SMU SINGAPORE MANAGEMENT UNIVERSITY | School of Information Systems

Institute for Infocomm Research I²R

The FANG Lab @ SMU

Functional Algorithms for Networks and Graphs

Context-Aware Adapter Tuning for Few-Shot Relation Learning in Knowledge Graphs

Ran Liu, Zhongzhou Liu, Xiaoli Li, Yuan Fang

EMNLP'24 @ Miami

14 November 2024

Paper

- ❑ **Introduction**
- ❑ Methodology
- ❑ Experiment

Problem setting – Few-shot Relation Learning (FSRL)

Long-tail Problem (Sparse KG)

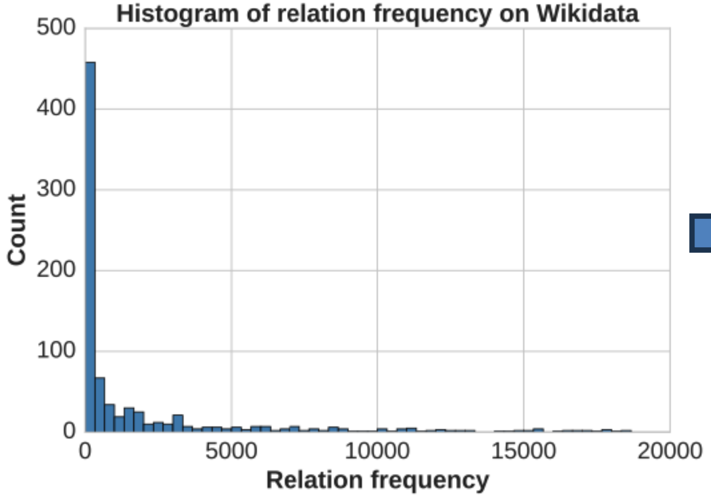
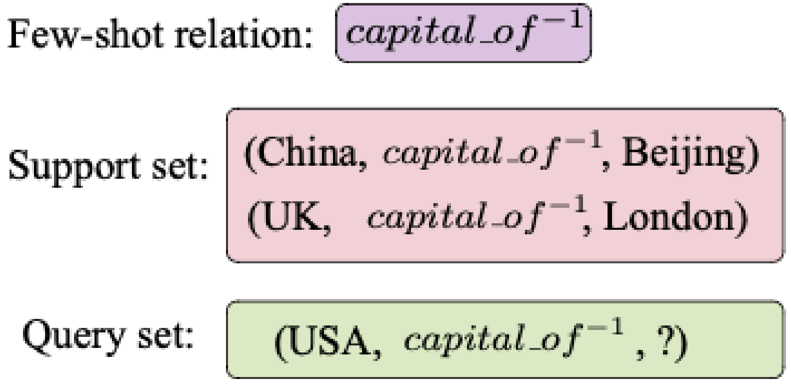


Figure 1: The histogram of relation frequencies in Wikidata. There are a large portion of relations that only have a few triples.

[Xiong, Wenhan and et al. One-shot relational learning for knowledge graphs. EMNLP 2018]



a. Few-shot knowledge graph completion (FKGC).

[Linhao Luo and et al. Normalizing Flow-based Neural Process for Few-Shot Knowledge Graph Completion. SIGIR 2023]

Meta-Learning Framework : MetaR

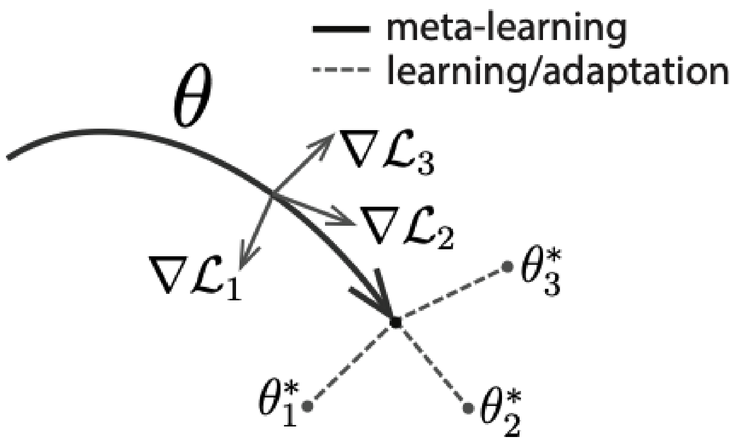


Figure 1. Diagram of our model-agnostic meta-learning algorithm (MAML), which optimizes for a representation θ that can quickly adapt to new tasks.

[Chelsea Finn and et al. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. ICML 2017]()

