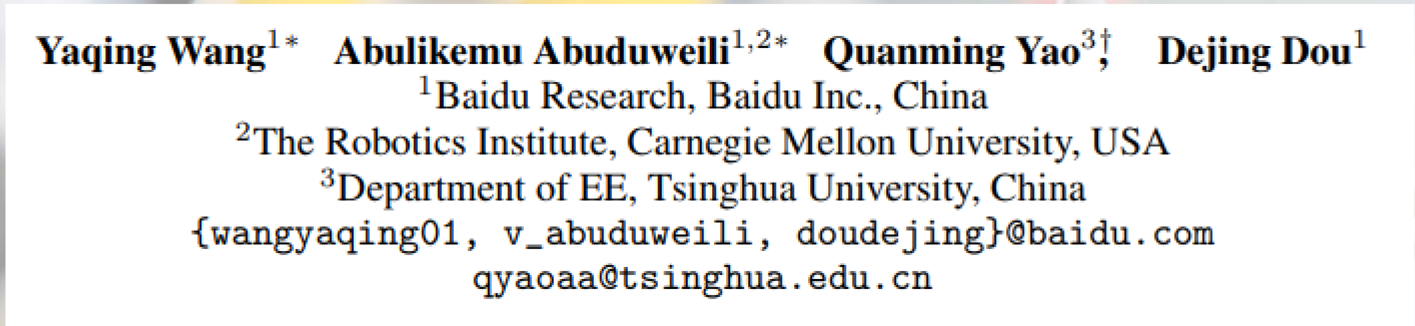
NEURAL INFORMATION
PROCESSING SYSTEMS

Property-aware Adaptive Relation Networks for Few-shot Molecular Property Prediction



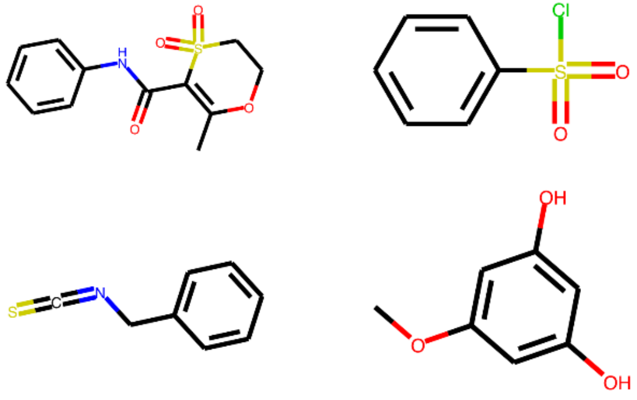
Yaqing Wang[1*] **Abulikemu Abuduweili**[1,2*] **Quanming Yao**[3†] **Dejing Dou**[1]
[1]Baidu Research, Baidu Inc., China
[2]The Robotics Institute, Carnegie Mellon University, USA
[3]Department of EE, Tsinghua University, China
{wangyaqing01, v_abuduweili, doudejing}@baidu.com
qyaoaa@tsinghua.edu.cn

Outline

- Background: Molecule property prediction (MPP)
- Preliminary: Few-shot learning (FSL)
- Related works: FSL for MPP
- The proposed approach PAR
- Summary

Molecular Property Prediction

- Molecules:
  - Mainly micromolecule organics



Examples of molecules

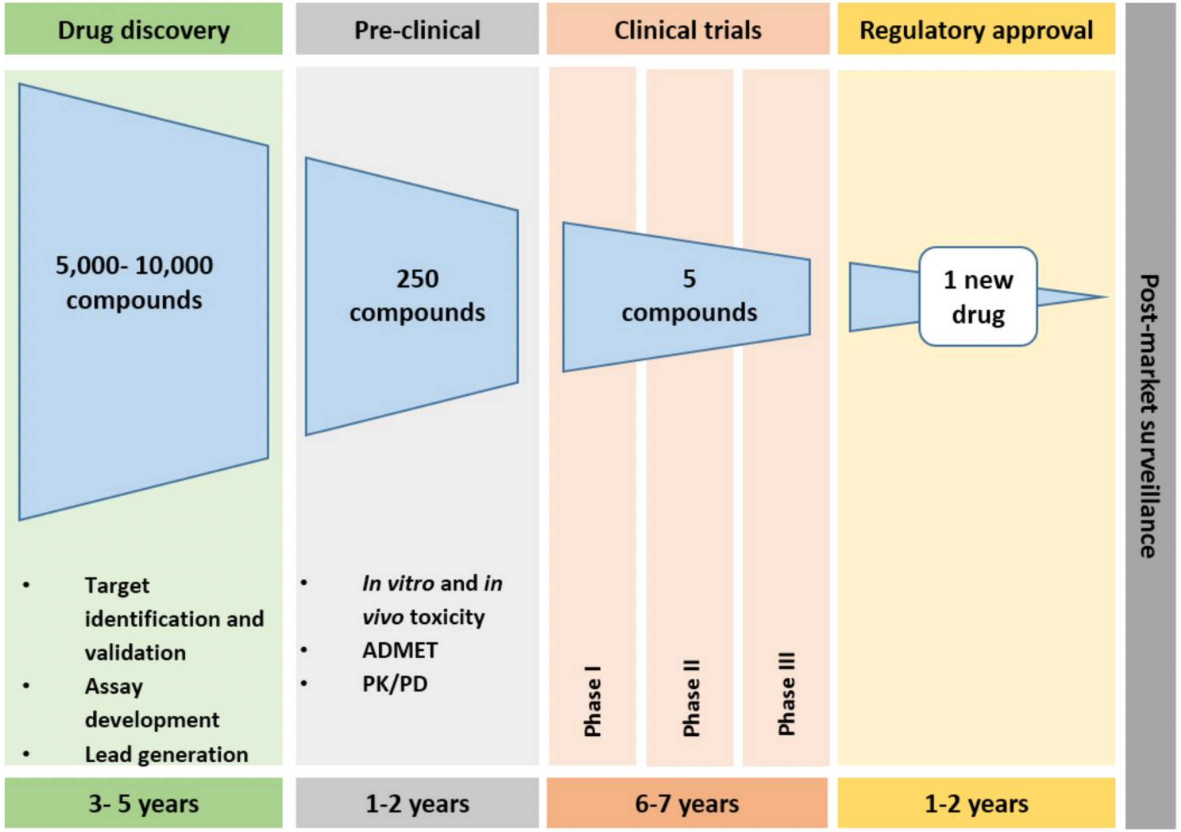
- Properties:
  - Physiology or Toxicity
    - 生理学上的性质、毒性
  - Examples in SIDER :
    - ‘SIDER’ : [‘Hepatobiliary disorders’ , ‘Infections and infestations’ , ‘Neoplasms benign, malignant and unspecified (incl cysts and polyps) ' , …]
      - [肝胆疾病，传染病和虫害侵扰性疾病，良性、恶性和未指定的肿瘤（包括囊肿和息肉）]

Needs for molecular property prediction

Drug discovery targets at finding new potential medical compounds with desired properties

Only a small amount of candidate molecules can pass virtual screening to be evaluated in the lead optimization stage

We only have few molecules with known pharmacological properties



Drug discovery and development timeline from [H. Matthews et al., Proteomes 2016]

Outline

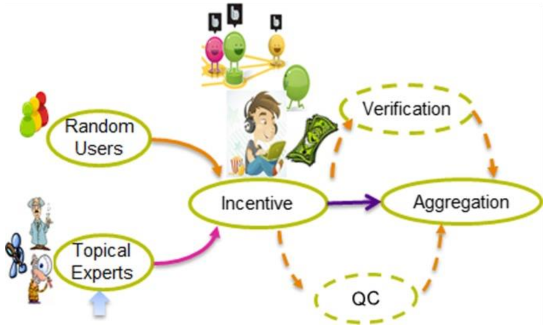
- Background: Molecule property prediction (MPP)
- Preliminary: Few-shot learning (FSL)
    - Definition and Typical Scenario of Few-shot Learning
    - A Common Problem Formulation
    - Exemplar method: MAML
- Related works: FSL for MPP
- The proposed approach PAR
- Summary

Few-shot Learning

- Definition: A type of machine learning problems contains only a limited number of labeled examples

- Typical Scenarios:

Reducing data gathering effort and computational cost

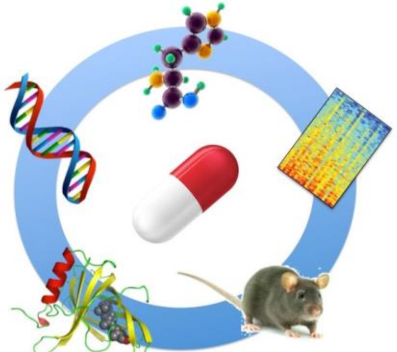


Example: Image / Text Classification

Labor Intensive / High Noise

(few labeled images/texts)

Learning for rare cases



Example: Drug Discovery

Dangerous / Private / Ethical

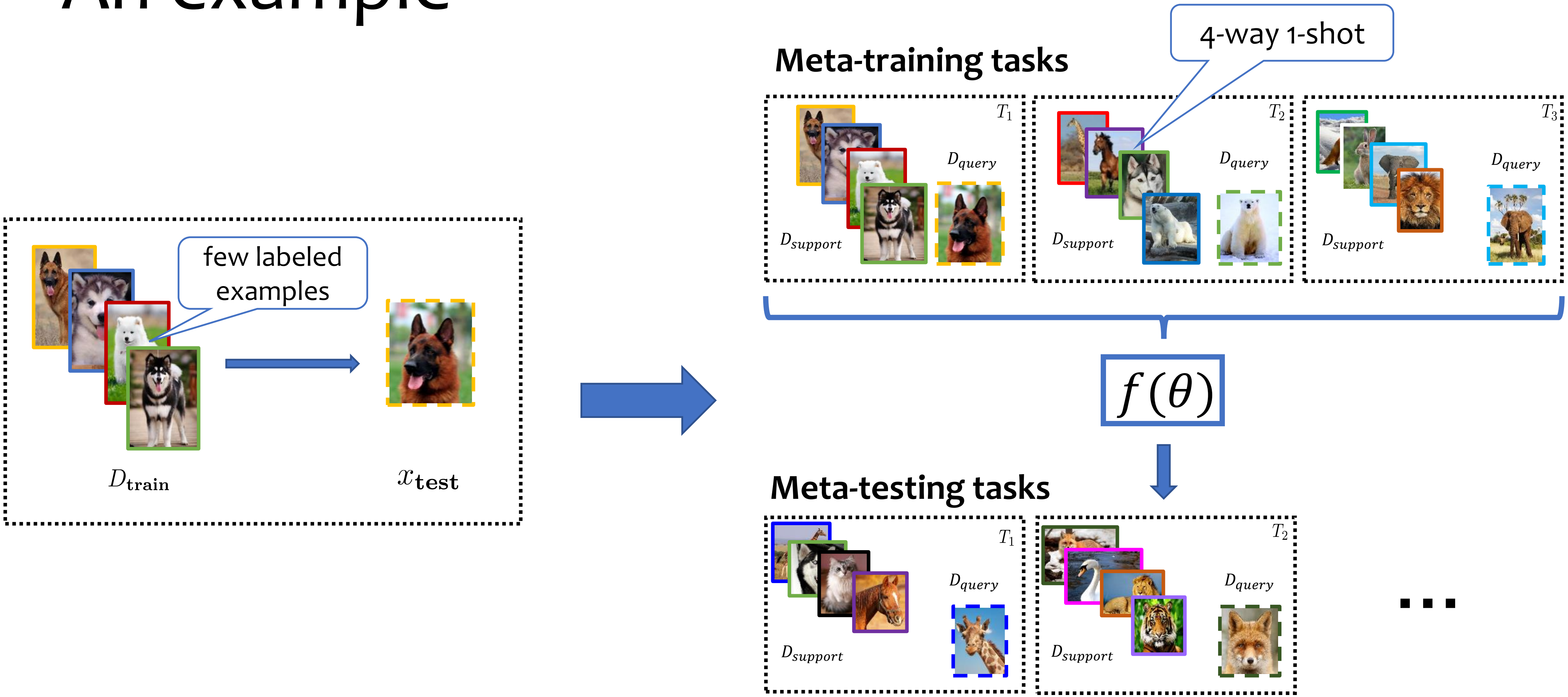
(few labeled drug molecules)

Shot: the number of labeled examples

A Common Problem Formulation of FSL

- Target: learning a predictor from **a set of prediction tasks** and generalize to solve new tasks with a few labeled examples
- Each task T_τ is a N-way K-shot classification task
  - Contains a support set S_τ, there are N*K examples
    - N-way: N different classes
    - K-shot: in each class we have K examples
  - Contains a query set Q_τ
    - Used for test
- Tasks are divided into meta-training tasks and meta-testing tasks

