MetaGL: Evaluation-Free Selection of Graph Learning Models via Meta-Learning

Namyong Park[1] Ryan Rossi[2]

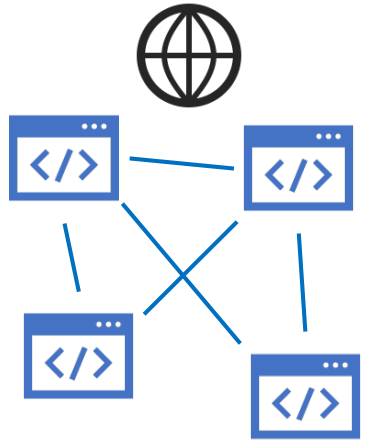
Nesreen Ahmed[3] Christos Faloutsos[1]

[1] Carnegie Mellon University Computer Science Department

[2] Adobe

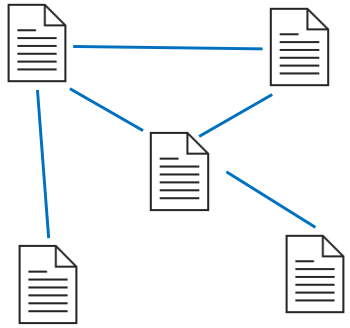
[3] intel labs

Social Networks

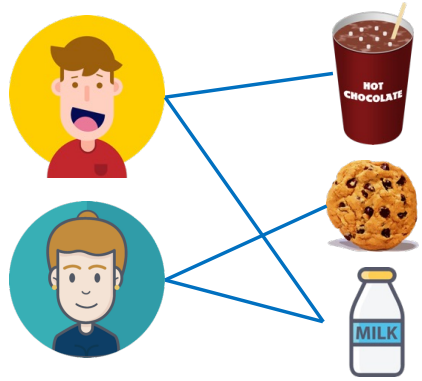


World Wide Web

Given a new graph,
how to find the best graph learning model
(e.g., link prediction model)?

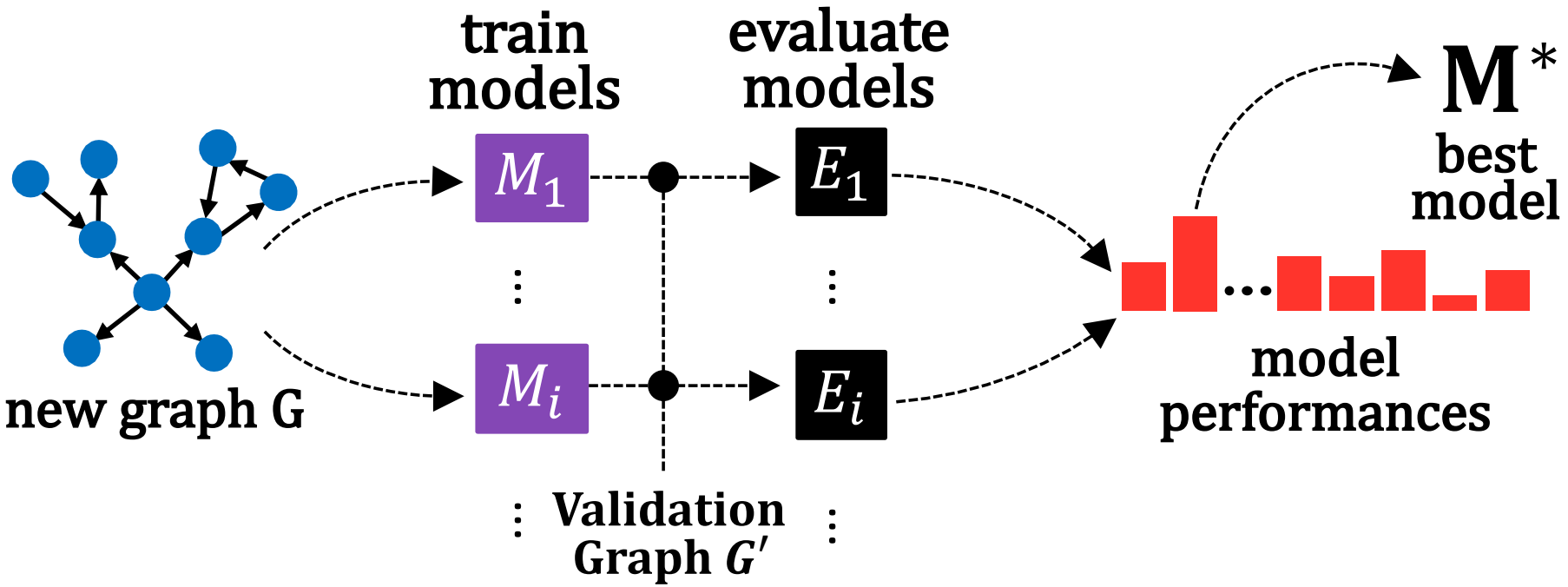


Citation Networks



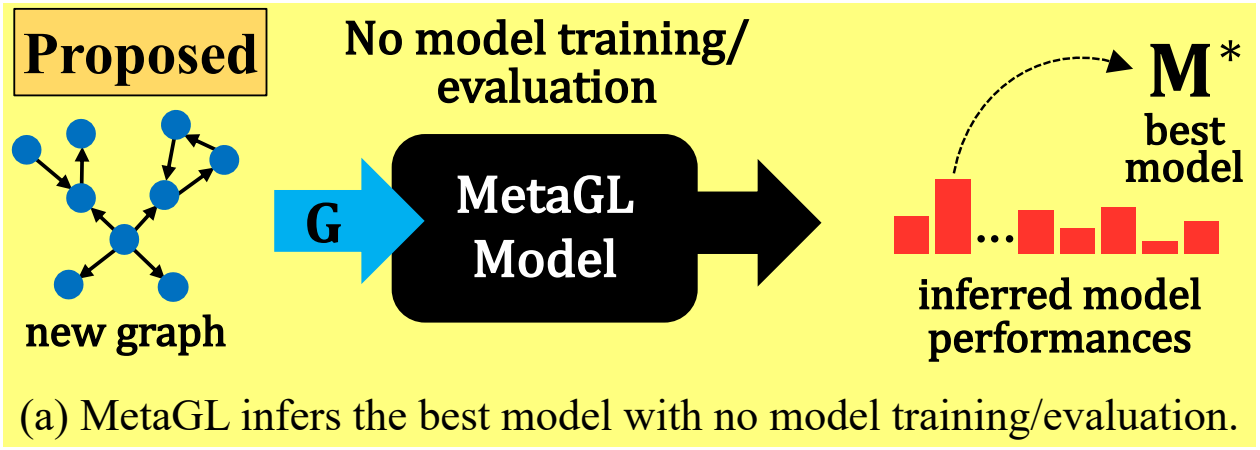
Purchase Graph

Given a new graph,
how to find the best graph learning model?

