ICLR 2022

PF-GNN: Differentiable Particle Filtering based Approximation of Universal Graph Representations

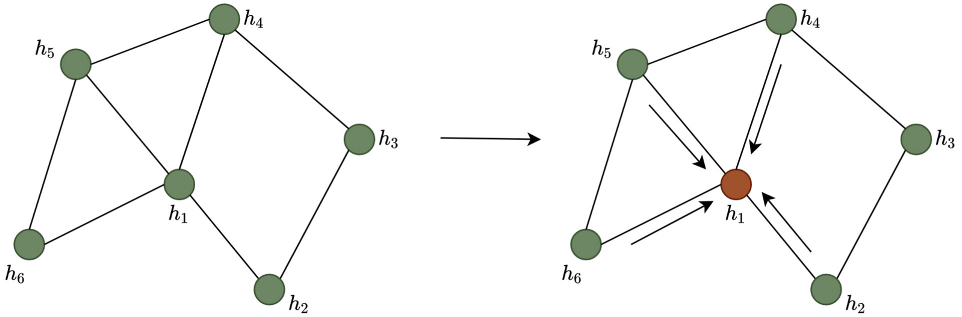
Mohammed Haroon Dupty[1], Yanfei Dong[1,2], Wee Sun Lee[1]

[1]National University of Singapore, [2]Paypal Innovation Lab

Graph Neural Networks

Message Passing

$$m_v^{t+1} = AGGR_{u \in N(v)}\left(f_{msg}^t(h_v, h_u, e_{uv} | \theta_m)\right)$$

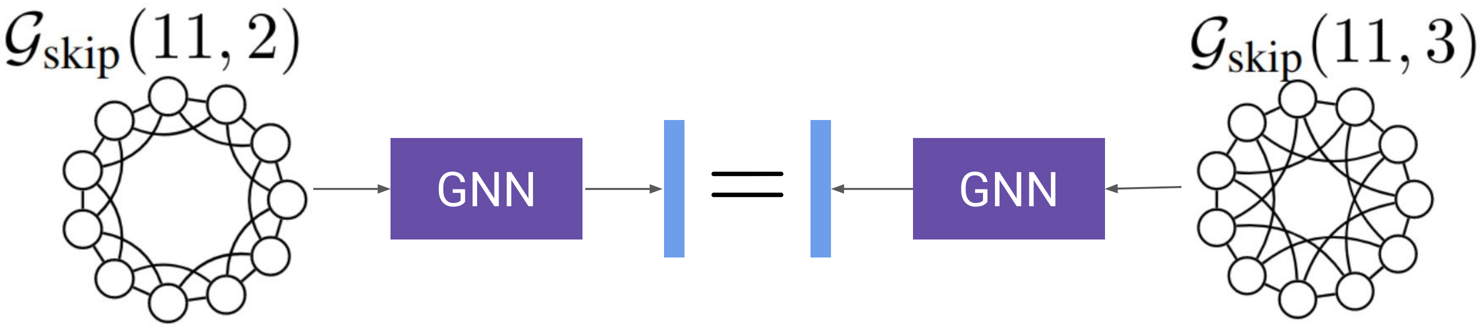


$$h_v^{t+1} = f_{upd}^t(h_v^t, m_v^t | \theta_u)$$

Scarselli et al. (2008), Li et al. (2015), Kipf et al. (2016), Defferand et al. (2016), Xu et al (2017), Gilmer et. al (2017)