An example



Meta-training tasks

Meta-testing tasks

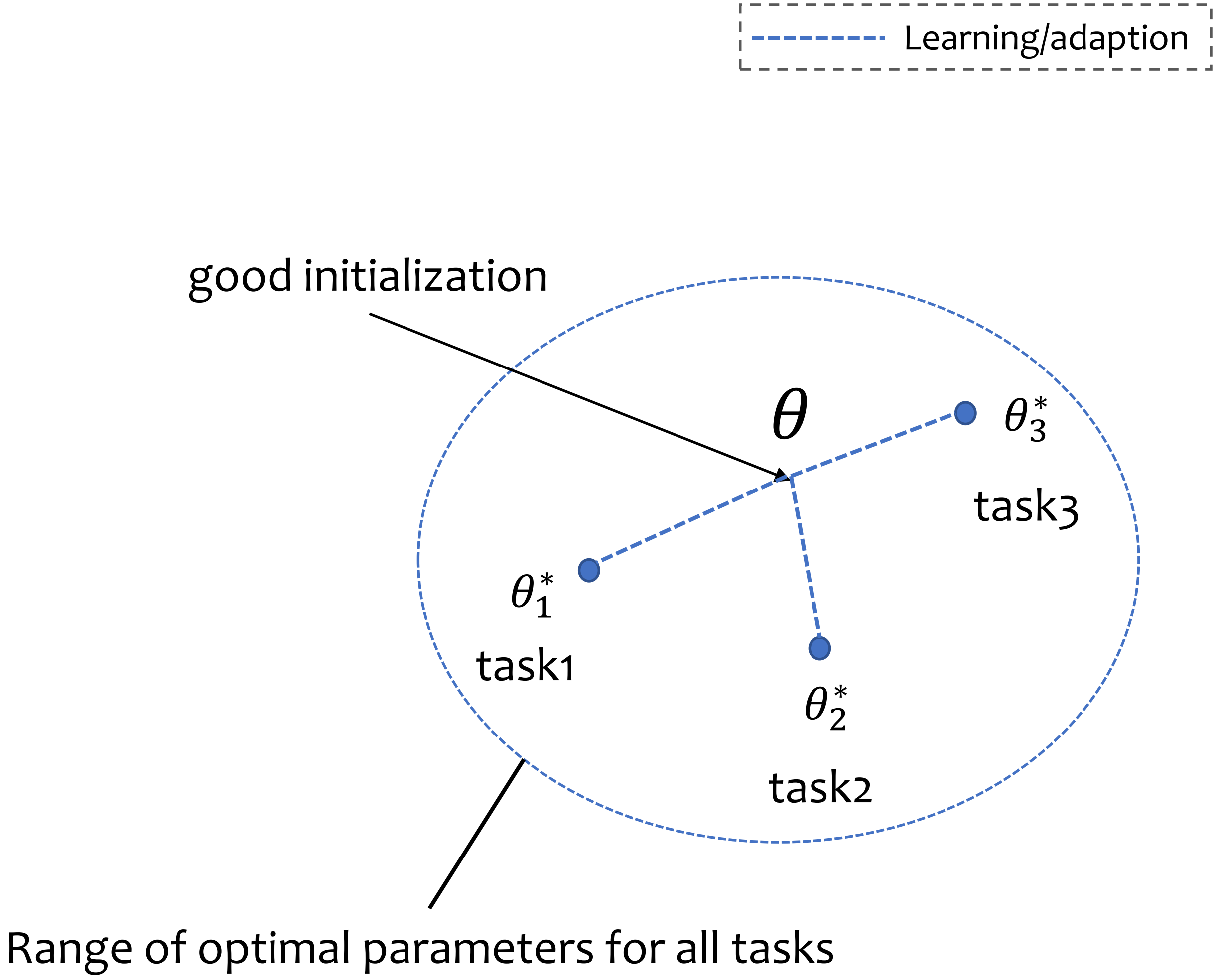
Exemplar method: MAML

- Full name: Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks
- Model: a function $f(\theta)$ with parameters θ
- Two assumptions:
    - Gradient based model
    - The tasks we choose are similar

Finn et al. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks, International Conference on Machine Learning

MAML: adaptation

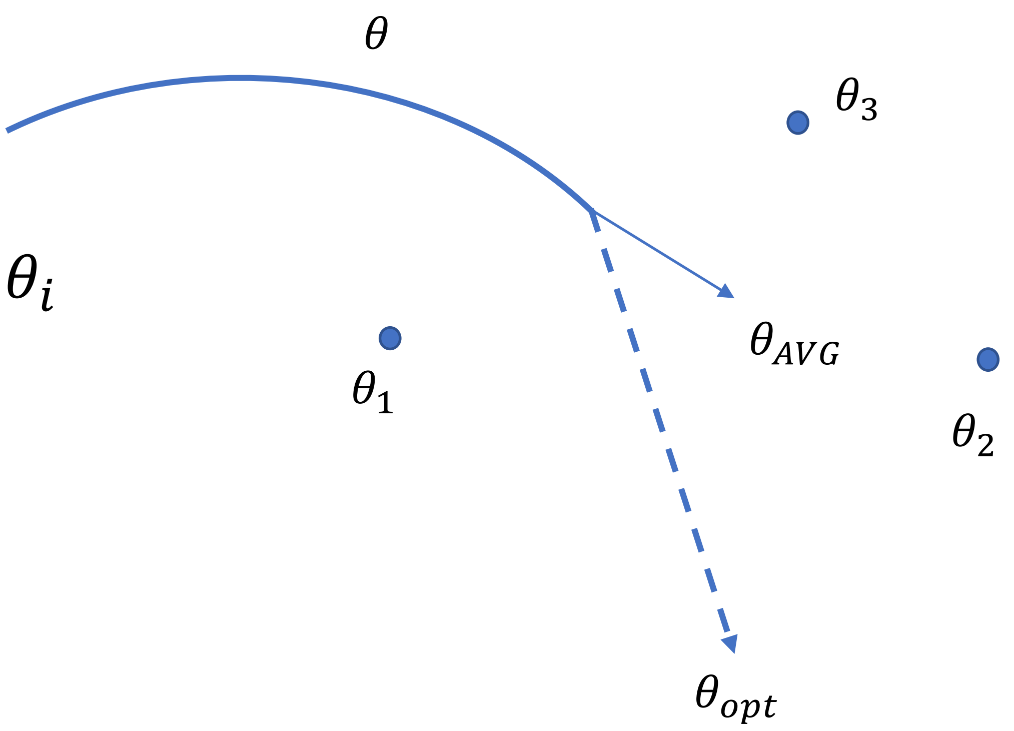
- $\theta_1^*, \theta_2^*, \theta_3^*$: best parameters for 3 tasks
- Target: find a good initialization so that θ can be adapted to tasks (train/test tasks)
- Requirement: the tasks are similar



How to find θ?

Baseline:

- Learn different θ_i from each tasks
  - Each $\mathcal{T}_i \sim P(\mathcal{T})$ is a FSL task $\longrightarrow$ a bad θ_i
- Average:
  - Lack of characteristics for each task
- Adaptation : fine-tune (gradient based)
  - $\theta' = \theta \ - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f(\theta))$



MAML: fast

- For task $\mathcal{T}_i$ model's parameter θ become:

$$\theta_i' = \theta \ - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f(\theta))$$

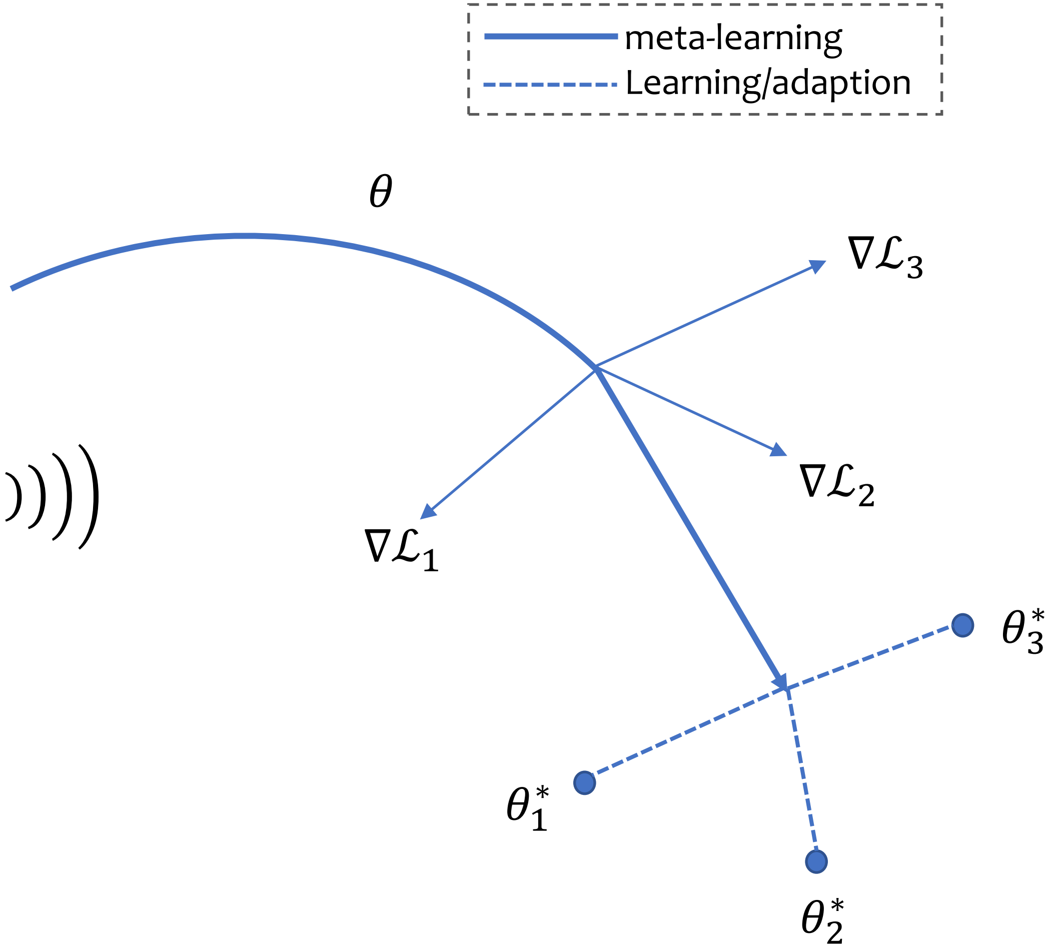
- Learning objective:

$$\min_\theta \sum_{\mathcal{T}_i \sim P(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f(\theta_i')) = \sum_{\mathcal{T}_i \sim P(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}\left(f\left(\theta \ - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}\left(f(\theta)\right)\right)\right)$$

- $\nabla_\theta \left(f(\theta \ - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f(\theta))\right) =$

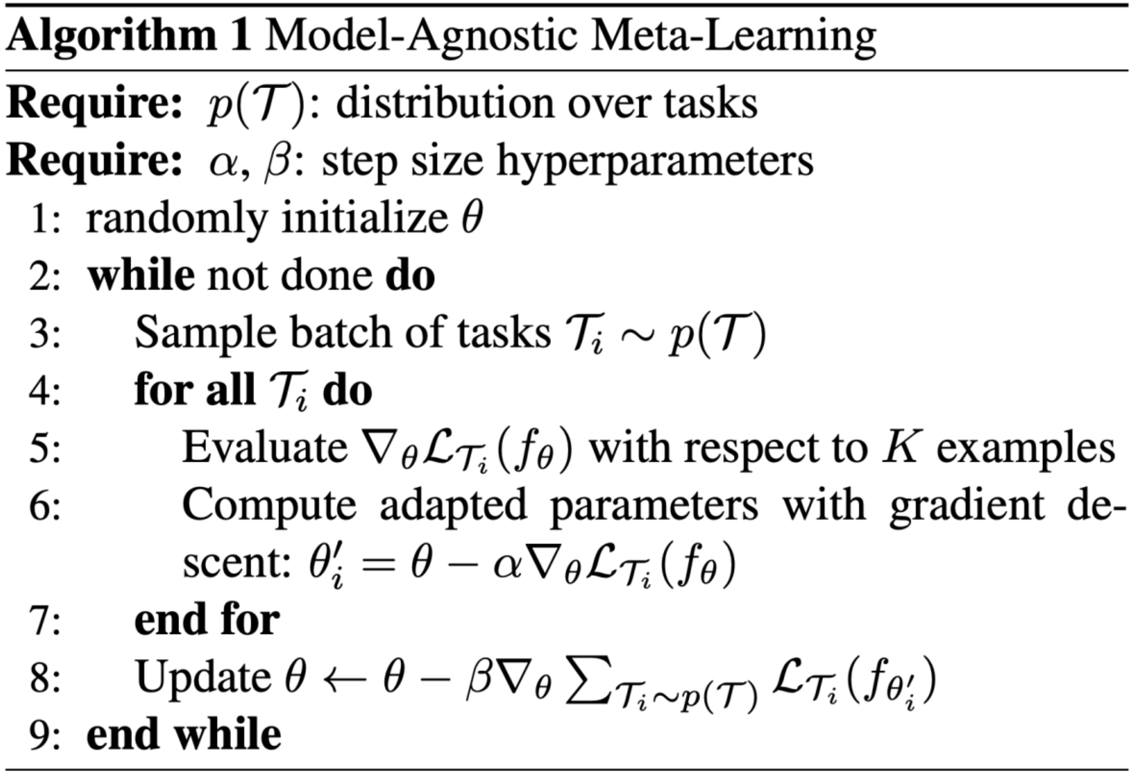
$$f'\left(\theta \ - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}\left(f(\theta)\right)\right)\left(1 - \alpha \nabla_{\theta\theta} \mathcal{L}_{\mathcal{T}_i}\left(f(\theta)\right)\right)$$

- Fast: update parameters only once
- Requirement: gradient based model



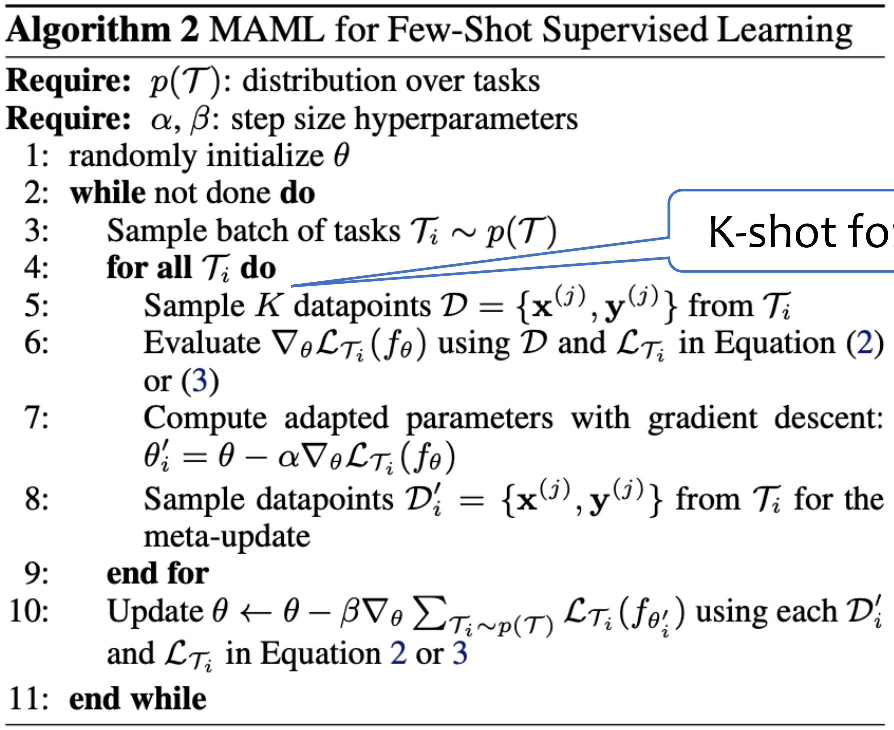
MAML: meta-learning process

- The meta-learner provides the initial value of parameters for each task and optimizes it through the accumulated loss of all tasks.
- Within each task, several steps of gradient descent are generalized to new tasks through training samples.



```
Algorithm 1 Model-Agnostic Meta-Learning
Require: $p(\mathcal{T})$: distribution over tasks
Require: $\alpha, \beta$: step size hyperparameters
randomly initialize $\theta$
while not done do
    Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
    for all $\mathcal{T}_i$ do
        Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ with respect to $K$ examples
        Compute adapted parameters with gradient descent: $\theta'_i = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
    end for
    Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta'_i})$
end while
```