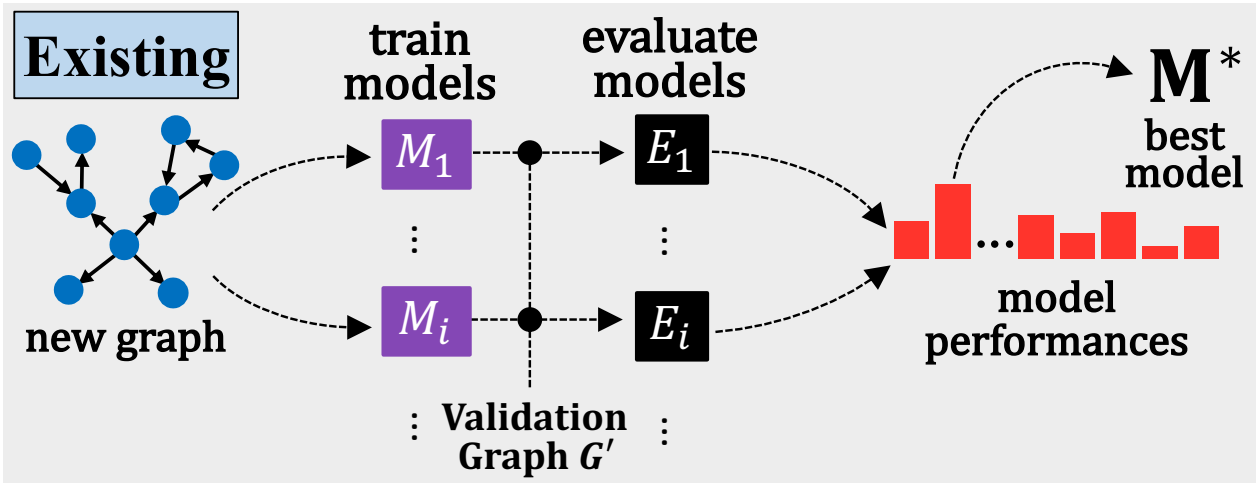


Existing model selection approach

Given a new graph, how to find the best graph learning model?

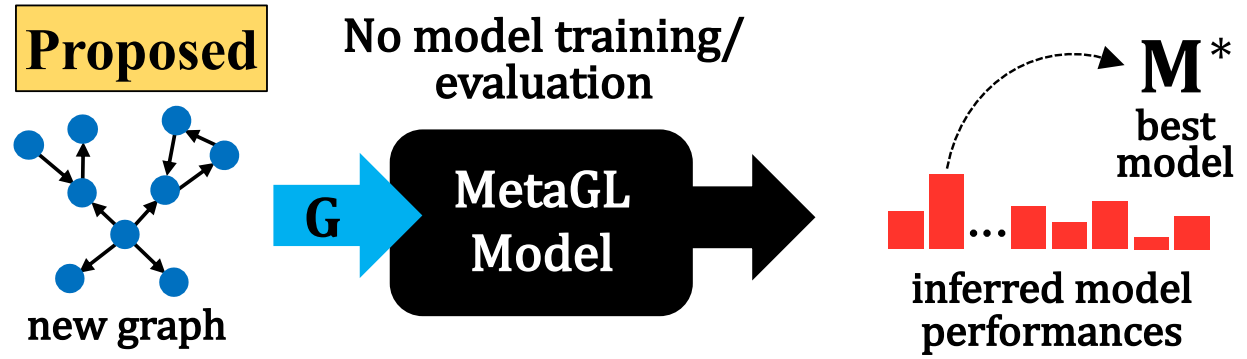


(a) MetaGL infers the best model with no model training/evaluation.

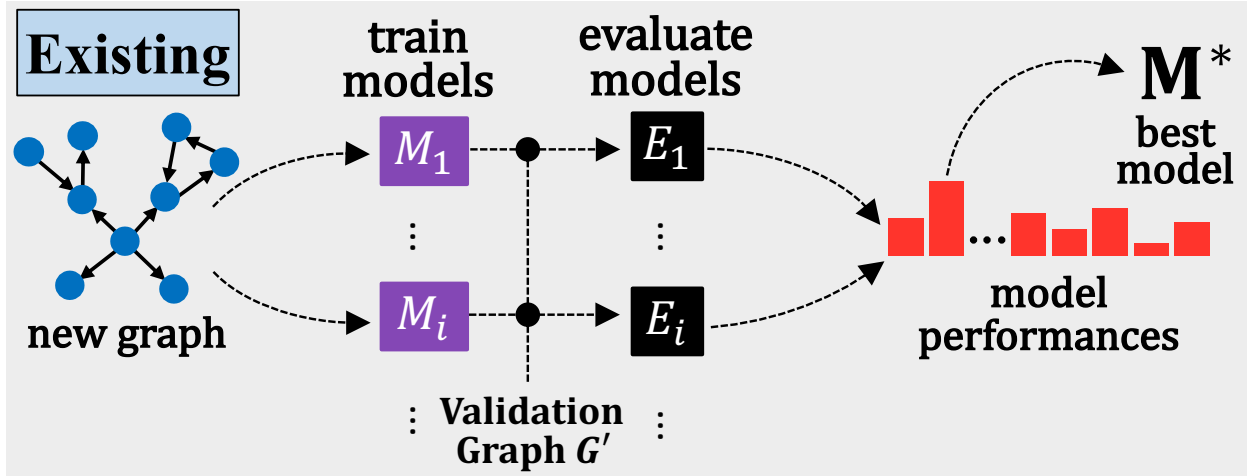


(b) Existing model selection methods train/evaluate multiple models.