MetaR

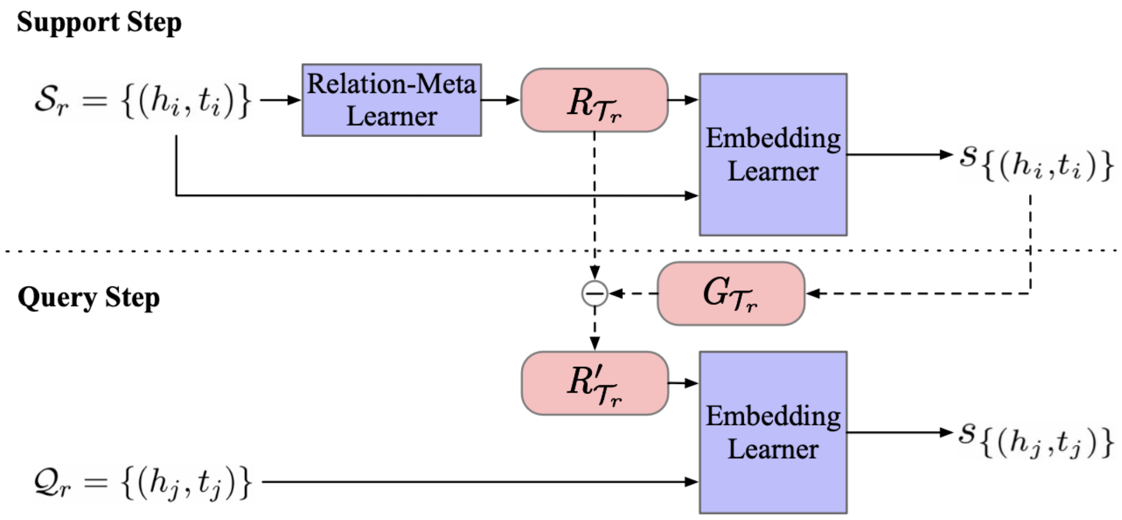


Figure 2: Overview of MetaR. $\mathcal{T}_r = \{\mathcal{S}_r, \mathcal{Q}_r\}$, $R_{\mathcal{T}_r}$ and $R'_{\mathcal{T}_r}$ represent relation meta and updated relation meta, and $G_{\mathcal{T}_r}$ represents gradient meta.

[Chen, Mingyang and et al. Meta Relational Learning for Few-Shot Link Prediction in Knowledge Graphs. EMNLP 2019]()

Assumption : independently and identically distributed (i.i.d.)

Limitations of Prior Work : Pilot test

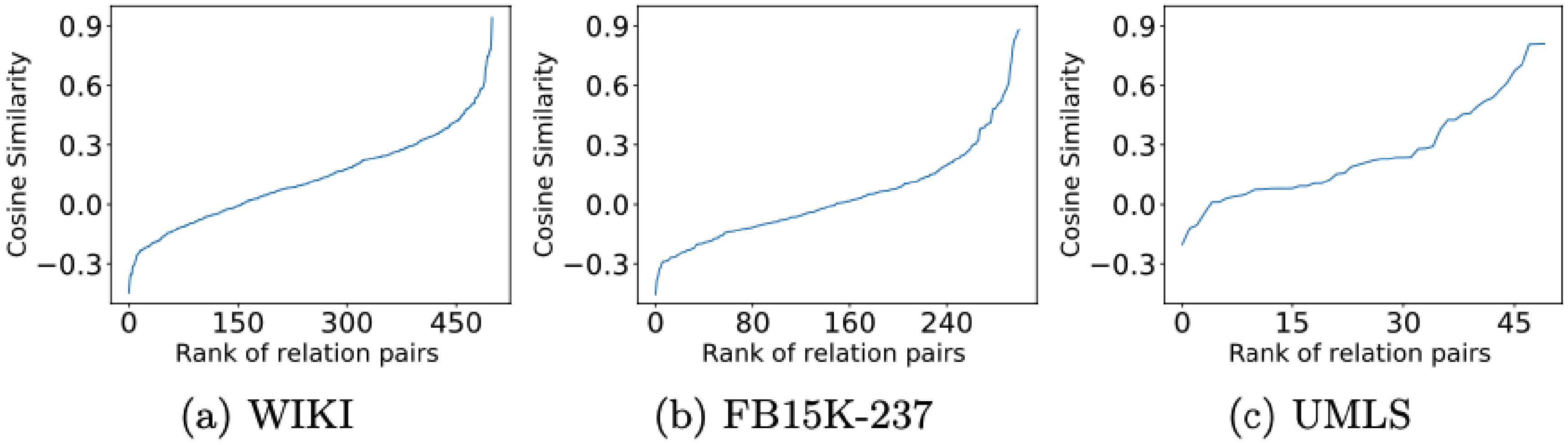


Figure 5: Pairwise cosine similarity of relations.

Problem : Performance degradation is anticipated unless the **Distribution Shift** between meta-training and meta-testing relations are addressed.

Contribution 1 – At model Level, design relation-specific adaptation to suit downstream task.
Contribution 2 – At data level, augment few-shot relation instances by injecting additional contextual information to enhance adaptation for downstream task.

