+ m tasks: $\{t_j\}_{j \in T} \mid T = \{1, 2, \ldots, m\}$

+ categorize → relational task
→ meta task

+ Supervised

$$\begin{cases} \text{Train}: \{x^{(i)} \rightarrow \{y_j^{(i)} \sim t_j\}_{j \in T}\} \\ \text{Test}: \{x^{(i)} \rightarrow \{y_j^{(i)} \sim t_j\}_{j \in T}\} \end{cases}$$

+ Relational

$$\begin{cases} \text{Train}: \{(x^{(i)}, \{y_j^{(i)}\}_{j \in T_{aux}^{(i)}}) \rightarrow \{y_j^{(i)}\}_{j \in T_{test}^{(i)}}\} \\ \text{Test}: \{(x^{(i)}, \{y_j^{(i)}\}_{j \in T_{aux}^{(i)}}) \rightarrow \{y_j^{(i)}\}_{j \in T_{test}^{(i)}}\} \end{cases}$$

tasks x has accesses to. → $T_{aux} \cap T_{test} = \emptyset$ → accesses to y

---

+ Meta

↳ Train: $S = \{(x^{(i)}, \{y_j^{(i)}\}_{j \in T_S})\} \Rightarrow Q = \{x^{(i)} \rightarrow \{y_j^{(i)}\}_{j \in T_S}\}$ (T_S → seen)

↳ Test: $S = \{(x^{(i)}, \{y_j^{(i)}\}_{j \in T_u})\} \Rightarrow Q = \{x^{(i)} \rightarrow \{y_j^{(i)}\}_{j \in T_u}\}$ (T_u → unseen.)

+ Relational Meta:

↳ Train: ⟨seen⟩

$S = \{(x^{(i)}, \{y_j^{(i)}\}_{j \in T_S^{(i)}})\} \Rightarrow$ predict $Q = \{(x^{(i)}, \{y_j^{(i)}\}_{j \in T_{aux}^{(i)}}) \rightarrow \{y_j^{(i)}\}_{j \in T_S^{(i)} \mid T_{aux}^{(i)}}\}$

$x \rightarrow T \nrightarrow y$ ↑ $x \not\rightarrow T \rightarrow y$ ↑

$x \rightarrow T \rightarrow y$

↳ Test ⟨unseen⟩

$S = \{(x^{(i)}, \{y_j^{(i)}\}_{j = T_u^{(i)}})\} \Rightarrow$ predict $Q = \{(x^{(i)}, \{y_j^{(i)}\}_{j \in T^{(i)}}) \rightarrow \{y_j^{(i)}\}$

$\langle T_{aux} \subseteq T_S \rangle$ ↑ unseen ↑

MetaLink:

↳ Data $\{x^{(i)}\}_{i=1}^{n}$ // $\{\{y_j^{(i)}\}_{j\in T}\}_{j=1}^{n}$ (label) | KG on task heads |

↳ Model: f_θ : encoder // f_w : task heads // f_ϕ : predictive model.

$x^{(i)} \xrightarrow{f_\theta(\cdot)} z^{(i)} \xrightarrow[w^T \cdot z^{(i)}]{f_w} y^{(i)}$ / m tasks: $y_j^{(i)} = w_{j\cdot}^T z^{(i)}$

↳ Knowledge graph: task weights: w_j / embeddings: $z^{(i)}$

↳ Graph: $G = \{V, E\}$ → V: node set → V_d: data nodes, V_t: task nodes

→ E: edge set $(E \subseteq V \times V)$

$E_{dd} \subseteq V_d \times V_d$

$E_{dt} \subseteq V_d \times V_t$

$E_{tt} \subseteq V_{tt}$

$E_{dt} = \{\{x^{(i)}, t_j\} \sim y^{(i)}_{j\in T_{aux}}\}$

Initialize Nodes

+ $z^{(i)} = f_\theta(x^i) = h_i^{(0)}$

+ $h_j^{(0)} = w_j$ (node embeddings)

+ $w_j \mid j \in T_u$ (init with $w_{ju} = 1$)

MetaLink ⇓ generalize to unseen tasks.