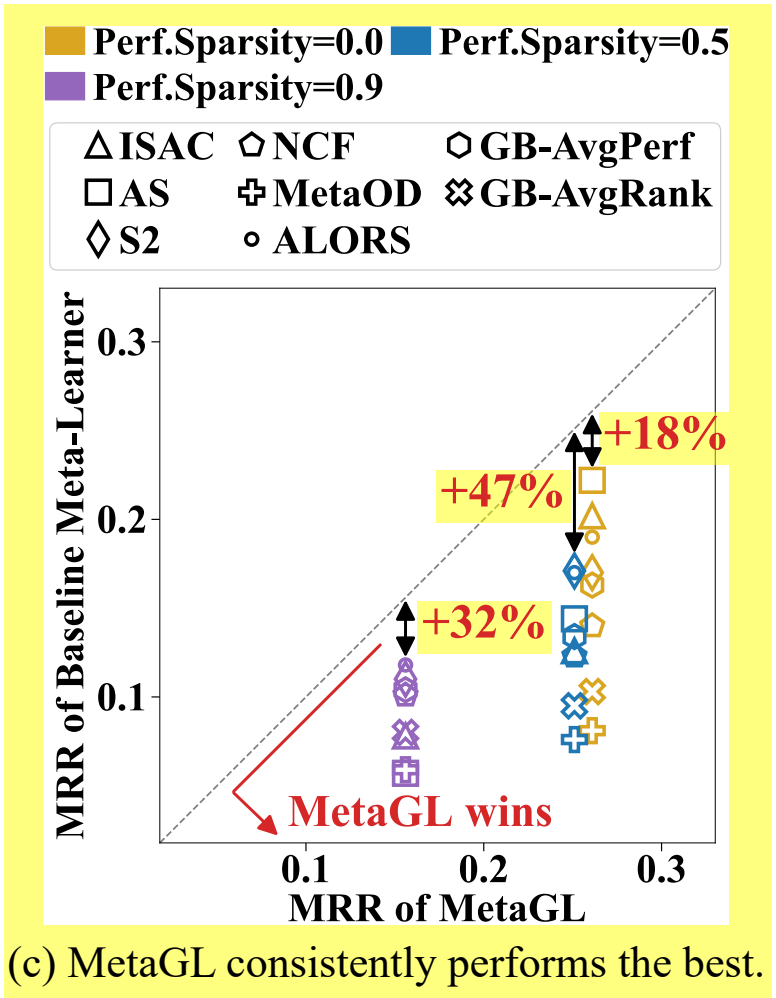
Given a new graph, how to find the best graph learning model?



(a) MetaGL infers the best model with no model training/evaluation.



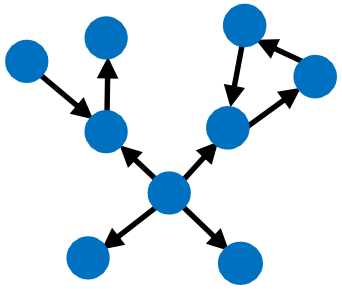
(b) Existing model selection methods train/evaluate multiple models.



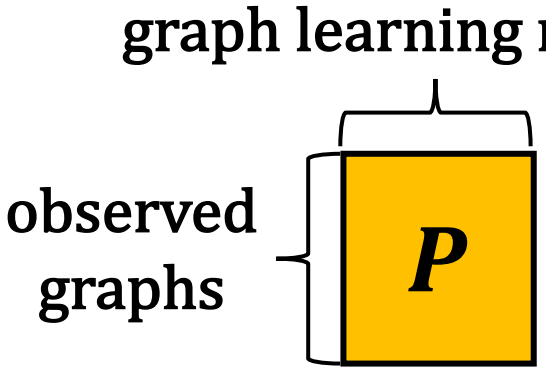
(c) MetaGL consistently performs the best.