- ❑ Introduction
- ❑ **Methodology**
- ❑ Experiment

Framework : RelAdapter

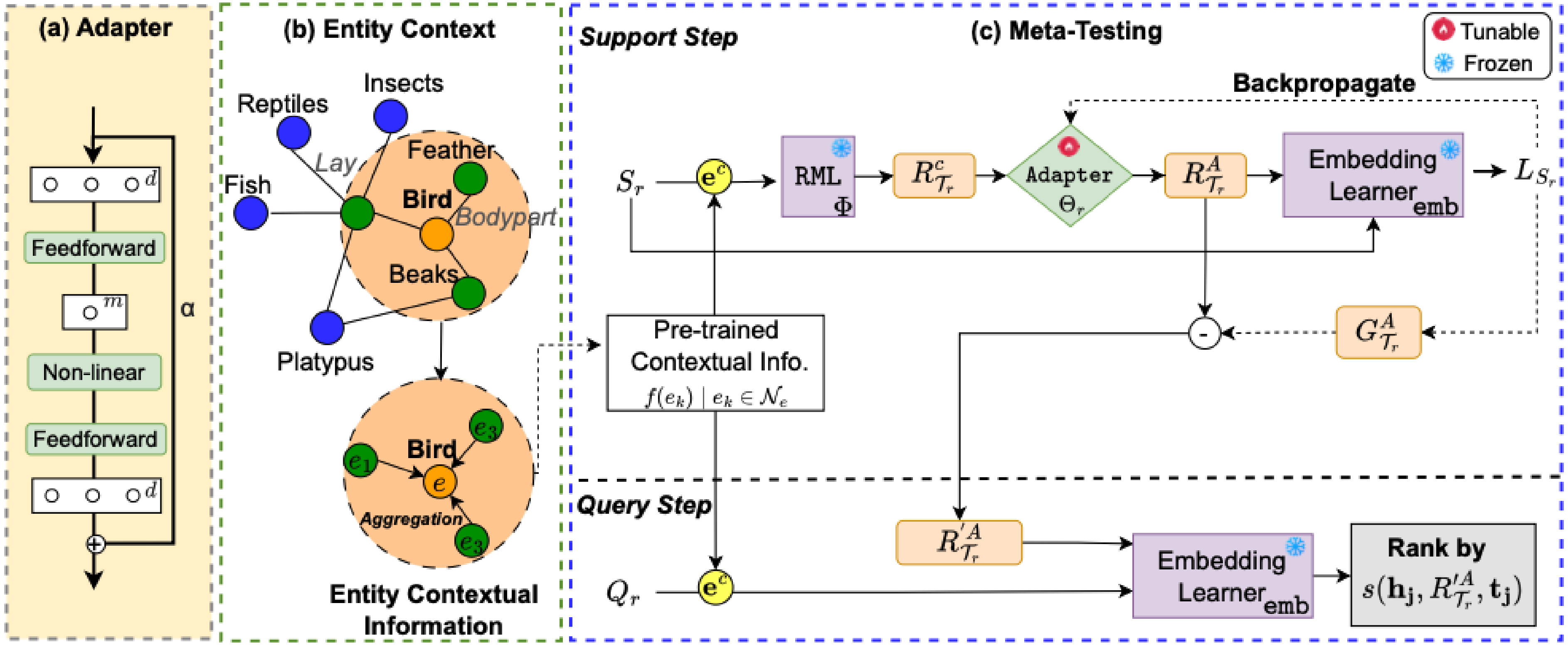
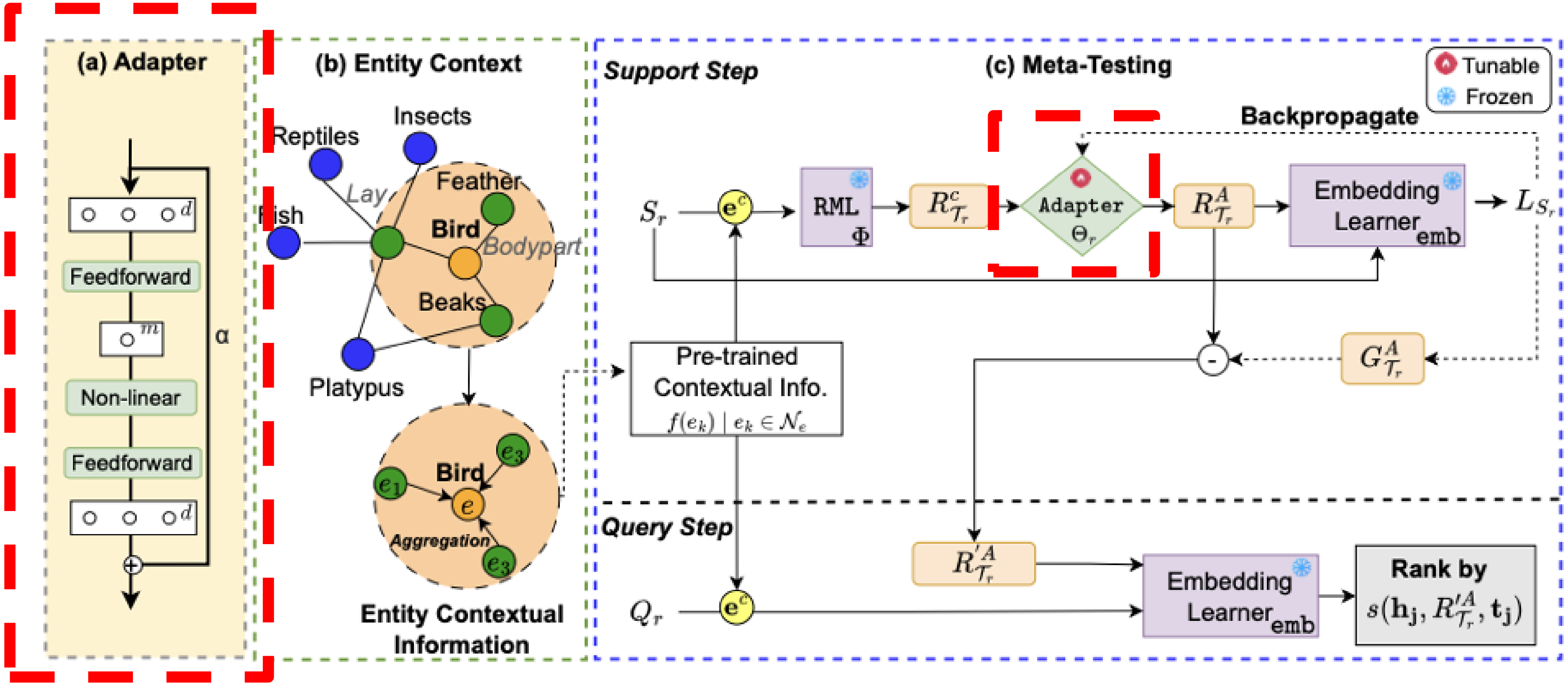


