Algorithm 1 LEARN$((\Delta_p, \Delta_n), H)$

Input: multi-relational dataset (Δ_p, Δ_n) and hyperparameters $H = \{D, \gamma, \gamma', C_e, C_n, T, \alpha_0\}$.
Output: model (A, R).
Initialize: randomly initialize A, R to satisfy Eqs. (1) and (2), and set $\bar{A} = A$, $\bar{R} = R$, $t = 1$, $\alpha = \alpha_0$.

repeat
  Sample $v \sim$ Bernoulli(0.5)
  if $v = 1$ **then**
    Sample $t_p = (i_p, j_p, k_p)$ from Δ_p and $\bar{t}_p = (\bar{i}_p, \bar{j}_p, \bar{k}_p)$ from $\Delta \backslash \Delta_p$.
    Compute indicators as

$$I_1 \leftarrow H\left(\gamma - \mathbf{a}_{i_p}^\top R_{k_p} \mathbf{a}_{j_p} + \mathbf{a}_{\bar{i}_p}^\top R_{\bar{k}_p} \mathbf{a}_{\bar{j}_p}\right),$$
$$I_2 \leftarrow H\left(\gamma' - \mathbf{a}_{i_p}^\top R_{k_p} \mathbf{a}_{j_p}\right).$$

    Update model parameters as

$$\mathbf{a}_{i_p} \leftarrow \mathbf{a}_{i_p} - \alpha\left[-(I_1 + I_2 C_e)\bar{R}_{k_p}\bar{\mathbf{a}}_{j_p}\right],$$
$$\mathbf{a}_{j_p} \leftarrow \mathbf{a}_{j_p} - \alpha\left[-(I_1 + I_2 C_e)\bar{R}_{k_p}^\top\bar{\mathbf{a}}_{i_p}\right],$$
$$R_{k_p} \leftarrow R_{k_p} - \alpha\left[-(I_1 + I_2 C_e)\bar{\mathbf{a}}_{i_p}\bar{\mathbf{a}}_{j_p}^\top\right],$$
$$\mathbf{a}_{\bar{i}_p} \leftarrow \mathbf{a}_{\bar{i}_p} - \alpha\left[I_1 \bar{R}_{\bar{k}_p}\bar{\mathbf{a}}_{\bar{j}_p}\right],$$
$$\mathbf{a}_{\bar{j}_p} \leftarrow \mathbf{a}_{\bar{j}_p} - \alpha\left[I_1 \bar{R}_{\bar{k}_p}^\top\bar{\mathbf{a}}_{\bar{i}_p}\right],$$
$$R_{\bar{k}_p} \leftarrow R_{\bar{k}_p} - \alpha\left[I_1 \bar{\mathbf{a}}_{\bar{i}_p}\bar{\mathbf{a}}_{\bar{j}_p}^\top\right].$$

  else if $v = 0$ **then**
    Sample $t_n = (i_n, j_n, k_n)$ from Δ_n and $\bar{t}_n = (\bar{i}_n, \bar{j}_n, \bar{k}_n)$ from $\Delta \backslash \Delta_n$.
    Compute indicators as

$$I_3 \leftarrow H\left(\gamma - \mathbf{a}_{\bar{i}_n}^\top R_{\bar{k}_n} \mathbf{a}_{\bar{j}_n} + \mathbf{a}_{i_n}^\top R_{k_n} \mathbf{a}_{j_n}\right),$$
$$I_4 \leftarrow H\left(\gamma' + \mathbf{a}_{i_n}^\top R_{k_n} \mathbf{a}_{j_n}\right).$$

    Update model parameters as

$$\mathbf{a}_{i_n} \leftarrow \mathbf{a}_{i_n} - \alpha\left[(I_3 C_n + I_4 C_e)\bar{R}_{k_n}\bar{\mathbf{a}}_{j_n}\right],$$
$$\mathbf{a}_{j_n} \leftarrow \mathbf{a}_{j_n} - \alpha\left[(I_3 C_n + I_4 C_e)\bar{R}_{k_n}^\top\bar{\mathbf{a}}_{i_n}\right],$$
$$R_{k_n} \leftarrow R_{k_n} - \alpha\left[(I_3 C_n + I_4 C_e)\bar{\mathbf{a}}_{i_n}\bar{\mathbf{a}}_{j_n}^\top\right],$$
$$\mathbf{a}_{\bar{i}_n} \leftarrow \mathbf{a}_{\bar{i}_n} - \alpha\left[-I_3 C_n \bar{R}_{\bar{k}_n}\bar{\mathbf{a}}_{\bar{j}_n}\right],$$
$$\mathbf{a}_{\bar{j}_n} \leftarrow \mathbf{a}_{\bar{j}_n} - \alpha\left[-I_3 C_n \bar{R}_{\bar{k}_n}^\top\bar{\mathbf{a}}_{\bar{i}_n}\right],$$
$$R_{\bar{k}_n} \leftarrow R_{\bar{k}_n} - \alpha\left[-I_3 C_n \bar{\mathbf{a}}_{\bar{i}_n}\bar{\mathbf{a}}_{\bar{j}_n}^\top\right].$$