Problem Formulation

Given



new graph G



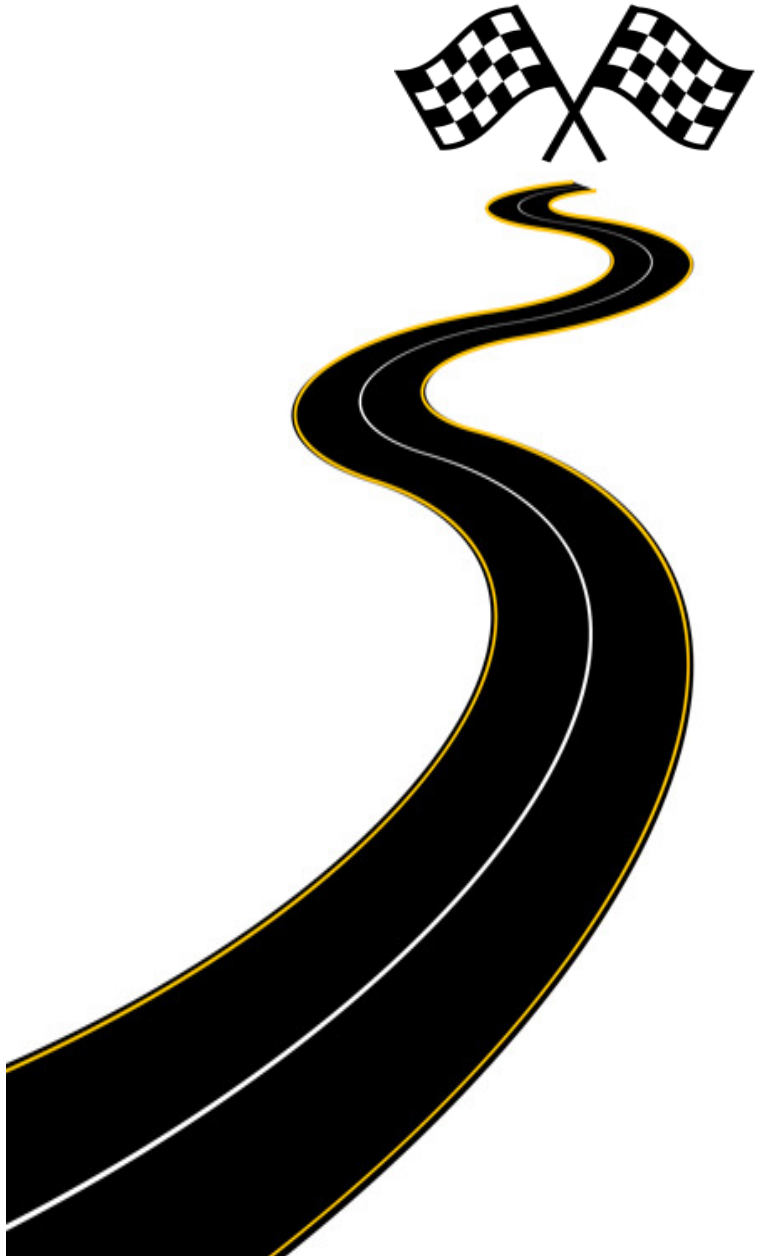
performance matrix $\boldsymbol{P}$

Select

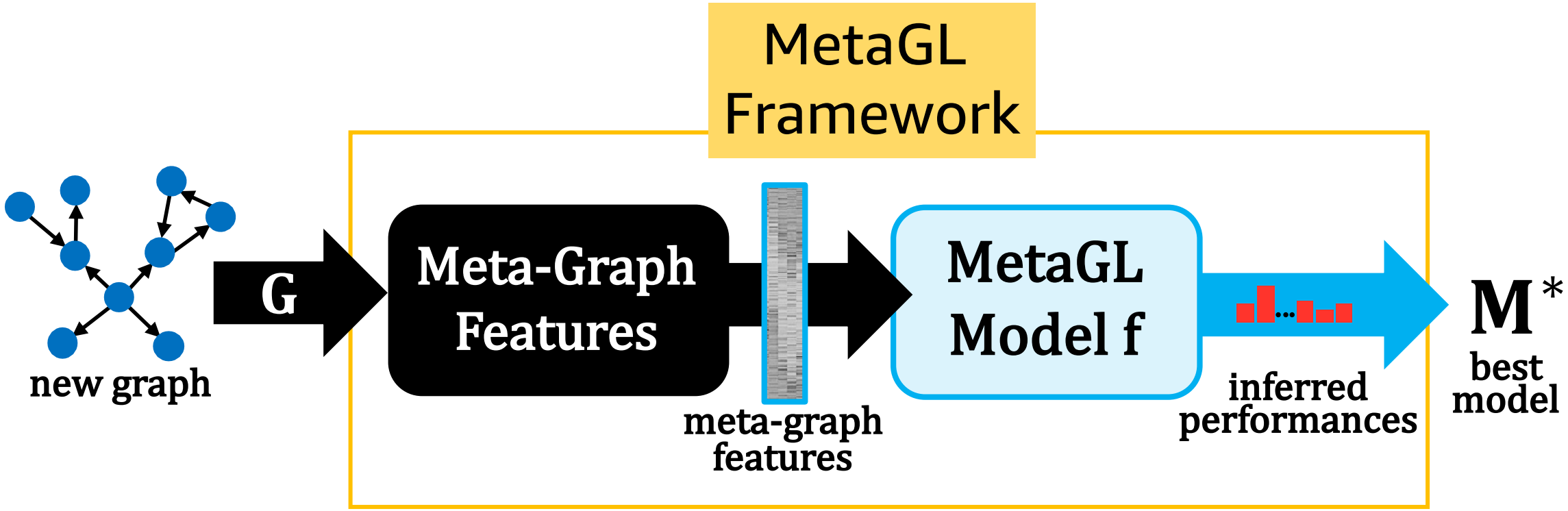
- the best model M^* ***without*** training/evaluating any model on the new graph G

Roadmap

- Introduction & Problem Formulation
- **Proposed Framework: MetaGL**
- Results
- Conclusion

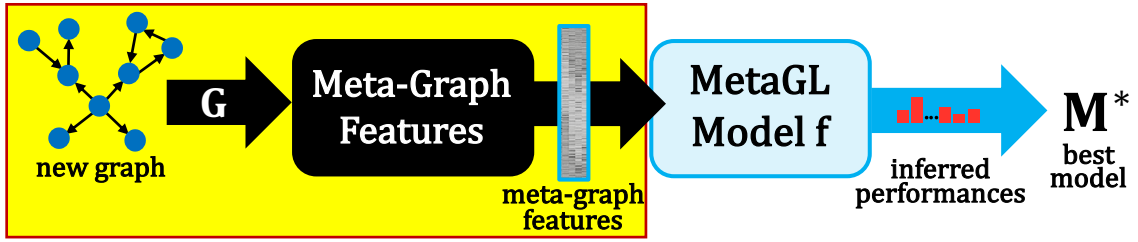


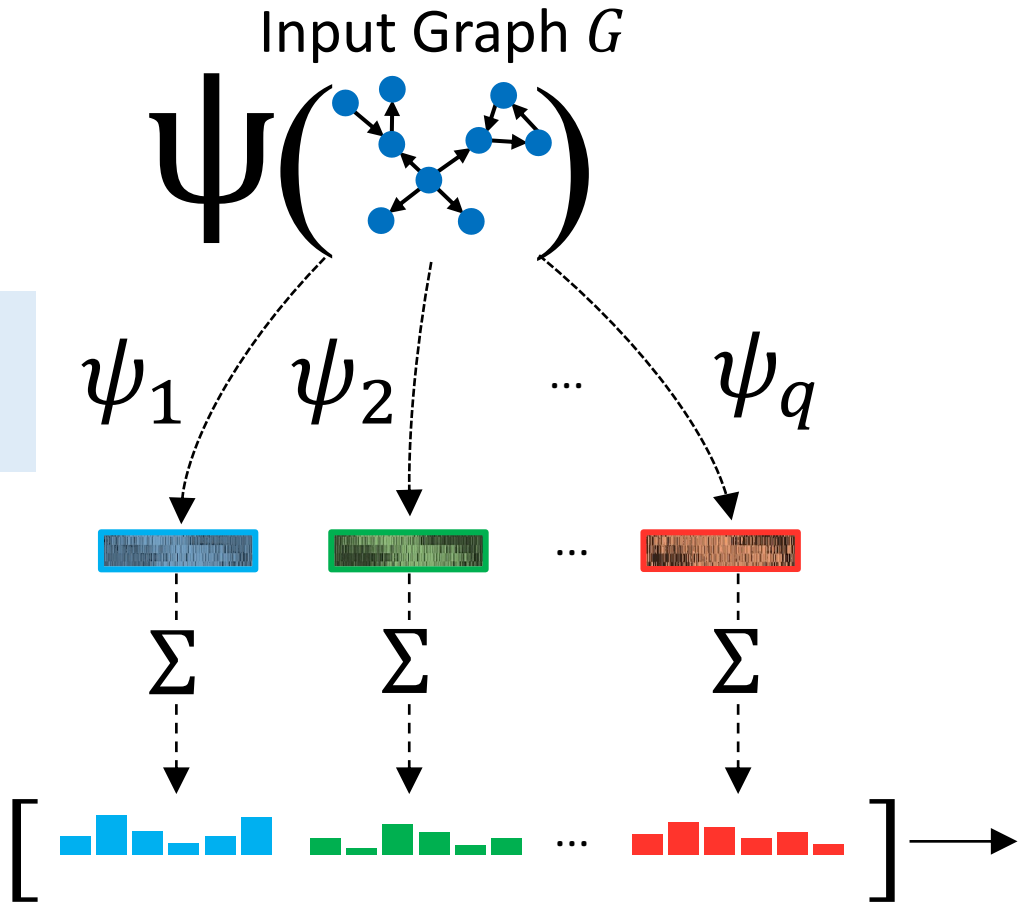
Proposed Framework: MetaGL



Details

MetaGL:
Meta-Graph Features





Structural meta-feature extractors $\{\psi_1, \ldots \psi_q\}$

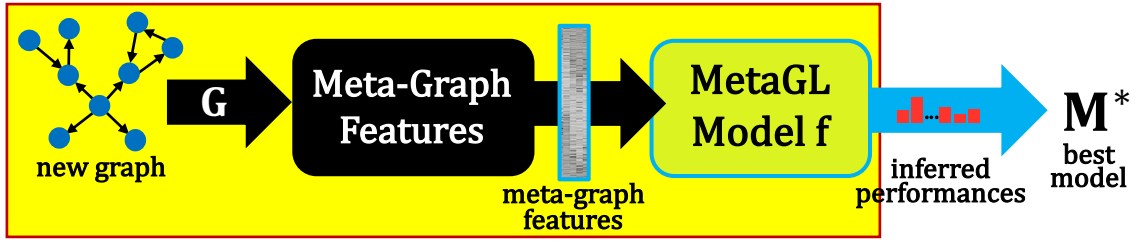
Global statistical functions Σ

Meta-Graph Features $\boldsymbol{m}$

For any graph G, $\boldsymbol{m}$ has a fixed size

Details

MetaGL:
Offline Meta-Training



Estimating performance p_{ij} of model M_j on graph G_i

$$p_{ij} \approx \hat{p}_{ij} = \langle f(\boldsymbol{W}[\underbrace{\boldsymbol{m}_i}_{\text{meta-graph feature}}; \underbrace{\boldsymbol{\phi}(\boldsymbol{m}_i)}_{\text{graph factor}}]), f(\underbrace{\boldsymbol{V}_j}_{\text{model factor}}) \rangle$$