Figure 6: Illustration of key concepts in RelAdapter, hinging on an entity-aware adapter (a, b) in the meta-testing stage (c). Note that we omit the meta-training stage, which is similar to meta-testing but with backpropagation of the query loss to update the model parameters (emb and Φ).

Framework : Adapter Structure

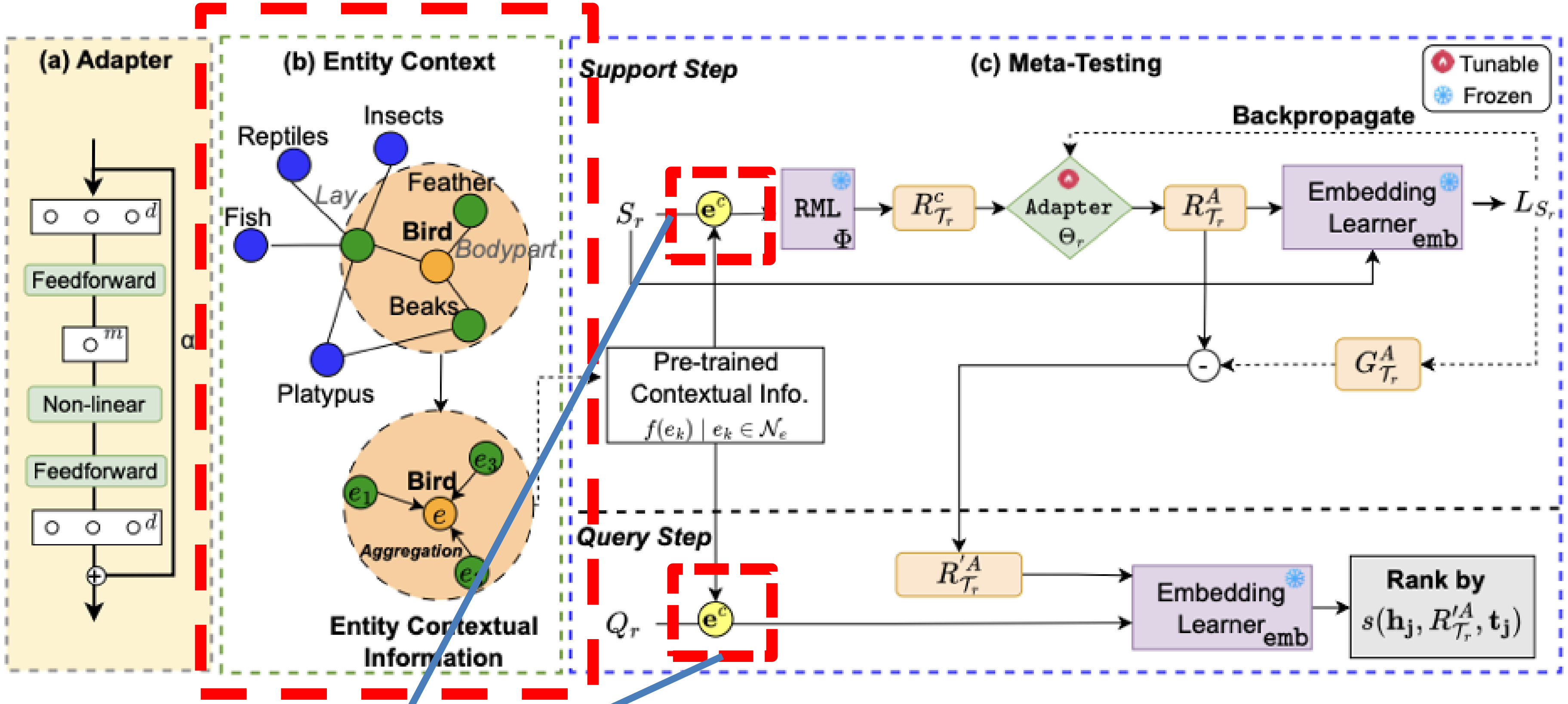


Residual network

$$R^A_{\mathcal{T}_r} = \texttt{Adapter}(R_{\mathcal{T}_r}; \Theta_r) = \alpha \cdot \underbrace{\texttt{FFN}(R_{\mathcal{T}_r}; \Theta_r) + (1-\alpha) \cdot R_{\mathcal{T}_r}}_{\text{Residual network}},$$

Lightweight feed-forward Network (FFN)