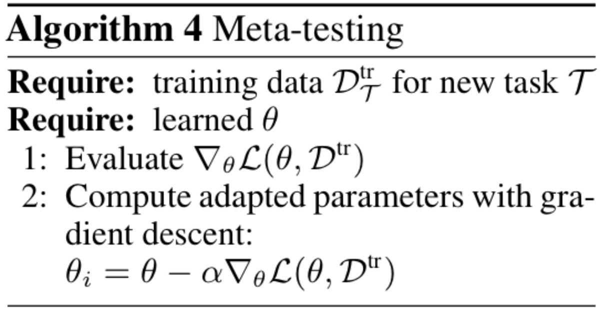
MAML for few-shot learning



Algorithm 2 MAML for Few-Shot Supervised Learning

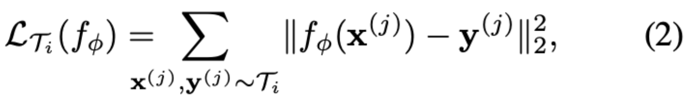
```
Require: $p(\mathcal{T})$: distribution over tasks
Require: $\alpha, \beta$: step size hyperparameters
randomly initialize $\theta$
while not done do
    Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
    for all $\mathcal{T}_i$ do
        Sample $K$ datapoints $\mathcal{D} = \{\mathbf{x}^{(j)}, \mathbf{y}^{(j)}\}$ from $\mathcal{T}_i$
        Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ using $\mathcal{D}$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation (2) or (3)
        Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
        Sample datapoints $\mathcal{D}_i' = \{\mathbf{x}^{(j)}, \mathbf{y}^{(j)}\}$ from $\mathcal{T}_i$ for the meta-update
    end for
    Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$ using each $\mathcal{D}_i'$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation 2 or 3
end while
```

K-shot for N classes



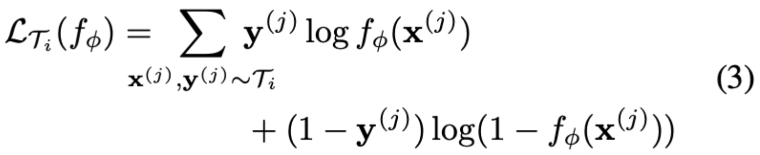
Algorithm 4 Meta-testing

```
Require: training data $\mathcal{D}_\mathcal{T}^{\text{tr}}$ for new task $\mathcal{T}$
Require: learned $\theta$
Evaluate $\nabla_\theta \mathcal{L}(\theta, \mathcal{D}^{\text{tr}})$
Compute adapted parameters with gradient descent: $\theta_i = \theta - \alpha \nabla_\theta \mathcal{L}(\theta, \mathcal{D}^{\text{tr}})$
```



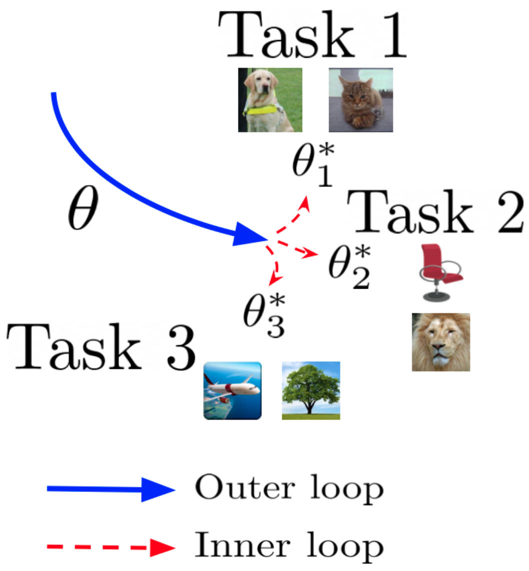
$$\mathcal{L}_{\mathcal{T}_i}(f_\phi) = \sum_{\mathbf{x}^{(j)}, \mathbf{y}^{(j)} \sim \mathcal{T}_i} \| f_\phi(\mathbf{x}^{(j)}) - \mathbf{y}^{(j)} \|_2^2, \qquad (2)$$

(regression)



$$\mathcal{L}_{\mathcal{T}_i}(f_\phi) = \sum_{\mathbf{x}^{(j)}, \mathbf{y}^{(j)} \sim \mathcal{T}_i} \mathbf{y}^{(j)} \log f_\phi(\mathbf{x}^{(j)}) + (1 - \mathbf{y}^{(j)}) \log(1 - f_\phi(\mathbf{x}^{(j)})) \qquad (3)$$

(classification)



Limitations

- Provides the same initialization for all tasks
  - Neglects task-specific information
  - Appropriate only when the set of tasks are all very similar
- Lack of mathematical strictness
- Refinement by a few gradient descent steps may not be reliable

Take-home message

- Basic concept in FSL:
  - Few labeled examples → difficult to train a good model
  - Problem Formulation:
    - Learn a predictor from a set of tasks → n-way k-shot

- MAML: how to adapt to tasks?
  - Find a good initialization of θ so that θ can be adapted to tasks

- MAML: learning objective
  - Minimize the loss for all the tasks when we adapt θ to that particular task

Outline

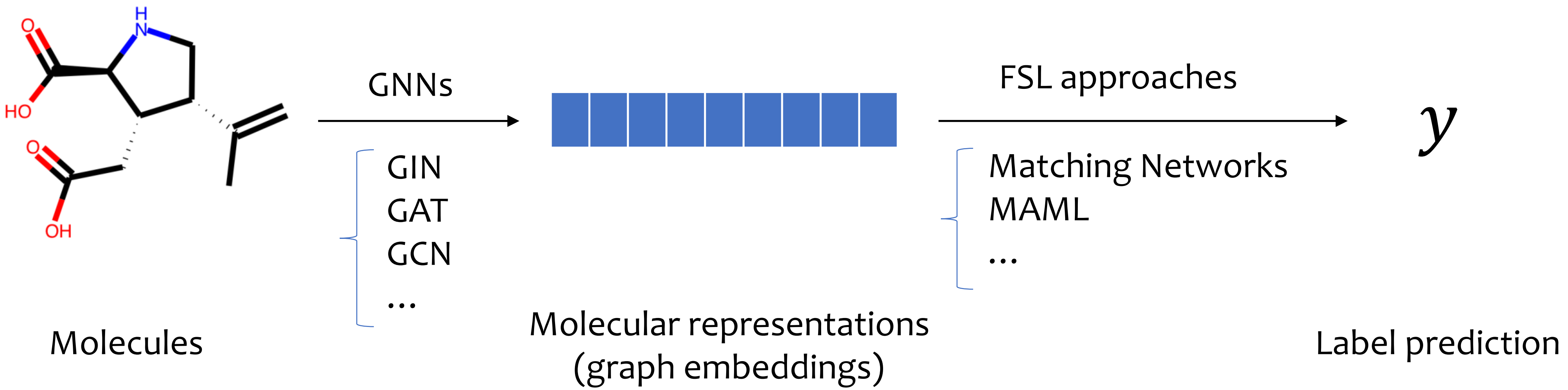
- Background: Molecule property prediction (MPP)
- Preliminary: Few-shot learning (FSL)
- Related works: FSL for MPP
  - MPP: a Few-shot graph learning problems
  - Existing Work: IterRefLSTM
  - Existing Work: Meta-MGNN
- The proposed approach PAR
- Summary

MPP: a Few-shot graph learning problems

- Few-shot:
  - *Application scenario*: even a hundred compounds is often too resource intensive for standard drug discovery campaigns
  - *Our datasets*: more than half of the properties only are shared by fewer than 100 molecules across several datasets

- Graph-learning:
  - Graph based molecular representation learning methods are popularly used and obtain state-of-the-art performance.
  - The paper ‘ MoleculeNet : a benchmark for molecular machine learning ’ shows the strength of graph models.

MPP: a common framework



Molecules

GNNs

- GIN
- GAT
- GCN
- …

Molecular representations (graph embeddings)

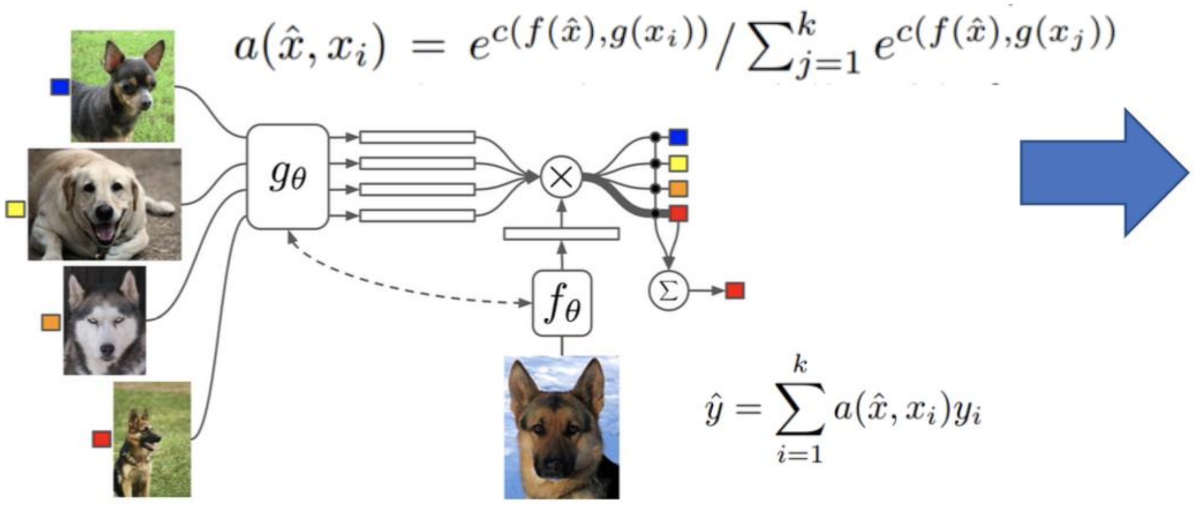
FSL approaches

- Matching Networks
- MAML
- …

Label prediction

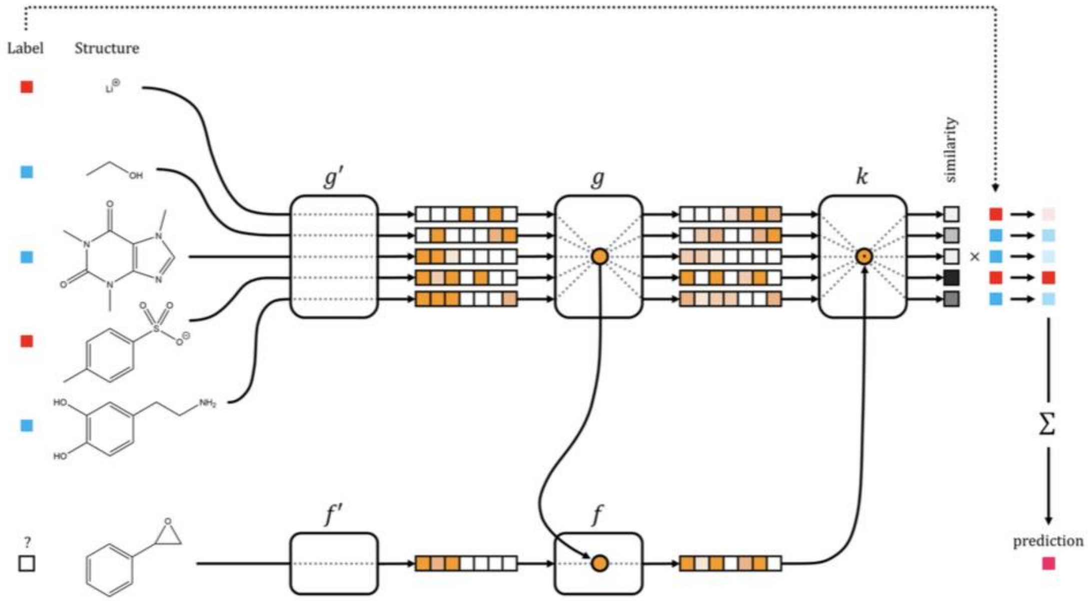
Existing Work: IterRefLSTM

- Motivation
  - Adapt Matching Networks(one-shot learning) to handle molecular property prediction tasks with few training data
  - Propose IterRefLSTM to modify Matching Networks architecture



$$a(\hat{x}, x_i) = e^{c(f(\hat{x}), g(x_i))} / \sum_{j=1}^{k} e^{c(f(\hat{x}), g(x_j))}$$

$$\hat{y} = \sum_{i=1}^{k} a(\hat{x}, x_i) y_i$$



Oriol et al. 2016. Matching Networks for One Shot Learning, NeurIPS

Han et al. 2017. Low Data Drug Discovery with One-Shot Learning, ACS Central Science

Architecture modification in IterRefLSTM

- Matching Networks
  - $f(\hat{x}, S) = attLSTM(f'(\hat{x}), g(S), K)$
  - $g(x|S) = BiLSTM(g'(x_1)| \ldots |g'(x_m))$
- Drawbacks:
  - the order dependence in the support-embedding g
  - The definition of $f(\hat{x}, S)$ relies on g