Representation Capacity of GNNs

GNNs learn functions on graphs



GNNs cannot distinguish many non-isomorphic graphs

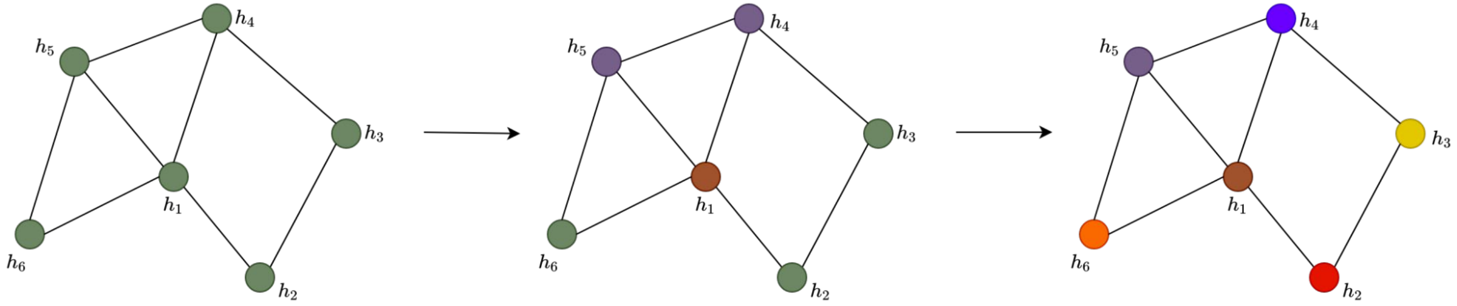
Why?

Murphy et. al (2019), Chen et. al (2020)

1-WL Color Refinement

GNN = 1-dim 1-WL Color Refinement

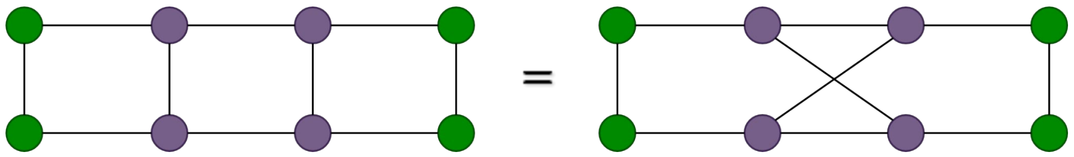
$$\pi^{t+1}(v) = \mathrm{HASH}\Big(\pi^t(v),\ \{\!\!\{\pi^t(u), u \in N(v)\}\!\!\}\Big)$$



1-WL Color Refinement

GNN = 1-dim 1-WL Color Refinement

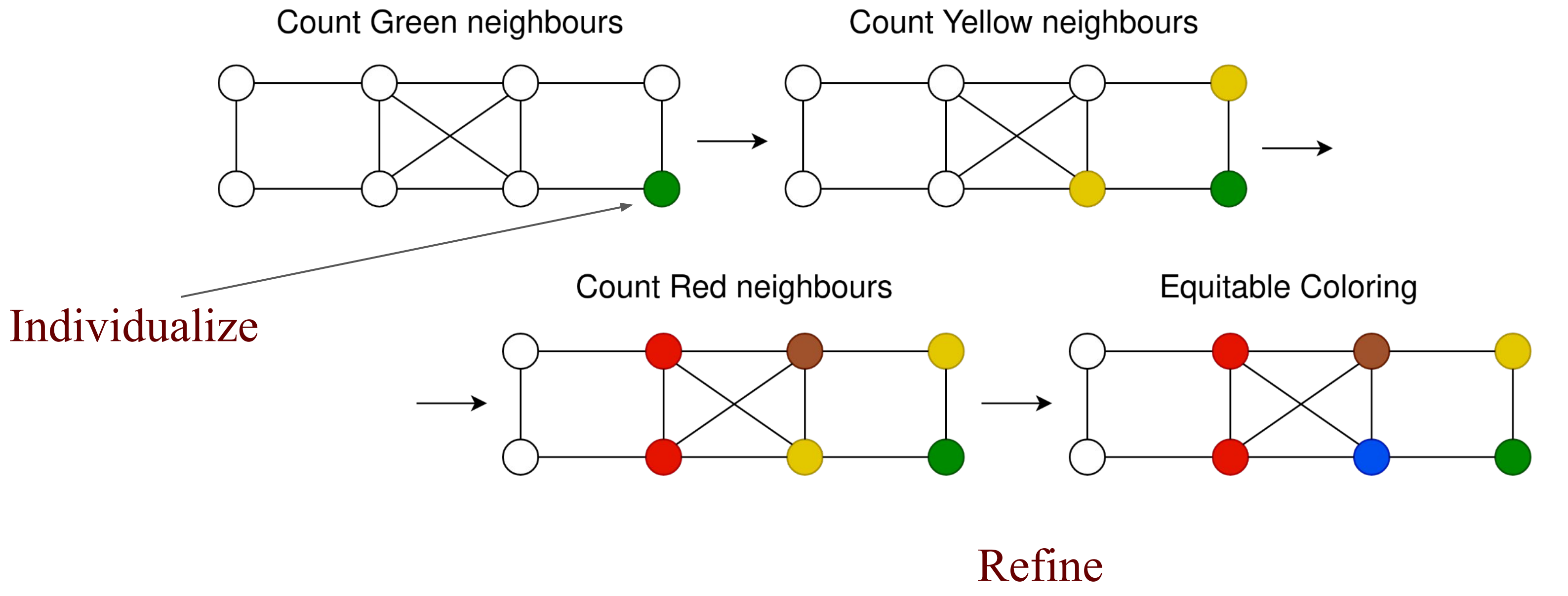
$$\pi^{t+1}(v) = \mathrm{HASH}\left(\pi^{t}(v),\ \{\!\{\pi^{t}(u), u \in N(v)\}\!\}\right)$$