Framework : Context-aware adaptation



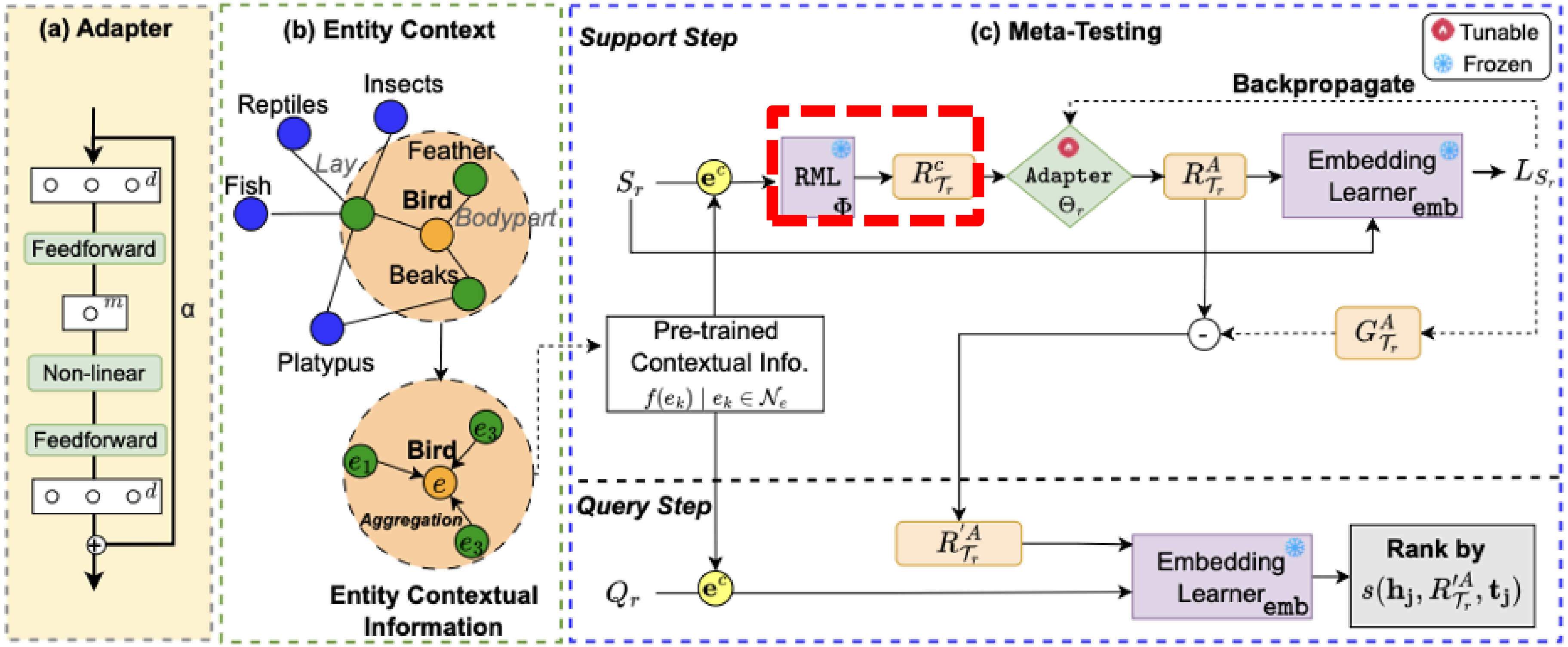
Pre-trained Contextual Info.

Meta-trained embedding

$$\mathbf{e}^c = \mu \cdot \mathtt{Mean}(\underbrace{\{f(e_k) \mid e_k \in \mathcal{N}_e\}}) + (1-\mu) \cdot \mathtt{emb}(e),$$