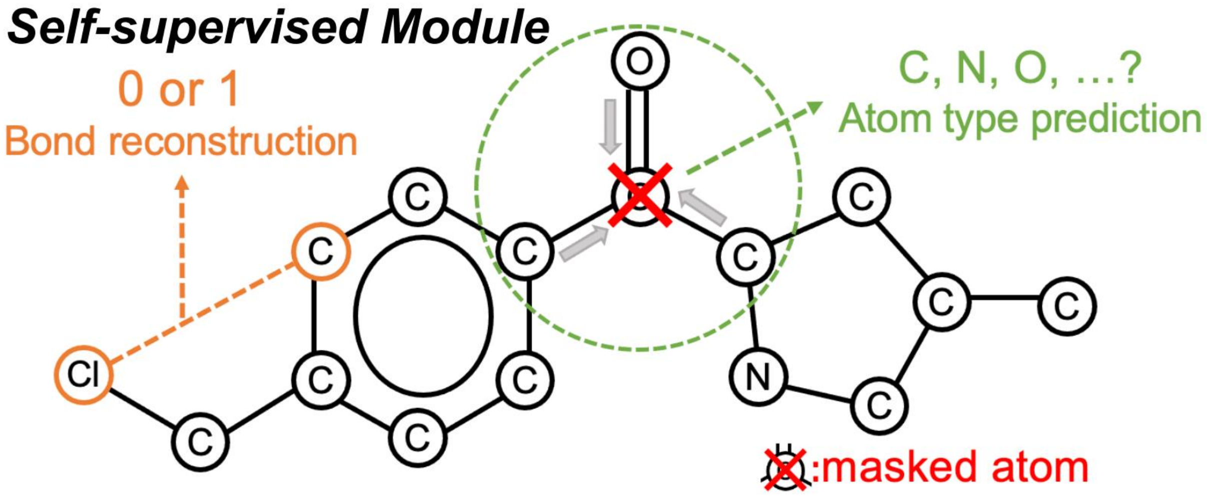
- IterRefLSTM
  - use an attLSTM to generate both query embedding f and support embedding g
  - iteratively evolves both embeddings(f and g) simultaneously
  - the output g is not related to the order of examples in the support set

Limitations

- Implement premature graph convolution approaches
- Neglect the relations between molecules in support set
  - Common points among positive examples
- Do not consider the differences between different properties
  - Iterative Refinement LSTMs do not take ‘y’ into consider

Existing Work: Meta-MGNN

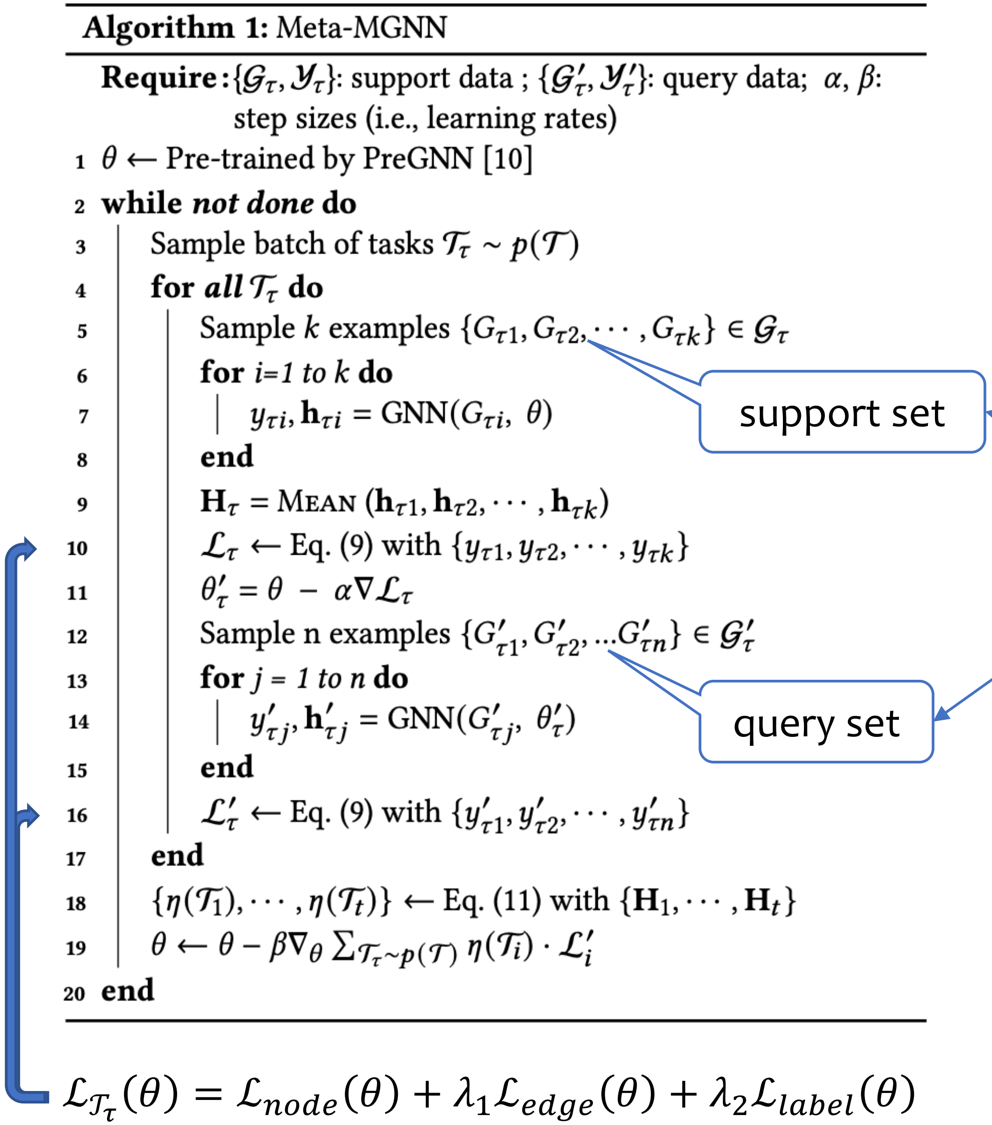
- Motivation:
  - Combine MAML and preGNN for Molecular Property Prediction
- preGNN:
  - exploit the useful unlabeled information in graphs



Hu et al. 2020. Strategies for Pre-training Graph Neural Networks, ICLR

Guo et al. 2021. Few-Shot Graph Learning for Molecular Property Prediction , WWW

Existing Work: Meta-MGNN



```
Algorithm 1: Meta-MGNN
Require: $\{\mathcal{G}_\tau, \mathcal{Y}_\tau\}$: support data ; $\{\mathcal{G}'_\tau, \mathcal{Y}'_\tau\}$: query data;  $\alpha$, $\beta$:
         step sizes (i.e., learning rates)
$\theta \leftarrow$ Pre-trained by PreGNN [10]
while not done do
    Sample batch of tasks $\mathcal{T}_\tau \sim p(\mathcal{T})$
    for all $\mathcal{T}_\tau$ do
        Sample $k$ examples $\{G_{\tau 1}, G_{\tau 2}, \cdots, G_{\tau k}\} \in \mathcal{G}_\tau$
        for $i$=1 to $k$ do
            $y_{\tau i}, \mathbf{h}_{\tau i} = \mathrm{GNN}(G_{\tau i},\ \theta)$
        end
        $\mathbf{H}_\tau = \mathrm{MEAN}\,(\mathbf{h}_{\tau 1}, \mathbf{h}_{\tau 2}, \cdots, \mathbf{h}_{\tau k})$
        $\mathcal{L}_\tau \leftarrow$ Eq. (9) with $\{y_{\tau 1}, y_{\tau 2}, \cdots, y_{\tau k}\}$
        $\theta'_\tau = \theta\ -\ \alpha \nabla \mathcal{L}_\tau$
        Sample n examples $\{G'_{\tau 1}, G'_{\tau 2}, ...G'_{\tau n}\} \in \mathcal{G}'_\tau$
        for $j$ = 1 to $n$ do
            $y'_{\tau j}, \mathbf{h}'_{\tau j} = \mathrm{GNN}(G'_{\tau j},\ \theta'_\tau)$
        end
        $\mathcal{L}'_\tau \leftarrow$ Eq. (9) with $\{y'_{\tau 1}, y'_{\tau 2}, \cdots, y'_{\tau n}\}$
    end
    $\{\eta(\mathcal{T}_1), \cdots, \eta(\mathcal{T}_t)\} \leftarrow$ Eq. (11) with $\{\mathbf{H}_1, \cdots, \mathbf{H}_t\}$
    $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_\tau \sim p(\mathcal{T})} \eta(\mathcal{T}_i) \cdot \mathcal{L}'_i$
end
```

$$\mathcal{L}_{\mathcal{T}_\tau}(\theta) = \mathcal{L}_{node}(\theta) + \lambda_1 \mathcal{L}_{edge}(\theta) + \lambda_2 \mathcal{L}_{label}(\theta)$$

- Molecule representation:
  - GIN
  - preGNN: initial θ + loss function
- FSL methods:
  - Based on MAML
- Other designs:
  - Loss functions
    - Prediction loss
    - Bond reconstruction loss
    - Atom type prediction loss
  - Task-aware attention

Limitations

- Neglect the relations between molecules
  - Support set —— Query set
  - Relations molecules with the same property
- Networks designed for bond reconstruction or atom type prediction do not directly influence the property prediction
  - A waste of resource?

Take home message

- MPP: a Few-shot graph learning problems

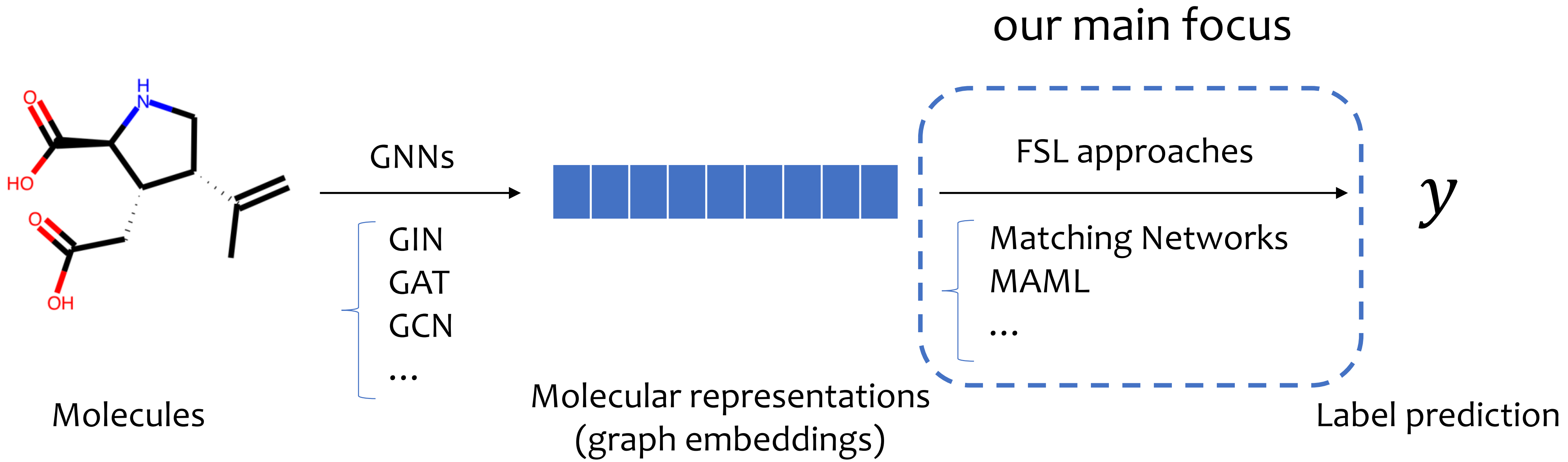
- Common framework
  - Molecules —(GNNs) →Molecular representations —(FSL approaches)→Label prediction

- Two related works
  - Follow the common framework
  - Both neglect the relations between molecules

Outline

- Background: Molecule property prediction (MPP)
- Preliminary: Few-shot learning (FSL)
- Related works: FSL for MPP
- The proposed approach PAR
  - Introduction of PAR
  - Experiments
- Summary

Review: a common framework of MPP



our main focus

Molecules

GNNs

GIN
GAT
GCN
…

Molecular representations
(graph embeddings)

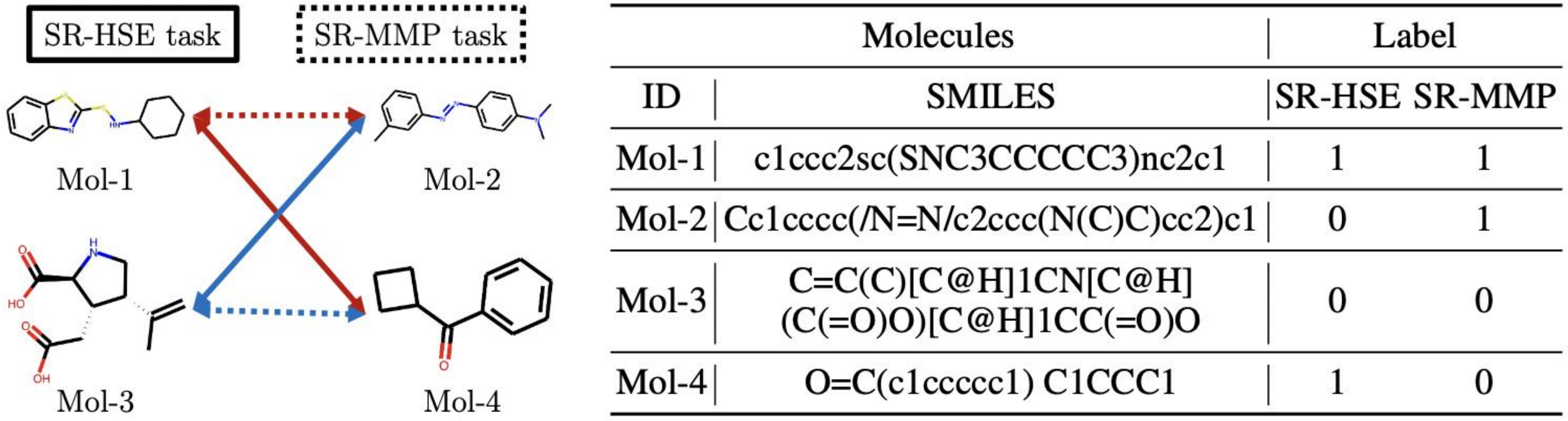
FSL approaches

Matching Networks
MAML
…

y

Label prediction

Motivation



Molecules		Label	
ID	SMILES	SR-HSE	SR-MMP
Mol-1	c1ccc2sc(SNC3CCCCC3)nc2c1	1	1
Mol-2	Cc1cccc(/N=N/c2ccc(N(C)C)cc2)c1	0	1
Mol-3	C=C(C)[C@H]1CN[C@H] (C(=O)O)[C@H]1CC(=O)O	0	0
Mol-4	O=C(c1ccccc1) C1CCC1	1	0