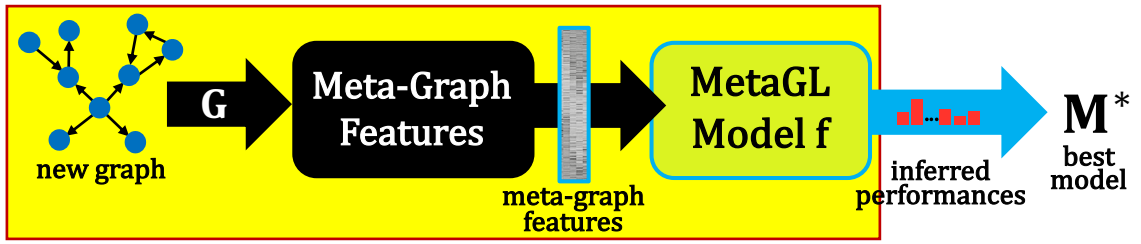
GNN-based embedding function $f(\cdot)$

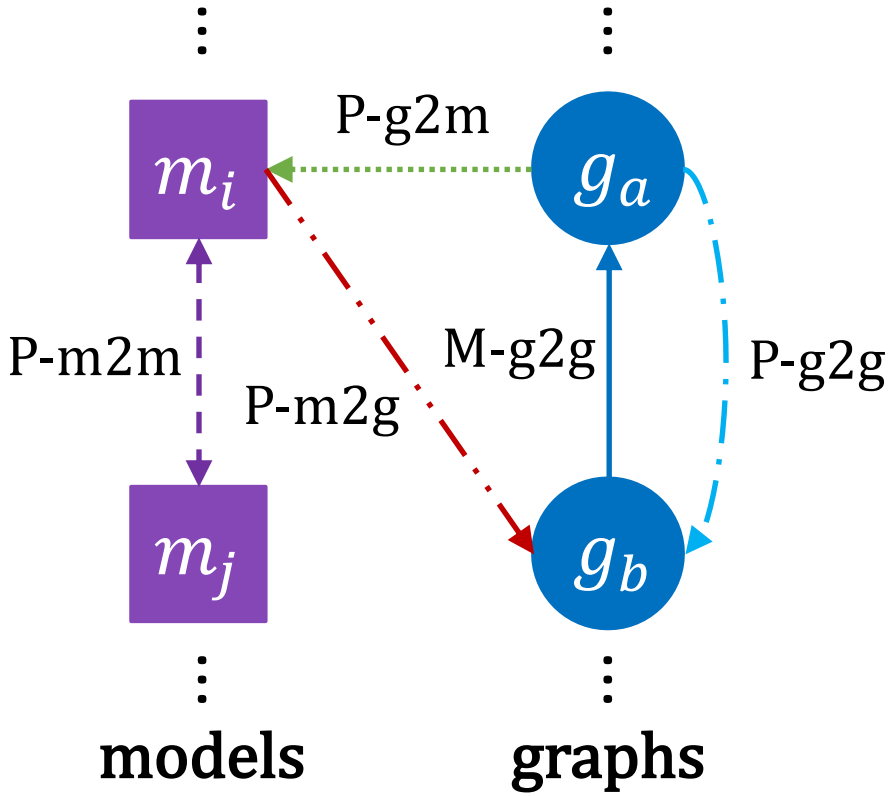
Meta-Learning Objective

- Optimize to find the best model via top-1 probability-based objective

Details

MetaGL:
Offline Meta-Training

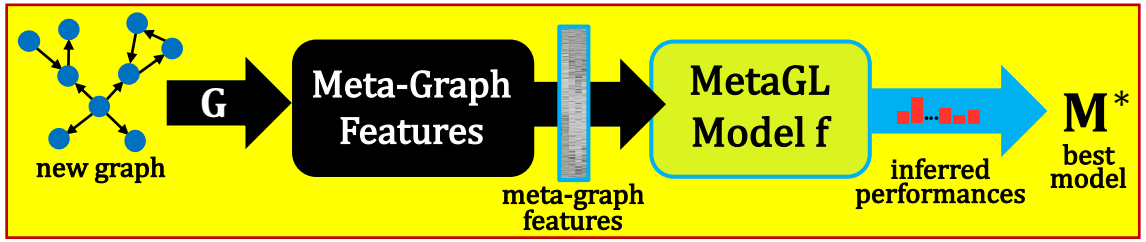




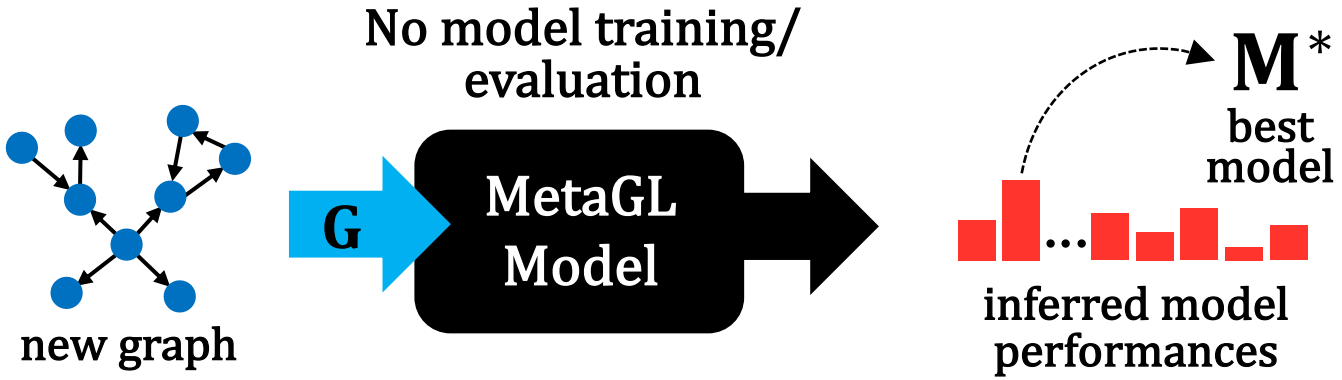
Meta-learner in MetaGL operates on a heterogeneous graph consisting of models and graphs