Augmented entity embedding

Framework : Context-aware relation meta



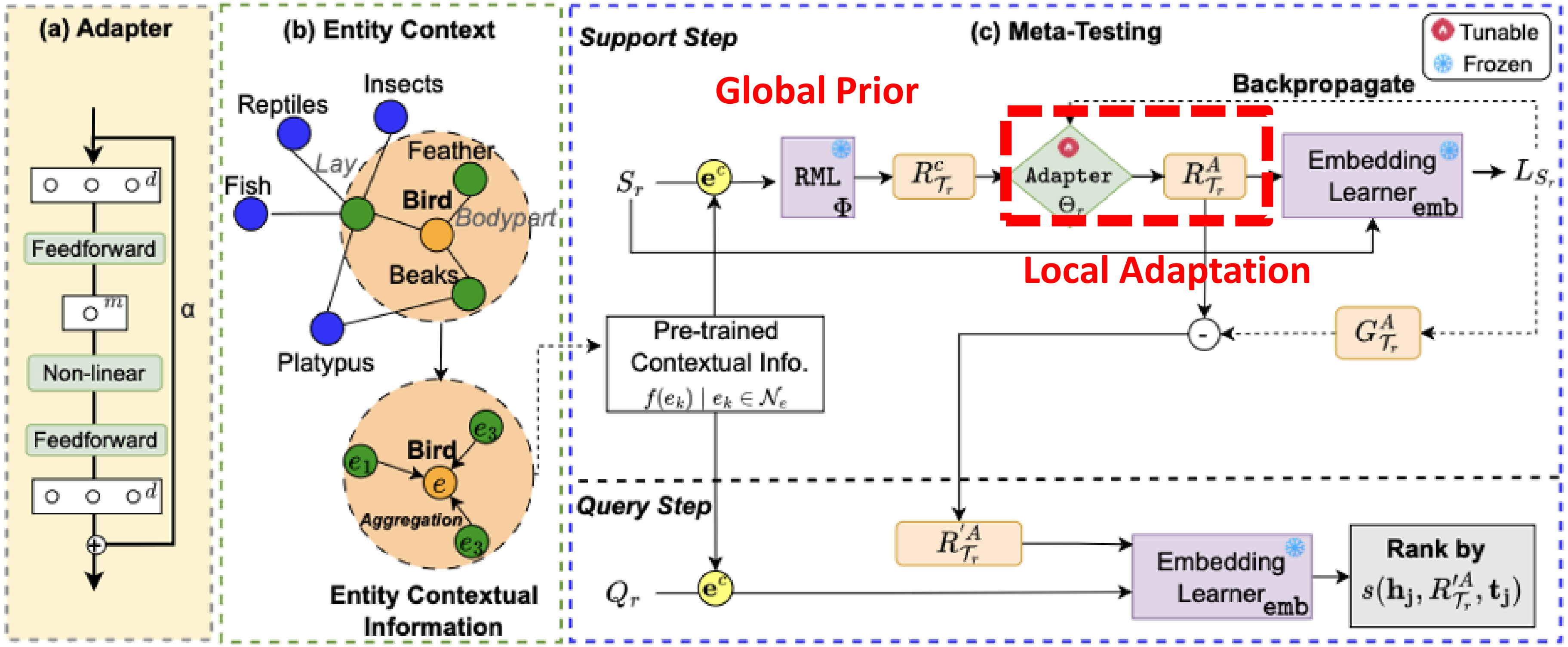
Derive context-aware relation meta

Given context-augmented entity embeddings, e^c

$$R^c_{\mathcal{T}_r} = \mathtt{Mean}(\{\mathtt{RML}(\mathbf{h}^c, \mathbf{t}^c; \Phi) \mid (h, r, t) \in \mathcal{S}_r\}).$$

Relation-Meta Learner : MLP(MeanPool[h,t])

Framework : Adaptation

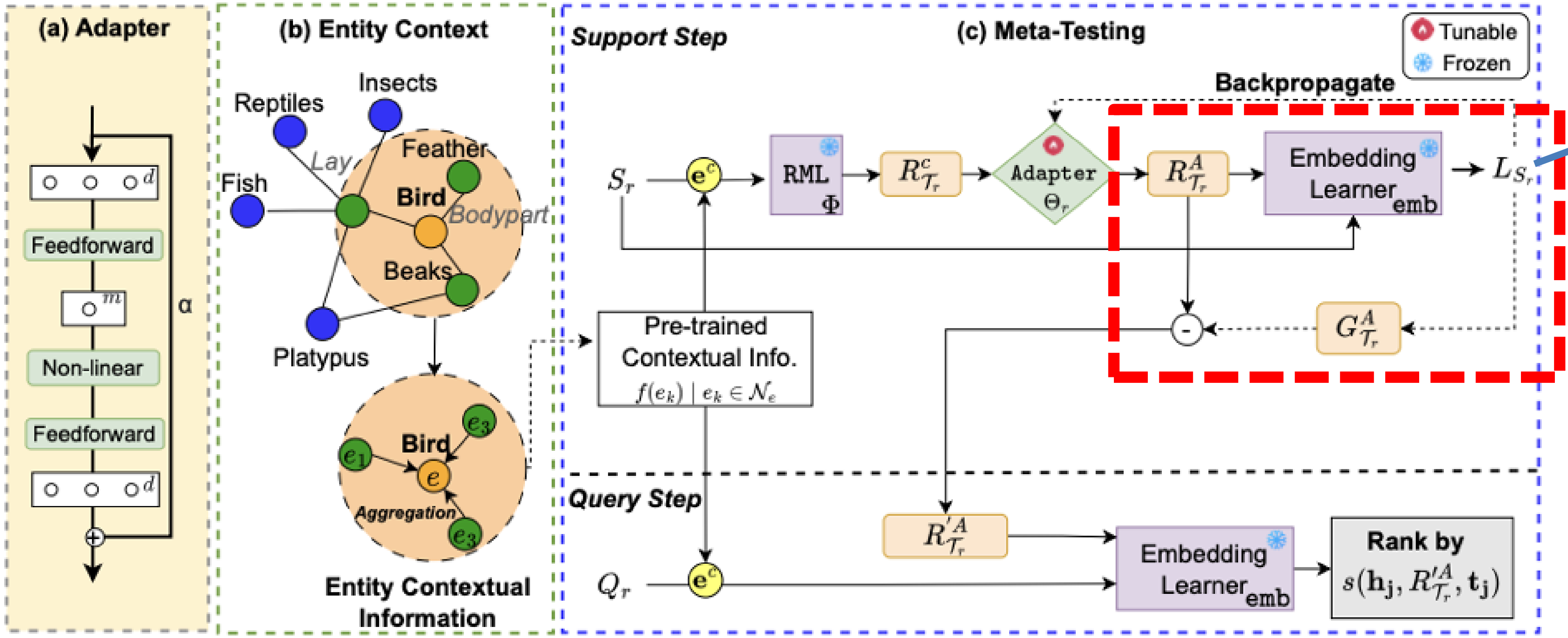


Adapted relation meta

$$R^{A}_{\mathcal{T}_r} = \texttt{Adapter}(R^{c}_{\mathcal{T}_r}; \Theta_r).$$

Adapter : Transform the context-aware relation meta

Framework : Gradient Update



Adapter is tuned on the Support Loss

Support Loss

$$L_{\mathcal{S}_r} = \sum_{(h,r,t)\in\mathcal{S}_r}[\gamma + s(\mathrm{emb}(h), R_{\mathcal{T}_r}, \mathrm{emb}(t)) - s(\mathrm{emb}(h), R_{\mathcal{T}_r}, \mathrm{emb}(t'))]_+,$$