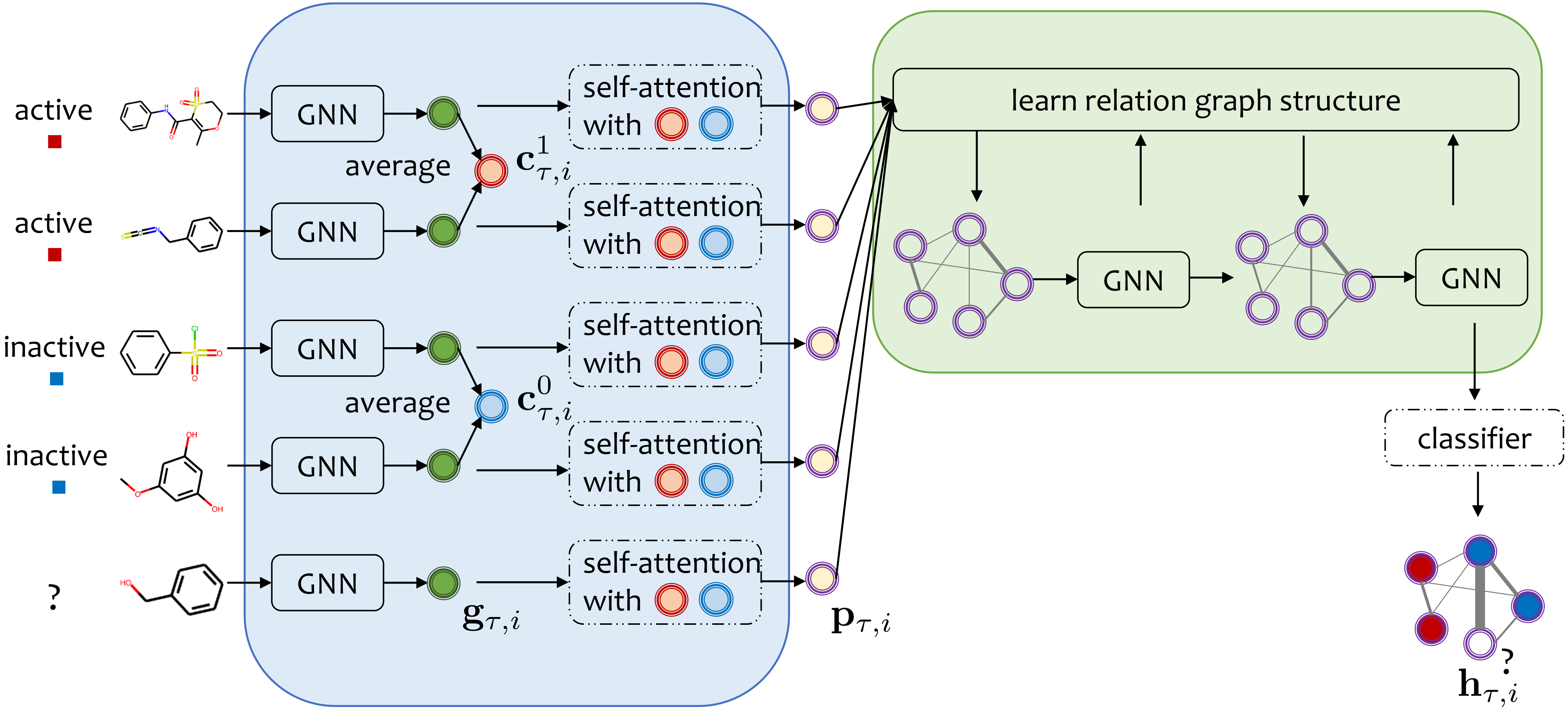
Figure 1: Examples of relation graphs for the same molecules coexisting in two tasks of Tox21. Red (blue) edges mean the connected molecules are both active (inactive) on the target property.

Existing works neglect two key facts

- Different molecular properties are attributed to different molecule substructures
- Relationship among molecules also vary w.r.t. the target property

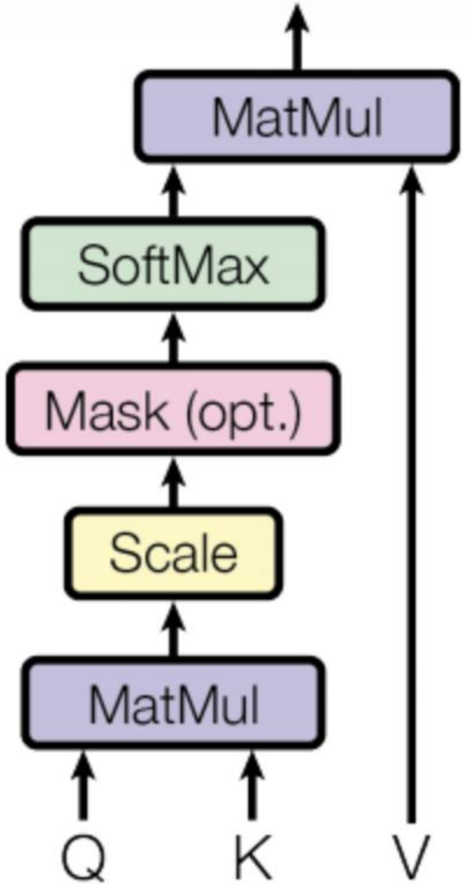
PAR Framework

We propose Property-Aware Relation networks (PAR)



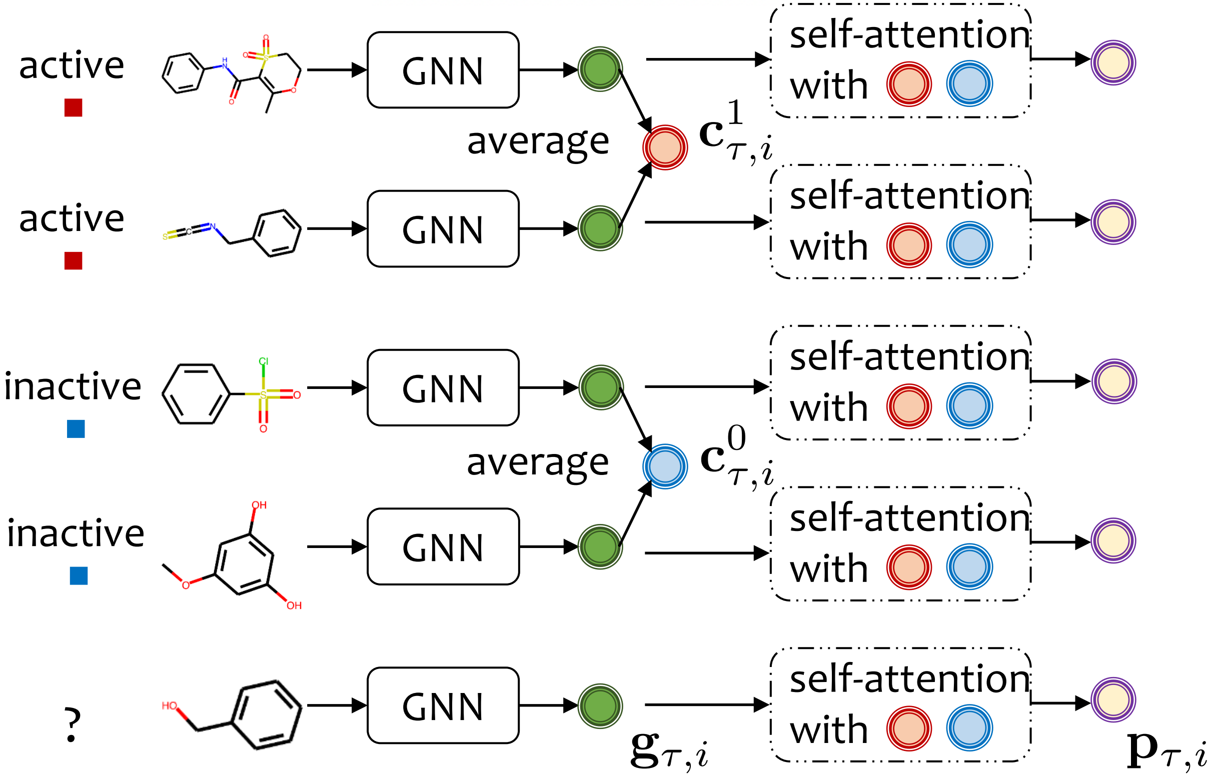
Self-Attention

- Q, K, V: the same input
- Main applications:
  - Machine reading
  - Abstractive summarization
  - Image description generation
- In PAR: Relating different molecules together



Property-aware Molecular Embedding

$$\mathbf{c}_{\mathcal{T}}^{c} = 1/|\mathcal{S}_{\mathcal{T}}^{c}| \sum\nolimits_{(\mathbf{x}_{\mathcal{T},i}, y_{\mathcal{T},i}) \in \mathcal{S}_{\mathcal{T}}^{c}} \mathbf{g}_{\mathcal{T},i}$$



As different molecular properties are attributed to different molecule substructures, we

- transform the generic molecular embeddings to substructure-aware space relevant to the target property
- contextualize each molecular embedding by dimensional wise comparing with class prototypes

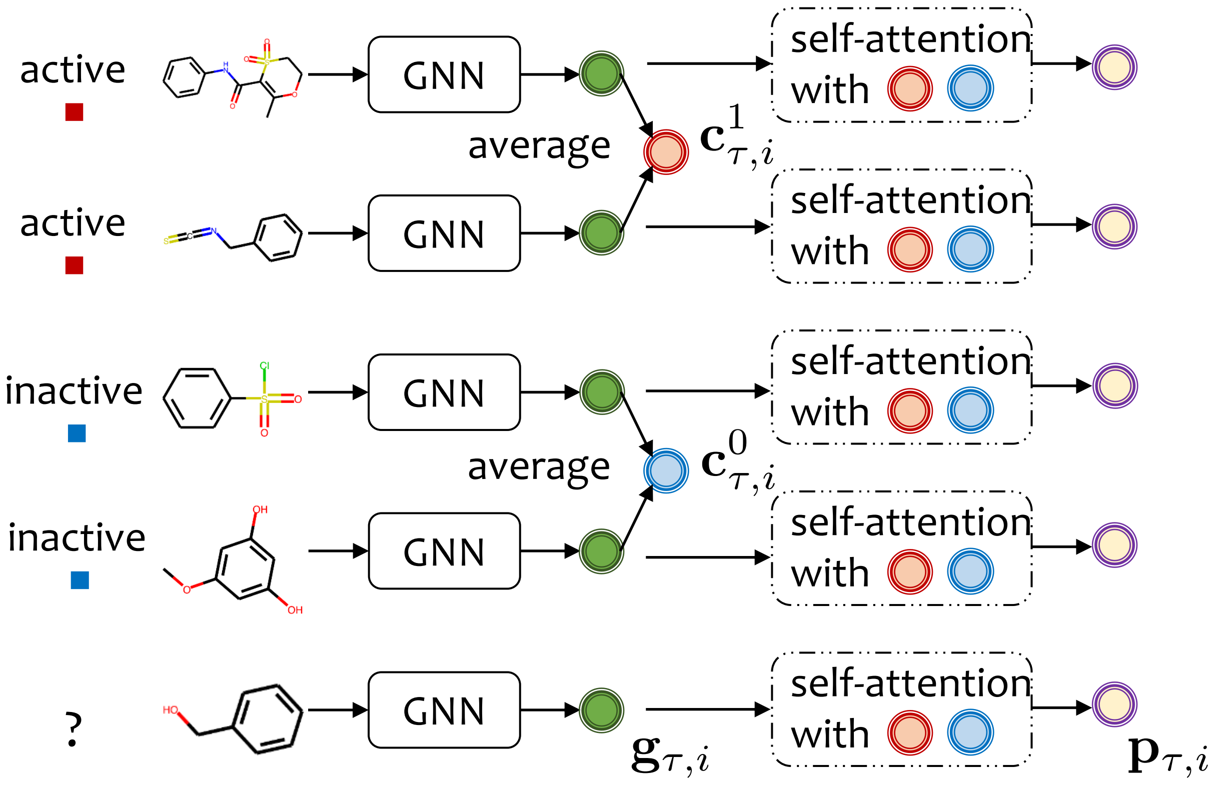
trained from large-scale tasks to capture generic information

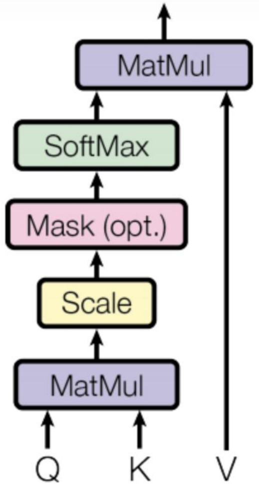
$$\mathbf{b}_{\mathcal{T},i} = \left[\texttt{softmax}(\mathbf{C}_{\mathcal{T},i}\mathbf{C}_{\mathcal{T},i}^{\top}/\sqrt{d^g})\mathbf{C}_{\mathcal{T},i}\right]_{1:} \text{ with } \mathbf{C}_{\mathcal{T},i}^{\top} = [\mathbf{g}_{\mathcal{T},i}, \mathbf{c}_{\mathcal{T}}^{0}, \mathbf{c}_{\mathcal{T}}^{1}] \in \mathbb{R}^{d^g \times 3}$$

$$\mathbf{p}_{\mathcal{T},i} = \text{MLP}_{\mathbf{W}_p}(\texttt{concat}[\mathbf{g}_{\mathcal{T},i}, \mathbf{b}_{\mathcal{T},i}])$$

Q, K, V

Self-attention step





- $g_{\tau,i}$: molecule representation after GNN
- $c^1_{\tau,i}$: representative of active molecules
- $c^0_{\tau,i}$: representative of inactive molecules
- $Q = K = V = [g_{\tau,i}, c^0_{\tau,i}, c^1_{\tau,i}]$

$$\mathbf{b}_{\tau,i} = \left[\texttt{softmax}(\mathbf{C}_{\tau,i}\mathbf{C}^\top_{\tau,i}/\sqrt{d^g})\mathbf{C}_{\tau,i}\right]_{1:} \text{ with } \mathbf{C}^\top_{\tau,i} = [\mathbf{g}_{\tau,i}, \mathbf{c}^0_\tau, \mathbf{c}^1_\tau] \in \mathbb{R}^{d^g \times 3}$$