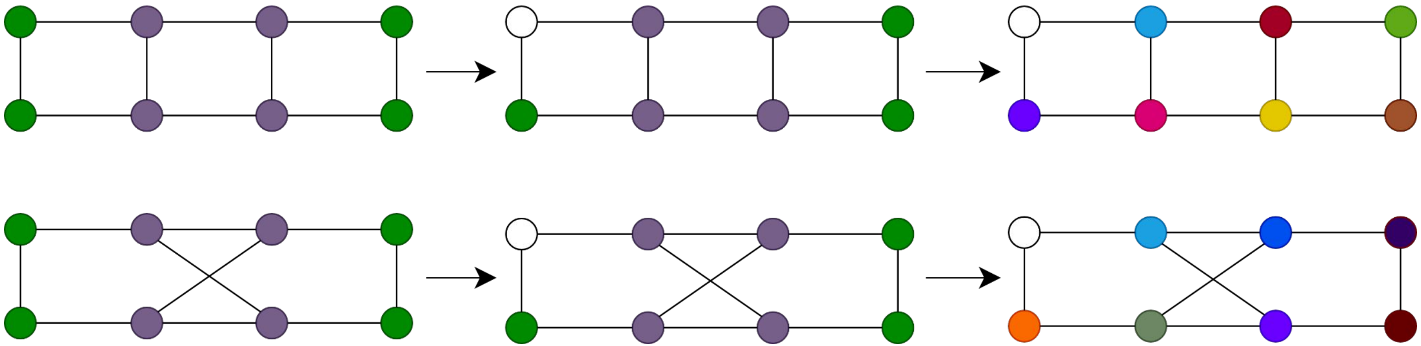
Equitable partition can not be further refined