  end if
  for $i \in \mathcal{E}$ such that $\mathbf{a}_i \neq \bar{\mathbf{a}}_i$ **do**
    $\mathbf{a}_i \leftarrow \mathbf{a}_i / \|\mathbf{a}_i\|_2$.
    $\bar{\mathbf{a}}_i \leftarrow \mathbf{a}_i$.
  end for
  for $k \in \mathcal{R}$ such that $R_k \neq \bar{R}_k$ **do**
    $U_k, \Sigma_k, V_k \leftarrow \mathrm{SVD}(R_k)$.
    $R_k \leftarrow U_k V_k^*$.
    $\bar{R}_k \leftarrow R_k$.
  end for
  $\alpha \leftarrow \alpha_0 / \sqrt{t+1}$.
  $t \leftarrow t + 1$.
until $t \geq T$ holds.
return (A, R)

Algorithm 2 LEARNDEFAULT$((\Delta_p, \Delta_n), (\Delta_p^{(v)}, \Delta_n^{(v)}), \mathcal{H})$

Input: training dataset (Δ_p, Δ_n), validation dataset $(\Delta_p^{(v)}, \Delta_n^{(v)})$, and a set of models $\mathcal{H} = \{H^{(l)} \mid l \in \{1, \ldots, L\}\}$.
Output: default model (A, R).

for $l = 1, \ldots, L$ **do**
  $(A^{(l)}, R^{(l)}) \leftarrow$ LEARN$(\Delta_p, \Delta_n, H^{(l)})$.
  $v^{(l)} \leftarrow$ VALSCORE$(A^{(l)}, R^{(l)}, \Delta_p^{(v)}, \Delta_n^{(v)})$.
end for
$l^\star \leftarrow \arg\max_{l=1,\ldots,L} v^{(l)}$.
return $(A^{(l^\star)}, R^{(l^\star)})$.

Algorithm 3 Active Multi-relational Data Construction

Input: budget $B > 0$, budget for validation $N_{\mathrm{val}} > 0$, initial dataset $(\Delta_p^{(0)}, \Delta_n^{(0)})$, a set of hyperparameters $\mathcal{H} = \{H^{(l)} \mid l \in \{1, \ldots, L\}\}$, the number of queries q, the number of unlabeled triples to calculate a query score Q, and a query score function.
Output: dataset (Δ_p, Δ_n) and model $(A^\star, R^\star)$.
Initialization:
$(\Delta_p, \Delta_n) \leftarrow (\Delta_p^{(0)}, \Delta_n^{(0)})$.
Create a validation dataset of size N_{val}, $(\Delta_p^{(v)}, \Delta_n^{(v)})$.
$(A^\star, R^\star) \leftarrow$ LEARNDEFAULT$((\Delta_p, \Delta_n), (\Delta_p^{(v)}, \Delta_n^{(v)}), \mathcal{H})$.
$t \leftarrow 0$.
Main algorithm:
while $tq \leq B$ **do**
  Randomly sample Q unlabeled triples to construct Δ_u.
  for each triple $t \in \Delta_u$ **do**
    Compute the query score q_t using the query score function.
  end for
  Choose q triples with the lowest query scores to construct $\Delta_q^\star$.
  for each triple $t \in \Delta_q^\star$ **do**
    $l \leftarrow \mathcal{O}(t)$.
    if $l = 1$ **then**
      $\Delta_p \leftarrow \Delta_p \cup \{t\}$.
    else
      $\Delta_n \leftarrow \Delta_n \cup \{t\}$.
    end if
  end for
  $(A^\star, R^\star) \leftarrow$ LEARNDEFAULT$((\Delta_p, \Delta_n), (\Delta_p^{(v)}, \Delta_n^{(v)}), \mathcal{H})$.
  $t \leftarrow t + 1$.
end while
return (Δ_p, Δ_n) and $(A^\star, R^\star)$.