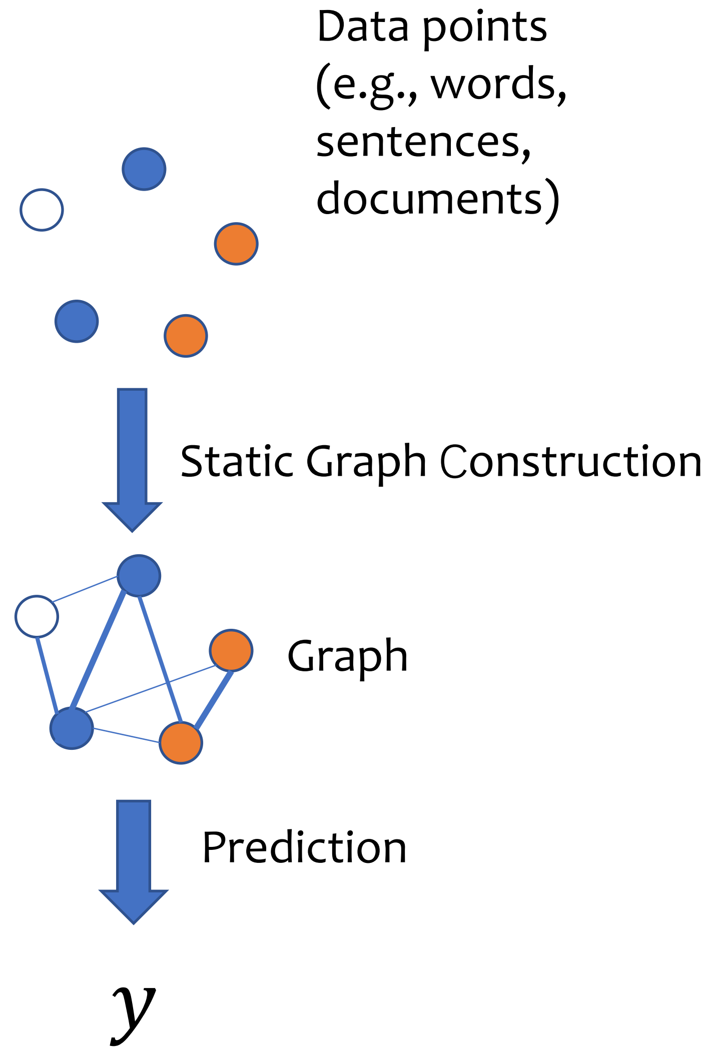
$$\mathbf{p}_{\tau,i} = \text{MLP}_{\mathbf{W}_p}(\texttt{concat}[\mathbf{g}_{\tau,i}, \mathbf{b}_{\tau,i}])$$

Static Graph Construction

Drawbacks:

- Extensive domain expertise
- Error-prone (e.g., noisy, incomplete)

  Sub-optimal
- End-to-end solution perhaps to be a better solution

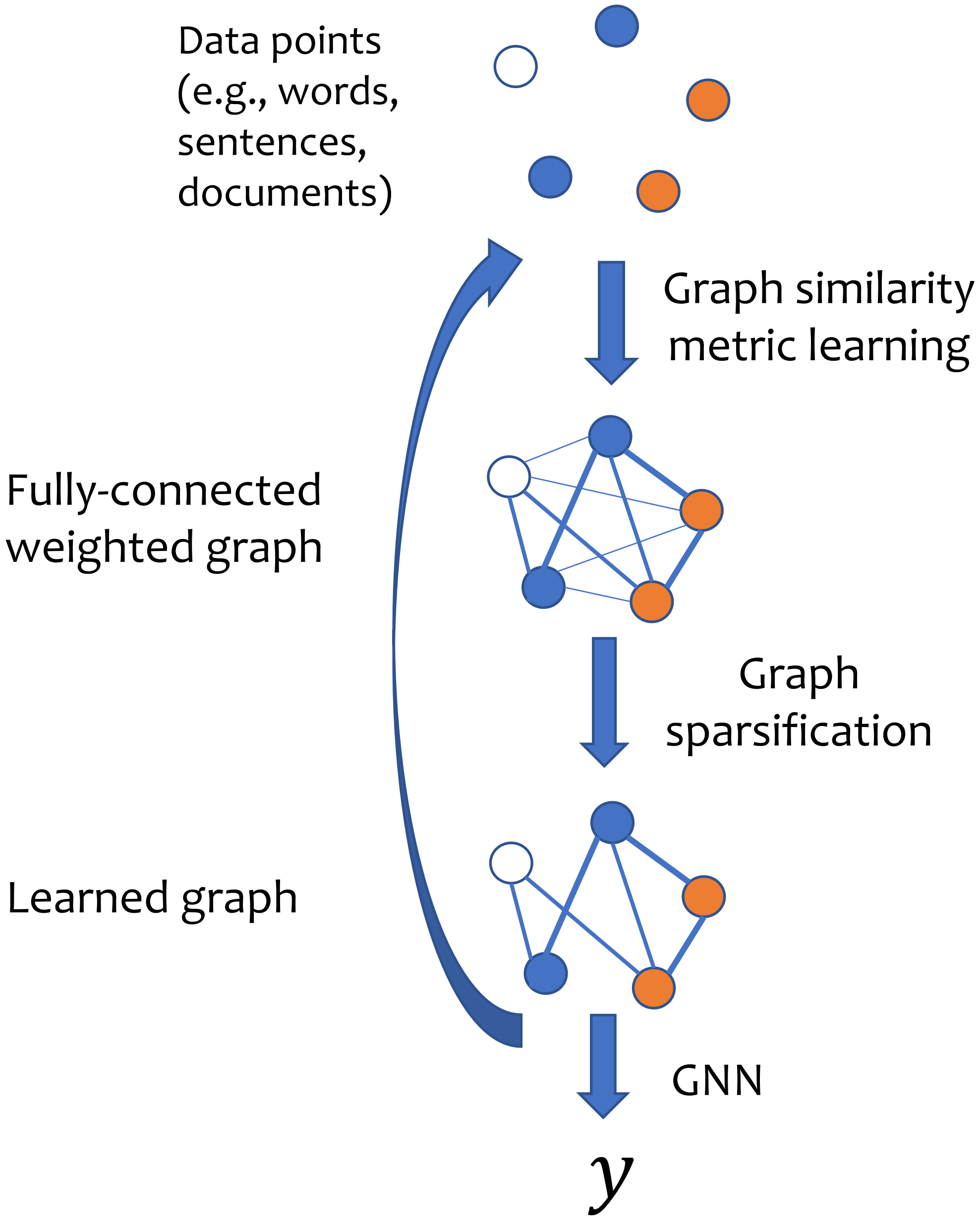


L. Wu, et al. 2021. Graph Neural Networks for Natural Language Processing: A Survey, arXiv 2021

Dynamic Graph Construction

Strength in MPP:

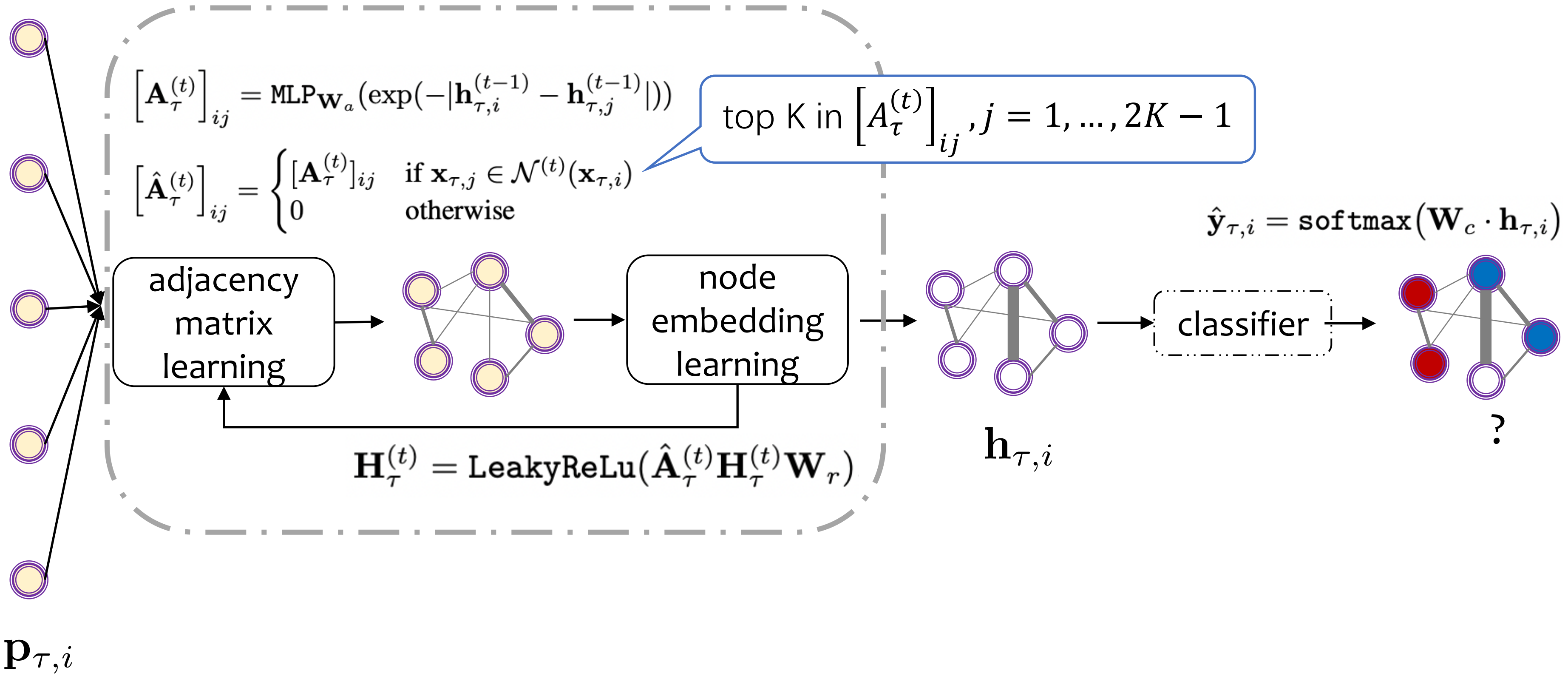
- FSL: No need to worry about computing resource constraints
- Learn better molecular representations through known intermolecular property relationships


Data points (e.g., words, sentences, documents)
Graph similarity metric learning
Fully-connected weighted graph
Graph sparsification
Learned graph
GNN
y

Relation Graph Learning

As relationship among molecules also vary w.r.t. the target property, we

- jointly estimate molecular relation graph and refine molecular embeddings w.r.t. the target property
- then can propagate limited labels efficiently between similar molecules



$$\left[\mathbf{A}_\tau^{(t)}\right]_{ij} = \mathrm{MLP}_{\mathbf{W}_a}(\exp(-|\mathbf{h}_{\tau,i}^{(t-1)} - \mathbf{h}_{\tau,j}^{(t-1)}|))$$

$$\left[\hat{\mathbf{A}}_\tau^{(t)}\right]_{ij} = \begin{cases} [\mathbf{A}_\tau^{(t)}]_{ij} & \text{if } \mathbf{x}_{\tau,j} \in \mathcal{N}^{(t)}(\mathbf{x}_{\tau,i}) \\ 0 & \text{otherwise} \end{cases}$$

top K in $\left[A_\tau^{(t)}\right]_{ij}, j = 1, \ldots, 2K-1$

$$\hat{\mathbf{y}}_{\tau,i} = \mathtt{softmax}(\mathbf{W}_c \cdot \mathbf{h}_{\tau,i})$$

$$\mathbf{H}_\tau^{(t)} = \mathtt{LeakyReLu}(\hat{\mathbf{A}}_\tau^{(t)}\mathbf{H}_\tau^{(t)}\mathbf{W}_r)$$

Training and Inference

Denote PAR as $f_{\theta,\tau}$

- $\theta = \{W_g, W_a, W_r\}$: parameters of molecular encoder and relation graph learning module
- $\Phi = \{W_p, W_c\}$: parameters of property-aware embedding function and classifier

We learn from a set of meta-training tasks a good initialized parameter

$$\min_{\boldsymbol{\theta},\boldsymbol{\Phi}} \sum\nolimits_{\tau=1}^{N_t} \mathcal{L}(\mathcal{Q}_\tau, f_{\boldsymbol{\theta},\boldsymbol{\Phi}_\tau})$$

Within each task, we fix θ while fine-tune Φ as Φ_τ

Ground-truth labels

$$\mathcal{L}(\mathcal{S}_\tau, f_{\boldsymbol{\theta},\boldsymbol{\Phi}}) = \sum\nolimits_{(\mathbf{x}_{\tau,i}, y_{\tau,i})\in\mathcal{S}_\tau} -\mathbf{y}_{\tau,i}^\top \cdot \log(\hat{\mathbf{y}}_{\tau,i}) + \underline{\|[\mathbf{A}^*_\tau]_{i:} - [\hat{\mathbf{A}}_\tau]_{i:}\|_2^2} \qquad \boldsymbol{\Phi}_\tau = \boldsymbol{\Phi} - \alpha\nabla_{\boldsymbol{\Phi}}\mathcal{L}(\mathcal{S}_\tau, f_{\boldsymbol{\theta},\boldsymbol{\Phi}})$$