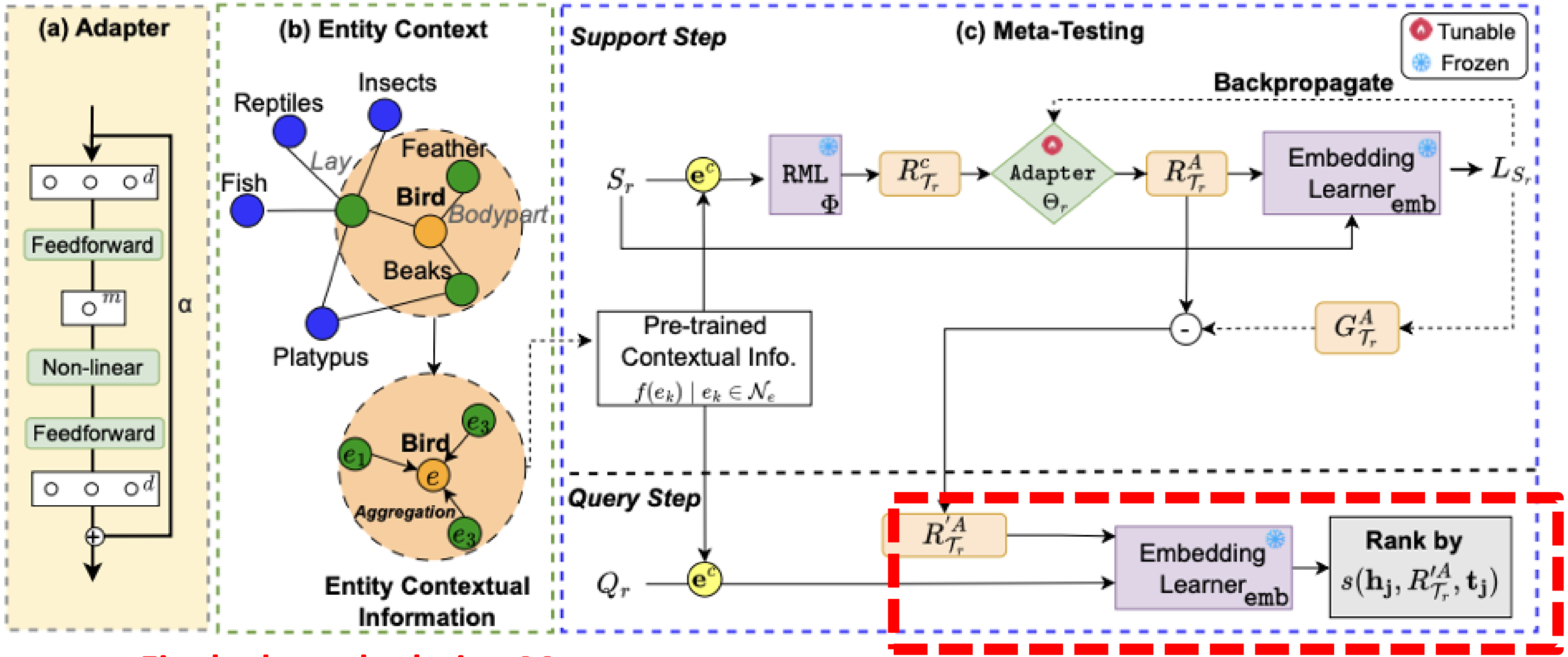
$$G^A_{\mathcal{T}_r} = \nabla_{R^A_{\mathcal{T}_r}} L_{\mathcal{S}_r},$$

$$R'^A_{\mathcal{T}_r} = \boxed{R^A_{\mathcal{T}_r} - \beta G^A_{\mathcal{T}_r}.}$$

Final adapted relation Meta
1. Relation specific
2. Context aware

Adapted by a quick gradient step

Framework : Scoring and Ranking



Final adapted relation Meta

$$s(\mathrm{emb}(h), R'^{A}_{\mathcal{T}_r}, \mathrm{emb}(t)) = \|\underbrace{\mathrm{emb}(h) + R'^{A}_{\mathcal{T}_r} - \mathrm{emb}(t))}_{\text{Ranking score}}\|,$$

- ❑ Introduction
- ❑ Methodology
- ❑ **Experiment**

Performance Comparison

Table 2: Performance comparison against baselines in the 3-shot setting. (Best: bolded, runners-up: underlined).