Edges (predict)

Edge predictor (f_ϕ):

$$h_v^{(l)} = AGG^{(l)}\left(\{MSG^{(l)}(h_u^{(l-1)}),\ u \in N_G(v)\}, h_v^{(l)}\right)$$

(iterations, neighbor $\in v$)

$(h_i / h_v \in 2$ different graphs$)$ ⇓ (L-GNN layers)

$$\hat{y}_j^{(i)} = MLP(CONCAT(h_i^{(L)}, h_j^{(L)}))$$

$$h_v^{(1)} = U^{(1)} \cdot \underline{CONCAT}(MEAN(\{Relu(W^{(1)} h_u^{(l-1)}),\ u \in N(v)\}), h_v^{(l-1)})$$

trainable / ↓ info loss | trainable | ↓ info loss

+ 2 types of nodes: data V_d / tasks V_t.
+ Different MSG passing / add edge features → msg computation trainable / allows $y_r^{(u)}$ → MSG

$$h_v^{(l)} = U^{(l)} \cdot \text{CONCAT}\left(\text{Mean}\left(\left\{\text{RELU}\left(W^{(l)}_{\mathbb{1}[v \in V_d,\ u \in V_t]} \cdot h_u^{(l-1)} + O^{(l)} \cdot y_v^{(u)}\right),\ u \in N(v)\right\}\right),\ h_v^{(l-1)}\right)$$

2 sets of params [1: data → task; 0: task → data]

+ Each time → sample new batch → new graph $G = \{V, E\}$
+ Edge: each data point → all T tasks ⟨bipartite graph⟩

$V = \{V_d, V_t\}$

$\mathbb{1}: 1 \to$ ⟨batch⟩ $\quad j: 0 \to T$

Meta task = in + tasks?

+ backwards:
  ↳ Compare ⟨logits ⟨edge pred⟩⟩ $\simeq \{y_j^{(i)}\}_{j \in T_s^{(i)} \mid T_{aux}^{(i)}}$
  * skip the real edge prediction
  * goal-oriented

$T_s^{(i)}$: seen tasks

$T_{aux}^{(i)}$: seen tasks has labels

Algorithm 1 MetaLink Training in Relational Meta Setting

Require: Dataset $\mathcal{D}_{\text{train}} = \{(\mathbf{x}, y)\}$. A parameterized embedding function f_θ. Last layer weights for each task $\mathbf{w}_j$. A parameterized heterogeneous GNN f_ϕ. Number of GNN layers L.

for each iteration **do**
  $S, Q \leftarrow \text{SampleMiniBatch}(\mathcal{D}_{\text{train}})$ ▷ Simulate meta setting in training
  $\{\mathbf{z}\} \leftarrow f_\theta(\mathbf{x})$ for $\mathbf{x} \in (S, Q)$
  $V_d^{(0)} = \{\mathbf{h}_i^{(0)} \leftarrow \mathbf{z} \text{ for } \mathbf{z} \in \{\mathbf{z}\}\}$ ▷ Initialize data nodes
  $V_t^{(0)} = \{\mathbf{h}_j^{(0)} \leftarrow \mathbf{1} \text{ if meta else } \mathbf{w}_j \text{ for each } \mathbf{w}_j\}$ ▷ Initialize task nodes
  $E = \{\mathbf{e}_{ij} \leftarrow (\mathbf{x}^{(i)}, t_j) \text{ for } y_j^{(i)} \in (S, Q)\}$ ▷ Initialize edges
  for $l = 1$ to L **do**
    $V_d^{(l)}, V_t^{(l)} \leftarrow \text{GraphConv}(V_d^{(l-1)}, V_t^{(l-1)}, E)$ with f_ϕ
  logits $\leftarrow \text{EdgePred}(V_d^{(L)}, V_t^{(L)})$ with f_ϕ
  Backward $\Big(\text{Criterion}(\text{logits}, \{\{y_j^{(i)}\}_{j \in T_s^{(i)} \setminus T_{\text{aux}}^{(i)}}\} \in Q)\Big)$