Many graphs map to same coloring

Individualization and Refinement



Individualization and Refinement



GNN/1-WL cannot distinguish two graphs

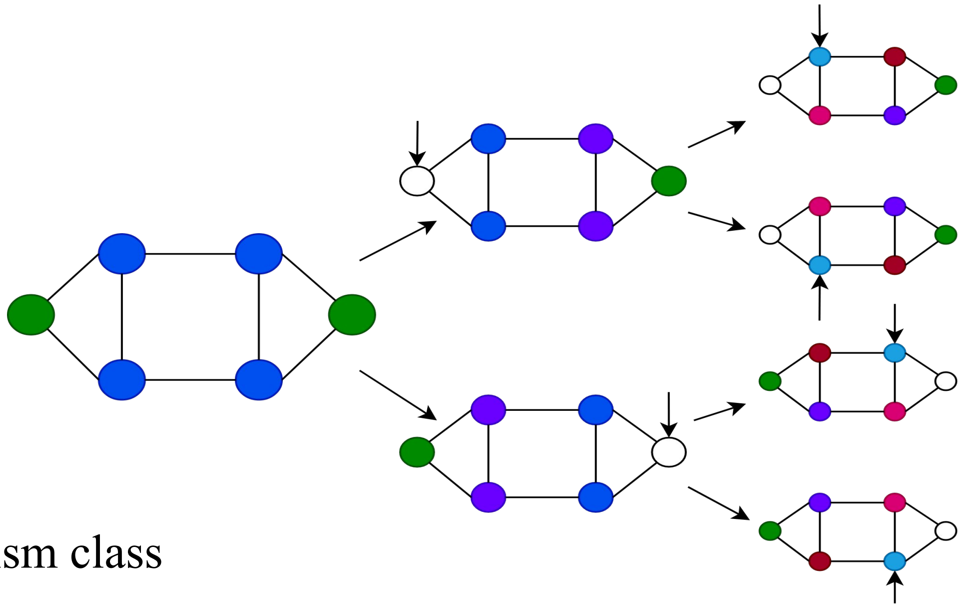
Graphs are **distinguishable** after one step of **IR**

Search Tree of Colorings

Preserve Permutation Invariance

↓

Search Tree of Colorings



Search Tree unique to Isomorphism class

PF-GNN

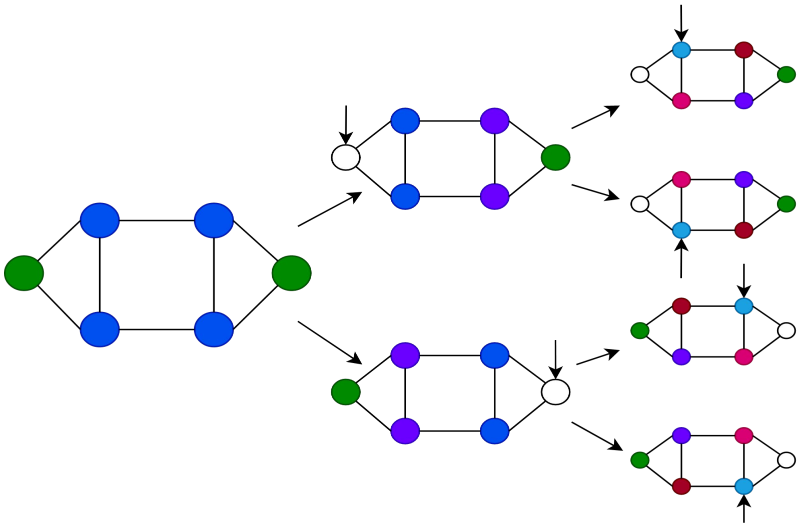
Universal representation on graphs

$$f(\mathcal{G}) = \rho\left(\sum_{\forall \mathcal{I}} \psi(\mathcal{G}, \boldsymbol{\pi}_T^{\mathcal{I}})\right)$$

Probabilistic representation

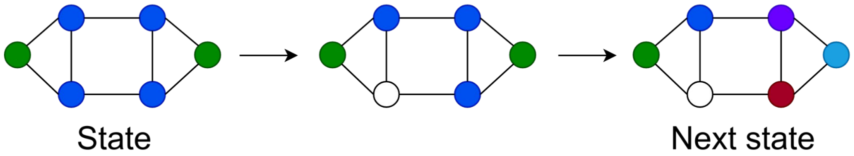
$$\tilde{f}(\mathcal{G}) = \frac{1}{|\Pi^{\mathcal{I}}|} \sum_{\mathcal{I} \in \Pi^{\mathcal{I}}} \psi(\mathcal{G}, \boldsymbol{\pi}_T^{\mathcal{I}}) = \mathbb{E}[\, \psi(\mathcal{G}, \boldsymbol{\pi}_T^{\mathcal{I}})]$$