Details

MetaGL:
Online Model Prediction



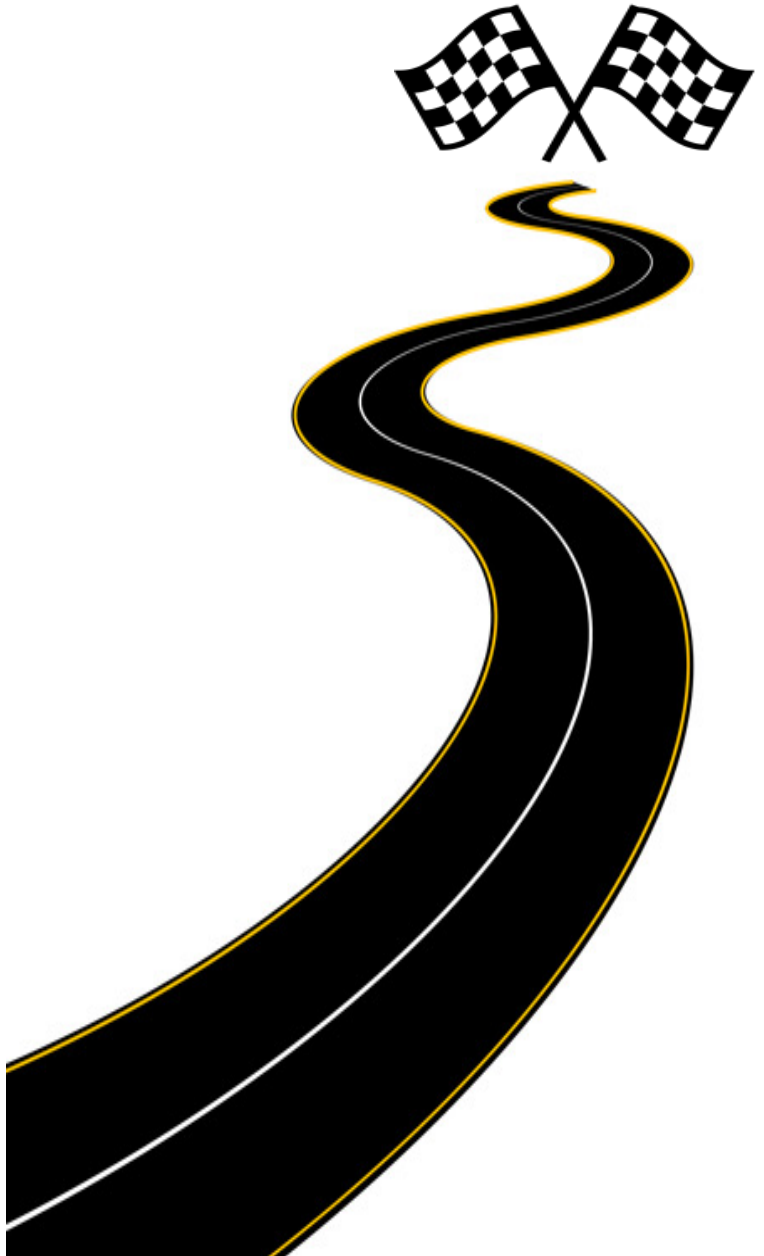
Best model $M^* = \arg\max_{M_j \in \mathcal{M}} \langle f(\boldsymbol{W}[\boldsymbol{m}_{\text{test}}; \phi(\boldsymbol{m}_{\text{test}})]), f(\boldsymbol{V}_j) \rangle$



MetaGL infers the best model M^* with no model training/evaluation

Roadmap

- Introduction & Problem Formulation
- Proposed Framework: MetaGL
- **Experiments**
- Conclusion



Experiments

Research Questions 

RQ1. How accurately can MetaGL select the best model?

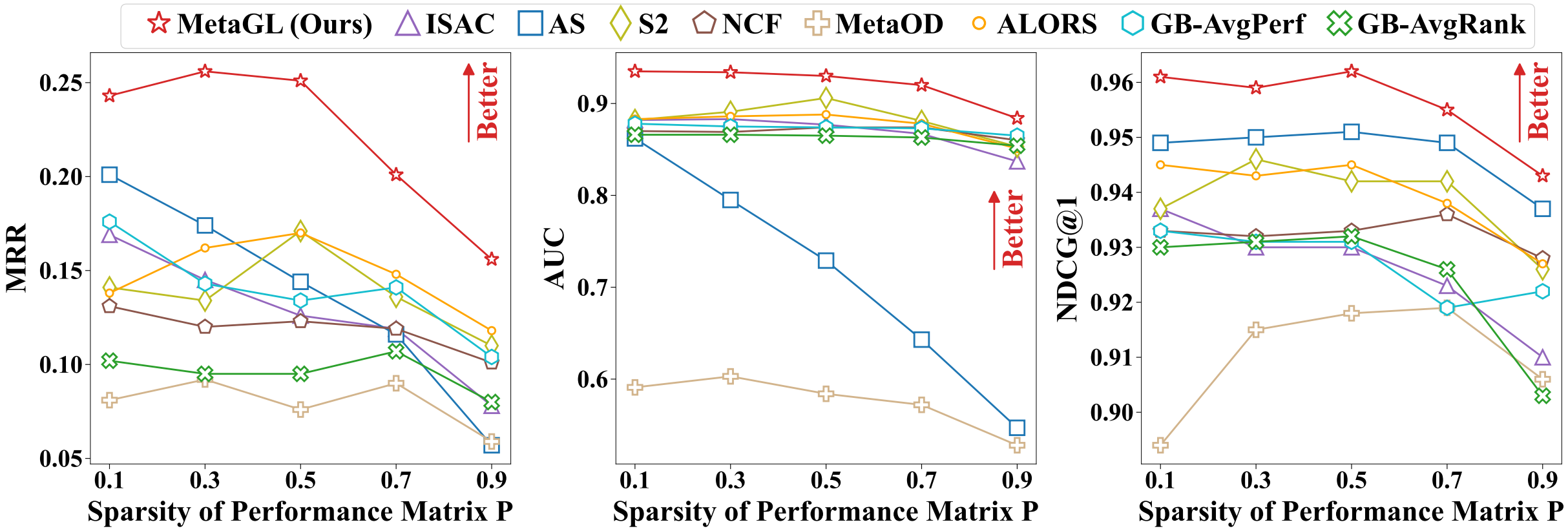
RQ2. How effective are the meta-graph features?

RQ3. How efficient is MetaGL?

