TensorLog` is $O(mNL(2|\mathcal{R}|+1)|\mathcal{E}|)$. □

A.12 Proof of Proposition 12

Proof. (I) We first prove that the space complexity of a forward computation step for `FastLog` is $O(m|\mathcal{K}|)$. For all $1 \le k \le N, 1 \le l \le L$, `FastLog` requires a space of the size $m|\mathcal{K}|$ to store the intermediate hidden state for all facts. Although `FastLog` also requires a space of $m|\mathcal{E}|$ to store the intermediate estimated truth degrees, it can reuse the previously opened space. In general, it holds that $|\mathcal{K}| > |\mathcal{E}|$. Therefore, a forward computation step for `FastLog` is $O(m|\mathcal{K}|)$.

(II) We then prove that the space complexity of a backward propagation step for `FastLog` is $O(mNL(|\mathcal{K}|+|\mathcal{E}|))$. For a backward propagation step, `FastLog` requires storing the intermediate hidden states for all L steps for all N rules to calculate the gradients. It also requires storing the intermediate estimated truth degrees for all L steps for all N rules to calculate the gradients. Therefore, the space complexity of a backward propagation step for `TensorLog` is $O(mNL(|\mathcal{K}| + |\mathcal{E}|))$. □

A.13 Proof of Proposition 13

Proof. (I) From Proposition 4, we know that the time complexity of a forward computation step for `FastLog` is $O(NL|\mathcal{K}|)$. From Equation (29), we know that the dynamic pruning strategy introduces an additional complexity of $O(NL|\mathcal{E}|)$ to calculate top-c_1 intermediate estimated truth degrees. From Equation (30), we know that the dynamic pruning strategy introduces an additional complexity of $O(NL|\mathcal{K}|)$ to calculate top-c_2 intermediate hidden states. Therefore, the time complexity of a forward computation step for $\texttt{FastLog}^{c_1,c_2}$ is $O(NL(|\mathcal{K}| + |\mathcal{E}|))$.

(II) From Equations (29), we know that only top-c_1 intermediate estimated truth degrees are used to calculate the gradients. From Equations (30), we know that only top-c_2 intermediate hidden states are used to calculate the gradients. Let $z = \hat{\mathcal{F}}^{c_2}_{\text{r2f}}(\hat{\mathcal{F}}^{c_1}_{\text{e2f}}(\phi^{(k,l-1)}_{r,x}), w^{(r,k,l)})$. The time complexity for calculating $\frac{\partial \hat{\mathcal{F}}_{\text{f2e}}(z)}{\partial z}$ is $O(c_2)$ because z only has c_2 elements. The time complexity for calculating $\frac{\partial z}{\partial w^{(r,k,l)}}$ is $O(c_2)$ because only the top-c_2 elements in $\hat{\mathcal{F}}^{c_1}_{\text{e2f}}(\phi^{(k,l-1)}_{r,x})$ are used to calculate gradients. The time complexity for calculating $\frac{\partial \hat{\mathcal{F}}^{c_1}_{\text{e2f}}(\phi^{(k,l-1)}_{r,x})}{\partial \phi^{(k,l-1)}_{r,x}}$ is $O(c_2)$ because only the top-c_2 elements in $\hat{\mathcal{F}}^{c_1}_{\text{e2f}}(\phi^{(k,l-1)}_{r,x})$ are used to calculate gradients. Therefore, the time complexity of a backward propagation step for `FastLog` is reduced to $O(NLc_2)$. □

A.14 Proof of Proposition 14

Proof. (I) We first prove that the space complexity of a forward computation step for `FastLog` is $O(m|\mathcal{K}|)$. For all $1 \le k \le N$, $1 \le l \le L$, `FastLog` requires a space of the size $m|\mathcal{K}|$ to store the intermediate hidden state for all facts. Although `FastLog` also requires a space of $m|\mathcal{E}|$ to store the intermediate estimated truth degrees, it can reuse the previously opened space. In general, it holds that $|\mathcal{K}| > |\mathcal{E}|$. Therefore, a forward computation step for `FastLog` is $O(m|\mathcal{K}|)$.

(II) We then prove that the space complexity of a backward propagation step for `FastLog` is $O(mNL(c_1 + c_2))$. For a backward propagation step, `FastLog` requires storing the intermediate estimated truth degrees with the size of c_1 for all L steps for all N rules to calculate the gradients. It also requires storing the intermediate hidden states with the size of c_2 for all L steps for all N rules to calculate the gradients. Therefore, the space complexity of a backward propagation step for `TensorLog` is $O(mNL(c_1 + c_2))$. □

A.15 Proof of Proposition 15

From Equations (15-17) and (29-31), we know that $\hat{\mathcal{F}}^{|\mathcal{E}|}_{\text{e2f}}$ (resp. $\hat{\mathcal{F}}^{|\mathcal{K}|}_{\text{r2f}}$ or $\hat{\mathcal{F}}_{\text{f2e}}$) is equivalent to $\mathcal{F}_{\text{e2f}}$ (resp. $\mathcal{F}_{\text{r2f}}$ or $\mathcal{F}_{\text{f2e}}$) because both $\mathcal{T}^{|\mathcal{E}|}(\mathbb{T})$ and $\mathcal{T}^{|\mathcal{K}|}(\mathbb{T})$ return the original set $\mathbb{T}$ of tuples. Therefore, Equation (18) can be derived by:

$$
\begin{aligned}
\texttt{FastLog}(\theta^{N,L}_r, a, b) \quad &= \quad \sum_{k=1}^{N} \phi^{(k,L)}_{r,a} v_b \\
&= \quad \sum_{k=1}^{N} \mathcal{F}_{\text{f2e}}(\mathcal{F}_{\text{r2f}}(w^{(r,k,L)}) \odot \mathcal{F}_{\text{e2f}}(\\
&\quad \cdots \\
&\quad \mathcal{F}_{\text{f2e}}(\mathcal{F}_{\text{r2f}}(w^{(r,k,2)}) \odot \mathcal{F}_{\text{e2f}}(\\
&\quad \mathcal{F}_{\text{f2e}}(\mathcal{F}_{\text{r2f}}(w^{(r,k,1)}) \odot \mathcal{F}_{\text{e2f}}(v_x^\top)))) \cdots))v_b \\
&= \quad \sum_{k=1}^{N} \hat{\mathcal{F}}_{\text{f2e}}(\hat{\mathcal{F}}^{|\mathcal{K}|}_{\text{r2f}}(\hat{\mathcal{F}}^{|\mathcal{E}|}_{\text{e2f}}(\\
&\quad \cdots \\
&\quad \hat{\mathcal{F}}_{\text{f2e}}(\hat{\mathcal{F}}^{|\mathcal{K}|}_{\text{r2f}}(\hat{\mathcal{F}}^{|\mathcal{E}|}_{\text{e2f}}(v_x^\top), w^{(r,k,1)})), \cdots), \\
&\quad w^{(r,k,L)})) \\
&= \quad \texttt{FastLog}^{|\mathcal{E}|,|\mathcal{K}|}(\theta^{N,L}_r, a, b)
\end{aligned}
$$