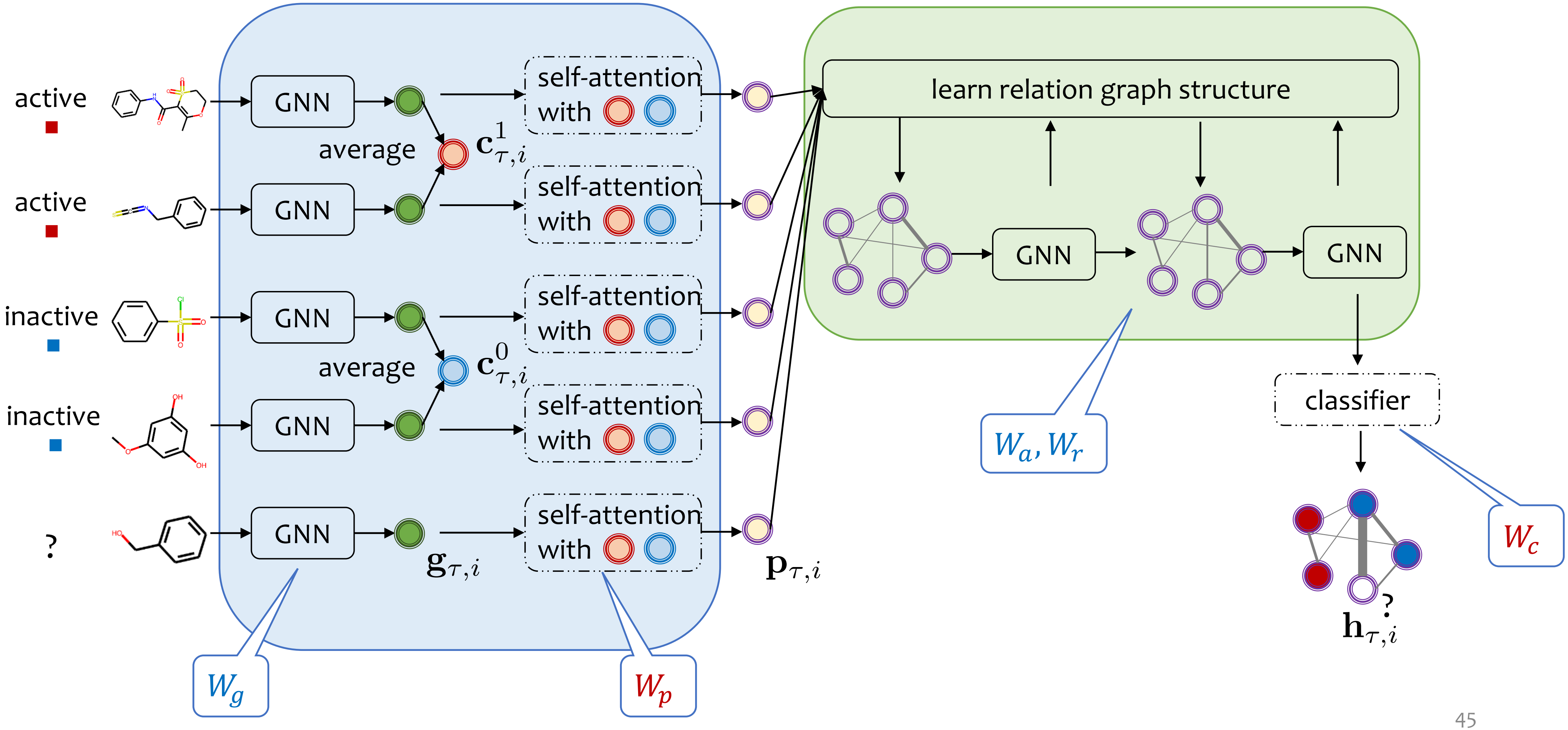
classification loss neighbor alignment regularizer

to separately capture the generic knowledge shared across different tasks and those property-aware

PAR Framework

$\theta = \{W_g, W_a, W_r\}$, generic knowledge

$\Phi = \{W_p, W_c\}$, property-aware knowledge



Training and Inference

Denote PAR as $f_{\theta,\tau}$

- $\theta = \{W_g, W_a, W_r\}$: parameters of molecular encoder and relation graph learning module
- $\Phi = \{W_p, W_c\}$: parameters of property-aware embedding function and classifier

We learn from a set of meta-training tasks a good initialized parameter

$$\min_{\boldsymbol{\theta},\boldsymbol{\Phi}} \sum\nolimits_{\tau=1}^{N_t} \mathcal{L}(\mathcal{Q}_\tau, f_{\boldsymbol{\theta},\boldsymbol{\Phi}_\tau})$$

Within each task, we fix θ while fine-tune Φ as Φ_τ

Ground-truth labels

$$\mathcal{L}(\mathcal{S}_\tau, f_{\boldsymbol{\theta},\boldsymbol{\Phi}}) = \sum\nolimits_{(\mathbf{x}_{\tau,i},y_{\tau,i})\in\mathcal{S}_\tau} -\mathbf{y}_{\tau,i}^\top \cdot \log(\hat{\mathbf{y}}_{\tau,i}) + \underline{\|[\mathbf{A}_\tau^*]_{i:} - [\hat{\mathbf{A}}_\tau]_{i:}\|_2^2} \qquad \boldsymbol{\Phi}_\tau = \boldsymbol{\Phi} - \alpha\nabla_{\boldsymbol{\Phi}}\mathcal{L}(\mathcal{S}_\tau, f_{\boldsymbol{\theta},\boldsymbol{\Phi}})$$

classification loss neighbor alignment regularizer

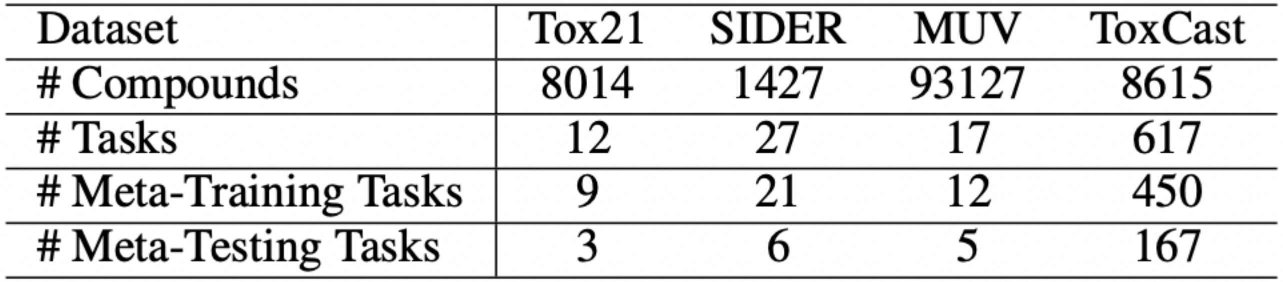
to separately capture the generic knowledge shared across different tasks and those property-aware

Comparison with related works

approaches	IterRefLSTMs	Meta-MGNN	PAR
FSL methods	Matching Networks	MAML	MAML
Property aware	×	×	✓
Molecule relations	×	×	✓
Selective update	×	×	✓
Pretrain	×	✓	Optional

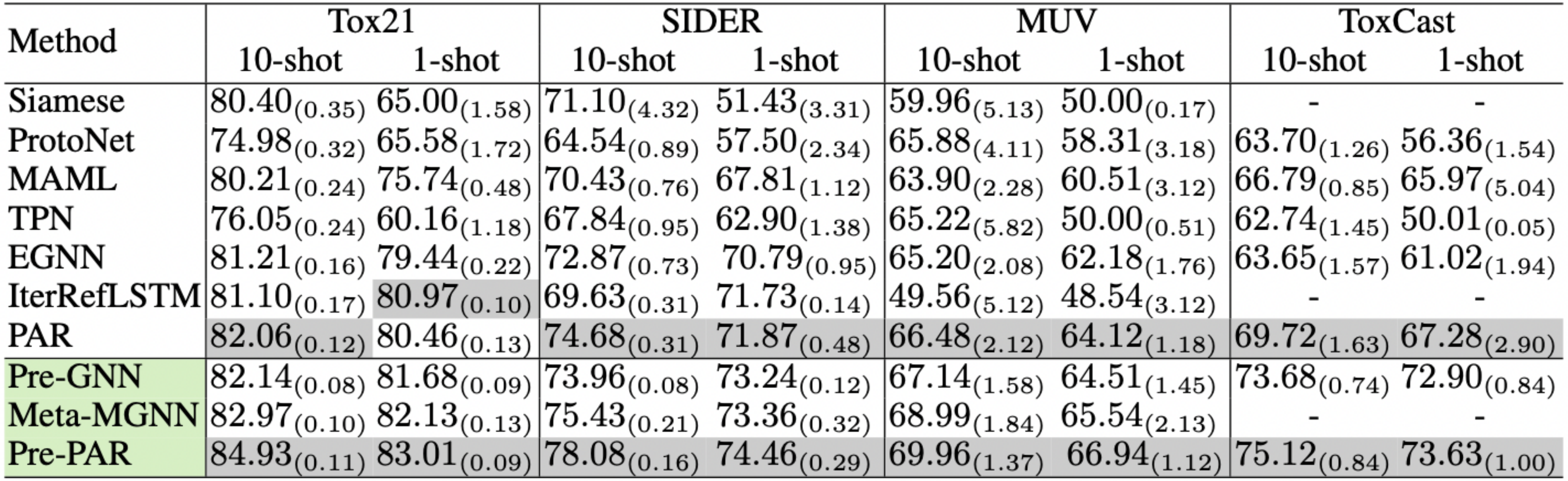
Experiment setup

- Two sets of baselines
  - Methods with graph- based encoder learned from scratch including Siamese [Koch et al., 2015], ProtoNet [Snell et al., 2017], MAML [Finn et al., 2017], TPN [Liu et al., 2018], and EGNN [Kim et al., 2019], IterRefLSTM [Altae-Tran et al., 2017];
  - Methods which leverage pretained graph-based molecular encoder including Pre-GNN [Hu et al., 2019], Meta-MGNN [Guo et al., 2021], and Pre-PAR which is our PAR equipped with Pre- GNN.
- Four datasets



Dataset	Tox21	SIDER	MUV	ToxCast
# Compounds	8014	1427	93127	8615
# Tasks	12	27	17	617
# Meta-Training Tasks	9	21	12	450
# Meta-Testing Tasks	3	6	5	167

FSL Results



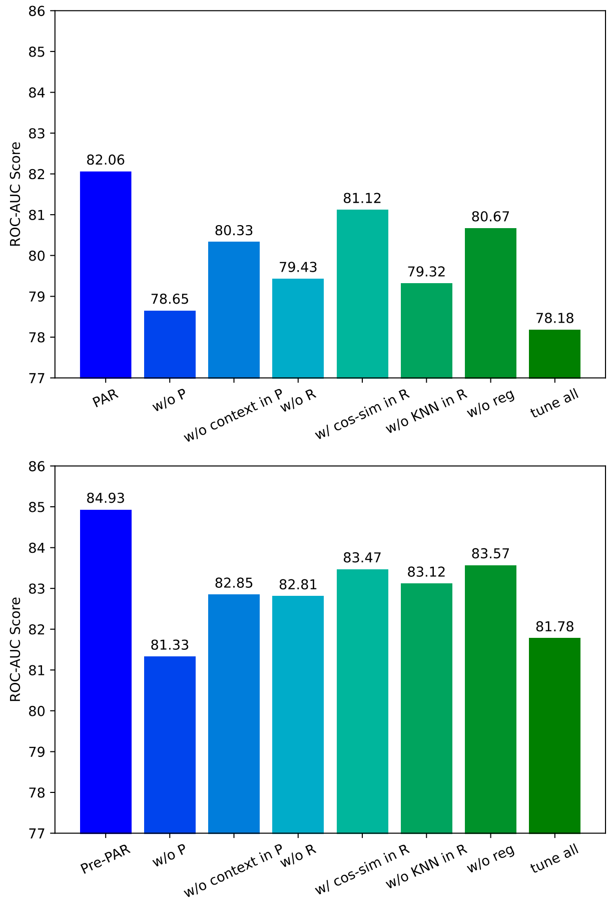
Method	Tox21		SIDER		MUV		ToxCast	
	10-shot	1-shot	10-shot	1-shot	10-shot	1-shot	10-shot	1-shot
Siamese	$80.40_{(0.35)}$	$65.00_{(1.58)}$	$71.10_{(4.32)}$	$51.43_{(3.31)}$	$59.96_{(5.13)}$	$50.00_{(0.17)}$	-	-
ProtoNet	$74.98_{(0.32)}$	$65.58_{(1.72)}$	$64.54_{(0.89)}$	$57.50_{(2.34)}$	$65.88_{(4.11)}$	$58.31_{(3.18)}$	$63.70_{(1.26)}$	$56.36_{(1.54)}$
MAML	$80.21_{(0.24)}$	$75.74_{(0.48)}$	$70.43_{(0.76)}$	$67.81_{(1.12)}$	$63.90_{(2.28)}$	$60.51_{(3.12)}$	$66.79_{(0.85)}$	$65.97_{(5.04)}$
TPN	$76.05_{(0.24)}$	$60.16_{(1.18)}$	$67.84_{(0.95)}$	$62.90_{(1.38)}$	$65.22_{(5.82)}$	$50.00_{(0.51)}$	$62.74_{(1.45)}$	$50.01_{(0.05)}$
EGNN	$81.21_{(0.16)}$	$79.44_{(0.22)}$	$72.87_{(0.73)}$	$70.79_{(0.95)}$	$65.20_{(2.08)}$	$62.18_{(1.76)}$	$63.65_{(1.57)}$	$61.02_{(1.94)}$
IterRefLSTM	$81.10_{(0.17)}$	$80.97_{(0.10)}$	$69.63_{(0.31)}$	$71.73_{(0.14)}$	$49.56_{(5.12)}$	$48.54_{(3.12)}$	-	-
PAR	$82.06_{(0.12)}$	$80.46_{(0.13)}$	$74.68_{(0.31)}$	$71.87_{(0.48)}$	$66.48_{(2.12)}$	$64.12_{(1.18)}$	$69.72_{(1.63)}$	$67.28_{(2.90)}$
Pre-GNN	$82.14_{(0.08)}$	$81.68_{(0.09)}$	$73.96_{(0.08)}$	$73.24_{(0.12)}$	$67.14_{(1.58)}$	$64.51_{(1.45)}$	$73.68_{(0.74)}$	$72.90_{(0.84)}$
Meta-MGNN	$82.97_{(0.10)}$	$82.13_{(0.13)}$	$75.43_{(0.21)}$	$73.36_{(0.32)}$	$68.99_{(1.84)}$	$65.54_{(2.13)}$	-	-
Pre-PAR	$84.93_{(0.11)}$	$83.01_{(0.09)}$	$78.08_{(0.16)}$	$74.46_{(0.29)}$	$69.96_{(1.37)}$	$66.94_{(1.12)}$	$75.12_{(0.84)}$	$73.63_{(1.00)}$

- Pre-PAR consistently obtains the best performance
- PAR outperforms among methods without pretrained GNNs

Ablation Study

We further compare with

- **w/o P:** w/o property-aware embedding
- **w/o context in P:** w/o context $b_{\tau,i}$ in P
- **w/o R:** w/o adaptive relation graph learning
- **w/ cos-sim in R:** use cosine similarity to obtain the adjacency matrix
- **w/o KNN in R:** w/o reducing the learned relation graph to KNN graph
- **w/o reg:** w/o the neighbor alignment regularizer
- **tune all:** fine-tune all parameters