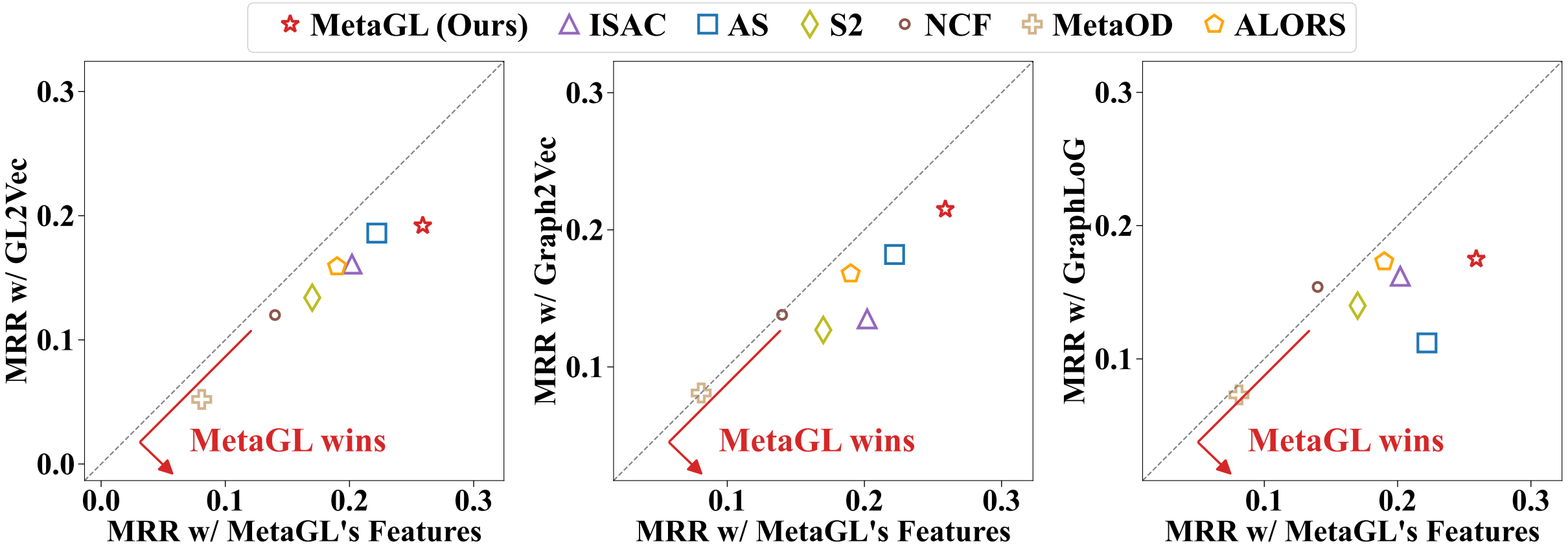
RQ1. Model Selection w/ Fully Observed Perf.

	Method	MRR	AUC	NDCG@1
	Random Selection	0.011	0.490	0.745
Simple	Global Best-AvgPerf	0.163	0.877	0.932
	Global Best-AvgRank	0.103	0.867	0.930
	MetaGL_AS	<u>0.222</u>	0.905	0.947
	MetaGL_ISAC	0.202	0.887	0.939
Optimization -based	MetaGL_S2	0.170	<u>0.910</u>	0.945
	MetaGL_ALORS	0.190	0.897	<u>0.950</u>
	MetaGL_NCF	0.140	0.869	0.934
	MetaGL_MetaOD	0.075	0.599	0.889
	MetaGL (Ours)	**0.259**	**0.941**	**0.962**

RQ1. Model Selection w/ Partially Observed Perf.