	Models	WIKI MRR	WIKI Hit@10	WIKI Hit@1	FB15K-237 MRR	FB15K-237 Hit@10	FB15K-237 Hit@1	UMLS MRR	UMLS Hit@10	UMLS Hit@1
Supervised Relation Learning	TransE	.031±.007	.043±.012	.021±.014	.294±.005	.437±.011	.204±.014	.178±.036	.310±.051	.146±.068
	DistMult	.047±.003	.082±.009	.031±.011	.234±.008	.364±.007	.208±.010	.231±.035	.337±.049	.214±.067
	ComplEx	.093±.004	.166±.011	.071±.012	.239±.007	.359±.010	.205±.013	.251±.038	.351±.041	.227±.058
	RGCN	.217±.012	.363±.023	.188±.031	.332±.011	.495±.013	.241±.031	.409±.059	.549±.072	.389±.089
Few-shot Relation Learning	GMatching	.133±.017	.331±.013	.114±.026	.309±.019	.441±.015	.245±.019	.296±.059	.532±.040	.257±.087
	FSKGC	.131±.003	.267±.010	.104±.016	.355±.005	.523±.004	.217±.011	.525±.031	.682±.024	.490±.038
	GANA	.291±.014	.384±.012	.272±.015	<u>.388</u>±.004	.553±.008	**.301**±.017	.541±.045	.721±.076	.502±.047
	FAAN	.278±.018	.421±.020	.275±.024	.363±.009	.542±.007	.279±.013	.545±.034	.746±.120	.505±.068
	HiRe	.300±.028	<u>.444</u>±.012	<u>.282</u>±.015	.378±.013	<u>.571</u>±.011	.281±.015	<u>.577</u>±.060	<u>.752</u>±.066	<u>.533</u>±.089
	MetaR	<u>.314</u>±.013	.420±.016	.274±.028	.368±.007	.536±.005	.251±.012	.435±.075	.601±.095	.417±.103
	RelAdapter	**.347**±.006	**.454**±.012	**.317**±.013	**.405**±.012	**.575**±.014	<u>.297</u>±.019	**.608**±.067	**.780**±.044	**.555**±.062

Average Improvement
MRR : 20.1%
Hit@10 : 15.1%
Hit@1 : 5.07%

1. **Relation specific**
2. **Context aware**

Additional Experiments – Efficiency Analysis

Table 7: Number of parameters of our adapter w.r.t. MetaR.

	WIKI	FB15K-237	UMLS
MetaR	241,967,556	1,650,206	234,306
Our adapter	5,125	5,125	5,125
% of MetaR	0.002	0.311	2.187

Negligible parameter overhead for Adapter

Table 8: Runtime (in seconds) for meta-training and meta-testing.

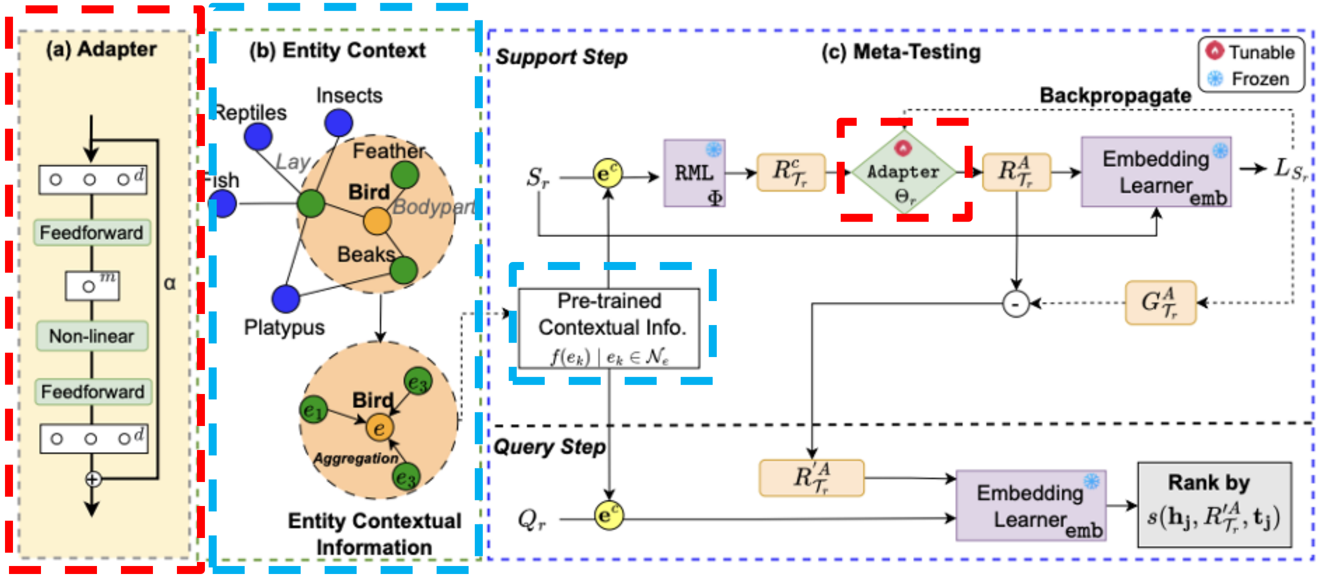
Meta-learning stage	Model	WIKI	FB15K-237	UMLS
Meta-train (total runtime)	MetaR	22,691	16,504	8,802
	RELADAPTER	24,085	17,529	9,656
Meta-test (per prediction)	MetaR	0.012	0.005	0.043
	RELADAPTER	0.045	0.008	0.053

Conclusion

Context-aware & Relation-specific adaptation

1. At the model level - Integrate a ***lightweight adapter module*** into the meta-learning framework to enable a relation-specific, **tunable adaptation** of the global prior to suit the local task in the meta-testing stage.

2. At the data level - Inject ***additional contextual information*** to enrich the few-shot relation instances through **data augmentation that enhances the adaptation** to meta-testing stage.



Q & A

Thank you !

Paper

https://arxiv.org/abs/2410.09123

Code

https://github.com/smufang/RelAdapter

Group

https://www.yfang.site/group