PAR

Pre-PAR

10-shot tasks from Tox21

All components are vital to the success of PAR

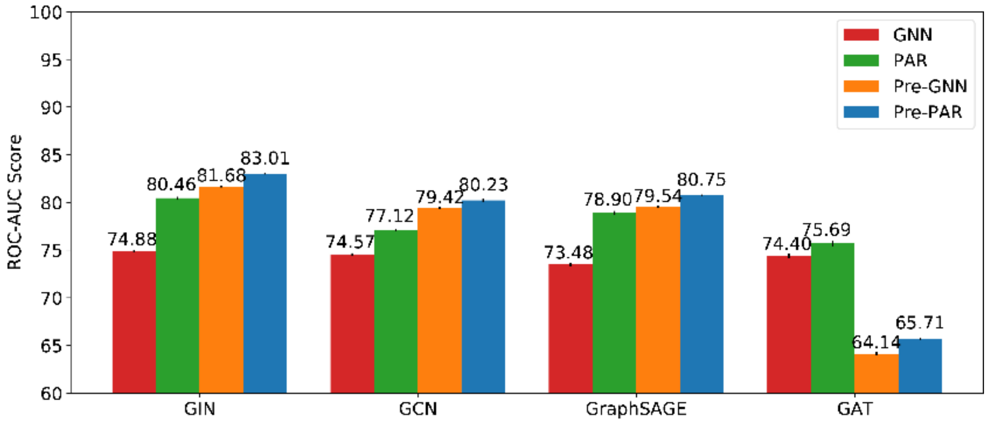
Varying Graph-based Molecular Encoders

We compare PAR with fine-tuning the encoder (denote as GNN)

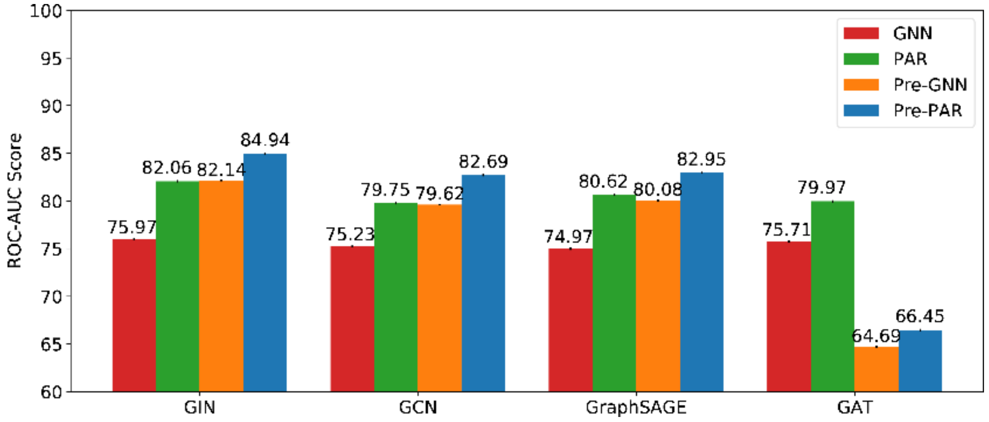
- GIN [Xu et al., 2018] (used)
- GCN [Duvenaud et al., 2015]
- GraphSAGE [Hamilton et al., 2017]
- GAT [Veliˇckovi´c et al., 2017]

GIN is the consistently better than the others

PAR consistently outperforms GNN



1-shot

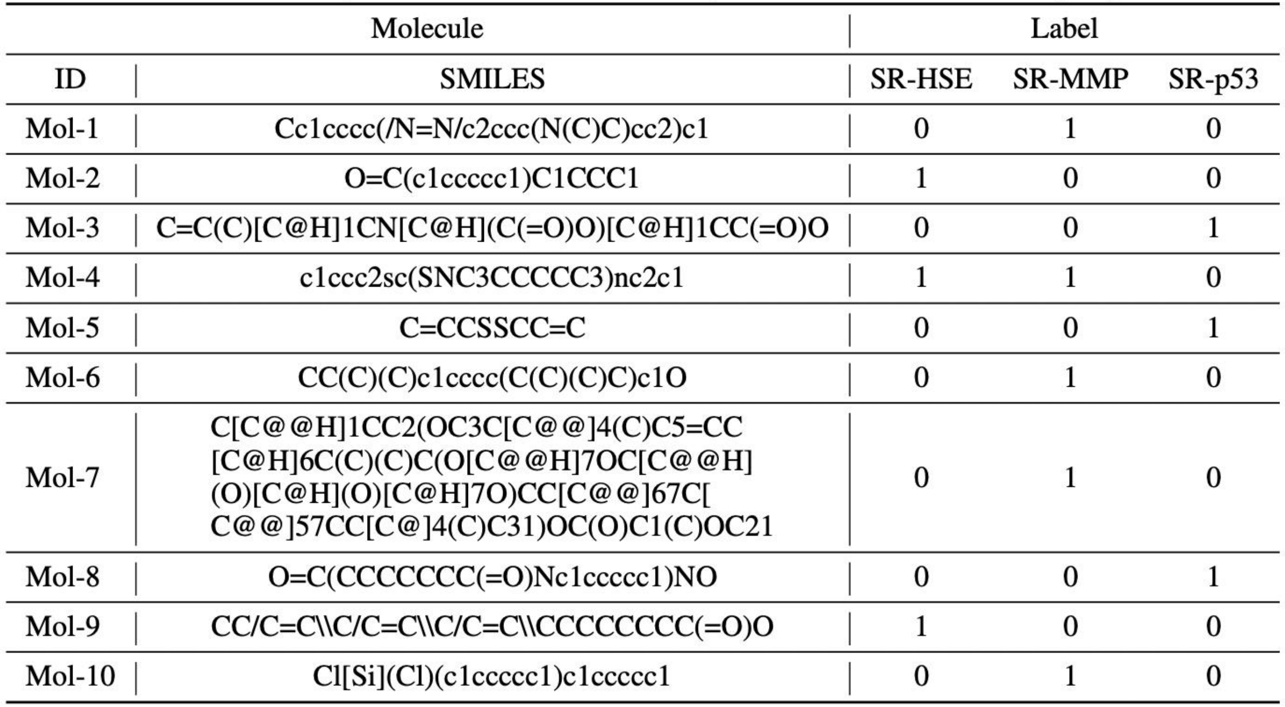


10-shot

Case Study on 10 Molecules

Can PAR obtain different property-aware molecular embeddings and relation graphs for tasks containing overlapping molecules but evaluating different properties?

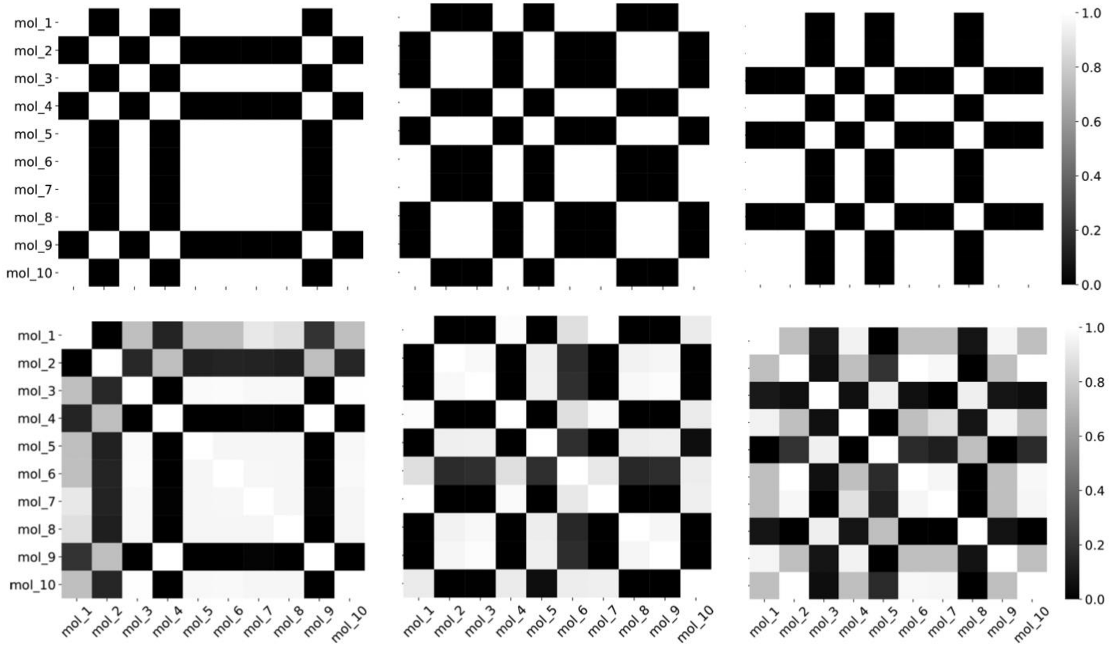
Table 5: The 10 molecules sampled from Tox21 dataset, which coexist in the three meta-testing tasks (the 10th task for SR-HSE, the 11th task for SR-MMP, and the 12th task for SR-p53).



Molecule		Label		
ID	SMILES	SR-HSE	SR-MMP	SR-p53
Mol-1	Cc1cccc(/N=N/c2ccc(N(C)C)cc2)c1	0	1	0
Mol-2	O=C(c1ccccc1)C1CCC1	1	0	0
Mol-3	C=C(C)[C@H]1CN[C@H](C(=O)O)[C@H]1CC(=O)O	0	0	1
Mol-4	c1ccc2sc(SNC3CCCCC3)nc2c1	1	1	0
Mol-5	C=CCSSCC=C	0	0	1
Mol-6	CC(C)(C)c1cccc(C(C)(C)C)c1O	0	1	0
Mol-7	C[C@@H]1CC2(OC3C[C@@]4(C)C5=CC[C@H]6C(C)(C)C(O[C@@H]7OC[C@@H](O)[C@H](O)[C@H]7O)CC[C@@]67C[C@@]57CC[C@]4(C)C31)OC(O)C1(C)OC21	0	1	0
Mol-8	O=C(CCCCCCC(=O)Nc1ccccc1)NO	0	0	1
Mol-9	CC/C=C\\C/C=C\\C/C=C\\CCCCCCCC(=O)O	1	0	0
Mol-10	Cl[Si](Cl)(c1ccccc1)c1ccccc1	0	1	0

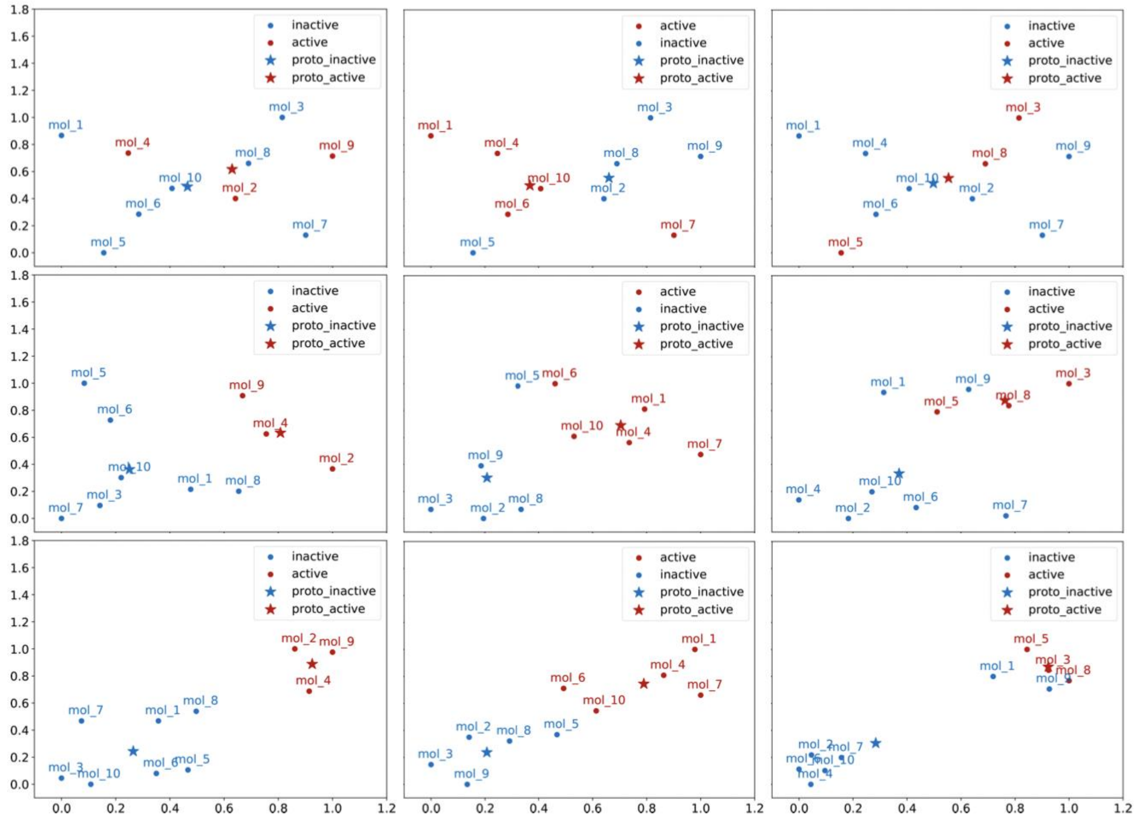
a fixed group of 10 molecules coexist in different meta-testing tasks

Visualization



(a) the 10th task (b) the 11th task (c) the 12th task

Figure 5: Comparison between $\mathbf{A}^*_\tau$ computed using ground-truth labels (the first row) and adjacency matrix $\mathbf{A}_\tau$ returned by PAR (the second row) for the ten molecules. We set $[\mathbf{A}^*_\tau]_{ij} = 1$ if molecules $\mathbf{x}_{\tau,i}$ and $\mathbf{x}_{\tau,j}$ have the same label and 0 otherwise.



(a) the 10th task (b) the 11th task (c) the 12th task

Figure 6: t-SNE visualization of $\mathbf{g}_{\tau,i}$ (the first row), $\mathbf{p}_{\tau,i}$ (the second row), and $\mathbf{h}_{\tau,i}$ (the third row) of the ten molecules. Proto_active (proto_inactive) denotes the class prototype of active (inactive) class.

PAR can model property-aware molecular embeddings and relation graphs

Take home message

We propose Property-Aware Relation network (PAR) for few-shot molecular property prediction problem

- Models substructures and molecule relationships w.r.t the target property
- Adopts a selective-update training strategy to separately capture generic and property-aware knowledge
- Consistently outperforms the others

Summary

- Background: Molecule property prediction (MPP)
- Preliminary: Few-shot learning (FSL)
  - Difficulty; MAML
- Related works: FSL for MPP
  - IterRefLSTM, Meta-MGNN
- The proposed approach PAR

Future works

- Transfer learning across datasets
- Few-shot molecule regression
- MPP with other FSL structures
    - RelationNet, …
- Design GNNs for better molecular representations

Thanks! Questions?