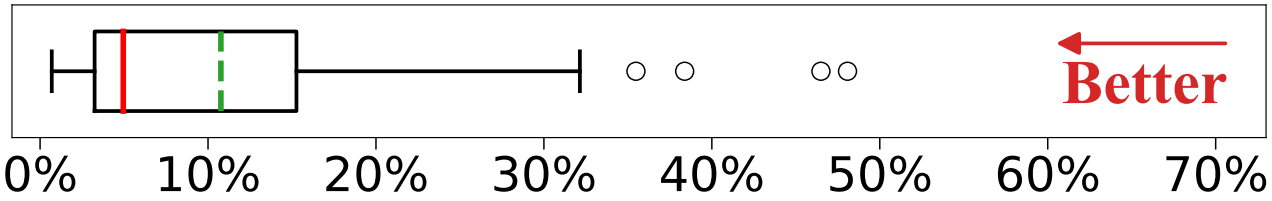


RQ2. Effectiveness of Meta-Graph Features



RQ3. Model Selection Efficiency

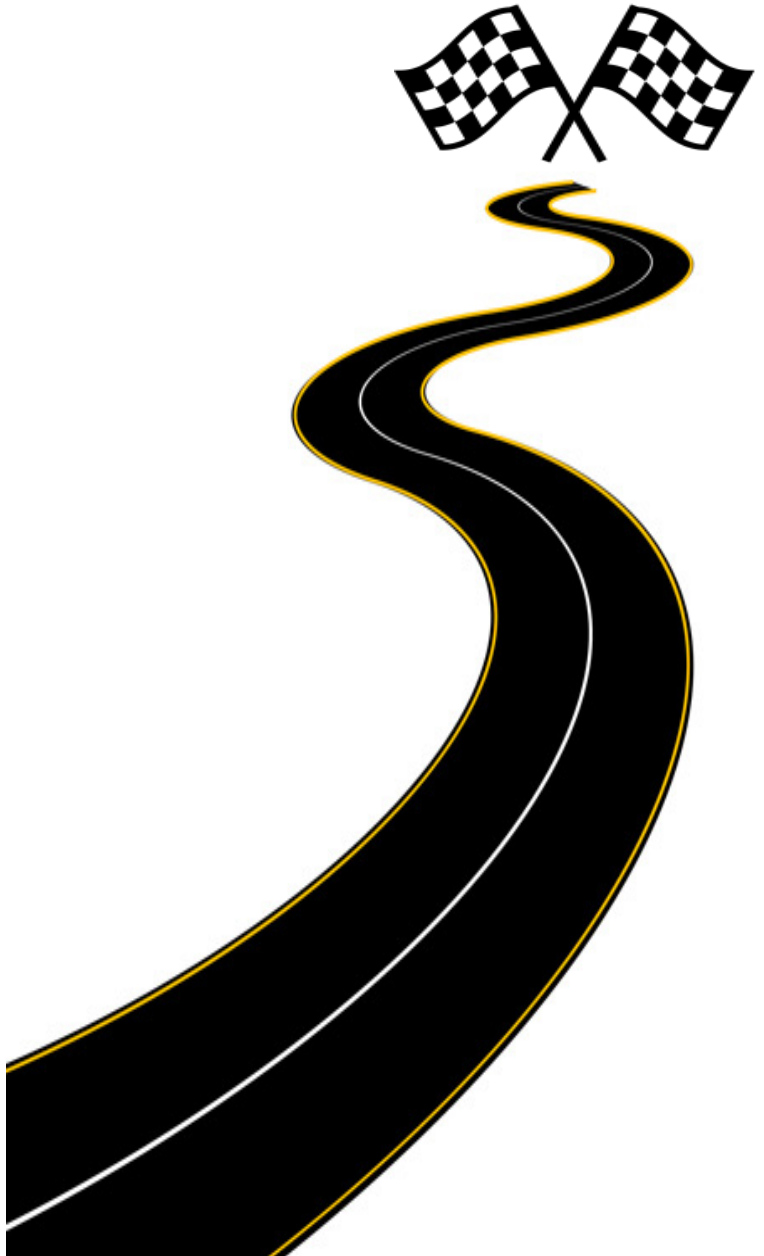
MetaGL's runtime (secs) at test time



MetaGL's runtime / time to train GL models

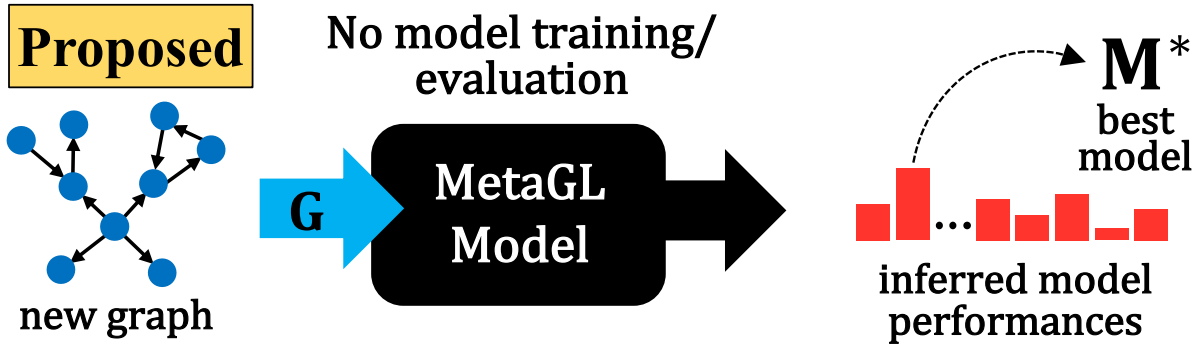
Roadmap

- Introduction & Problem Formulation
- Proposed Framework: MetaGL
- Experiments
- **Conclusion**



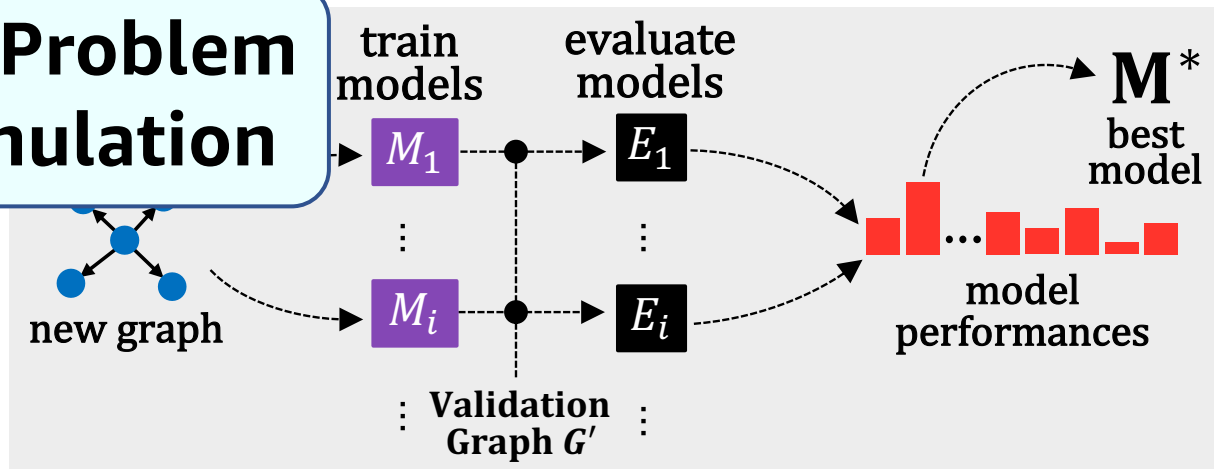
Conclusion

MetaGL: Meta-Learning Framework & Features



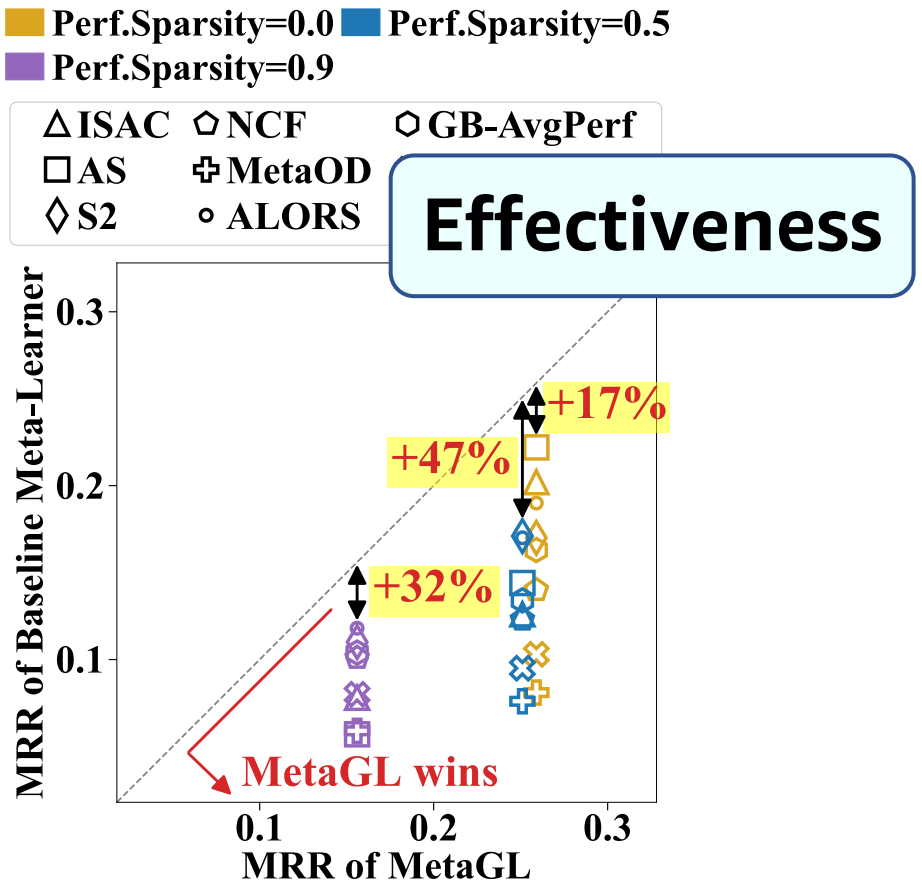
(a) MetaGL infers the best model with no model training/evaluation.

New Problem Formulation



(b) Existing model selection methods train/evaluate multiple models.

Effectiveness



(c) MetaGL consistently performs the best.