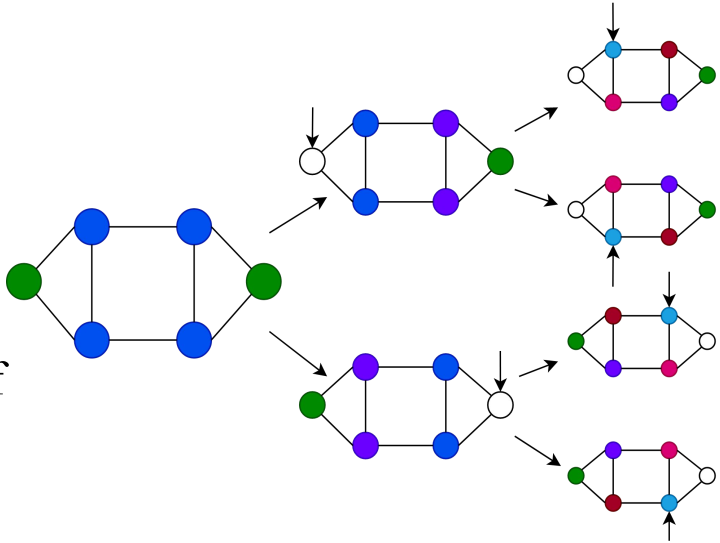
(ϵ, δ) approximation with $K \in \mathcal{O}\left(\frac{M^2 \ln(\frac{4D}{\delta})}{\epsilon^2}\right)$ sampled paths



PF-GNN

Observation: IR resembles **State Transition**

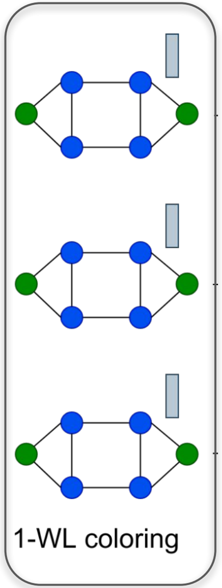




Particle Filters provide principled method of approximating the **belief** on embeddings.

$$b_t(\mathcal{G}) \approx \langle (\mathcal{G}, \boldsymbol{H}_t^k), w_t^k \rangle_{k=1:K}$$

PF-GNN



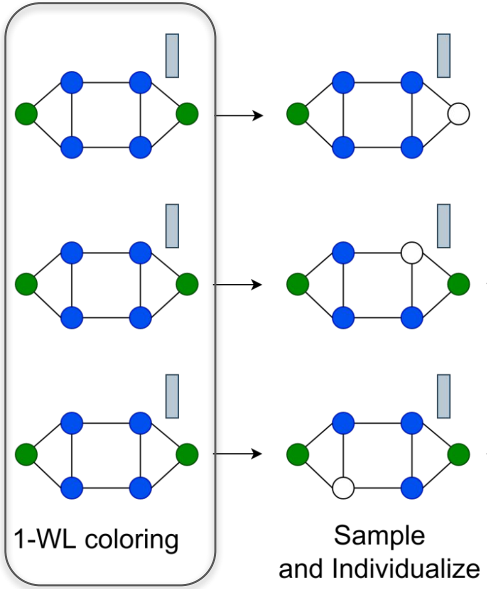
Initial Belief

Start with **K** GNN equitable refinements

$$b_1(\mathcal{G}) = \langle(\mathcal{G}, \boldsymbol{H}_1^k), w_1^k\rangle_{k=1:K}$$

$$\forall k, w_1^k = 1/K$$

PF-GNN



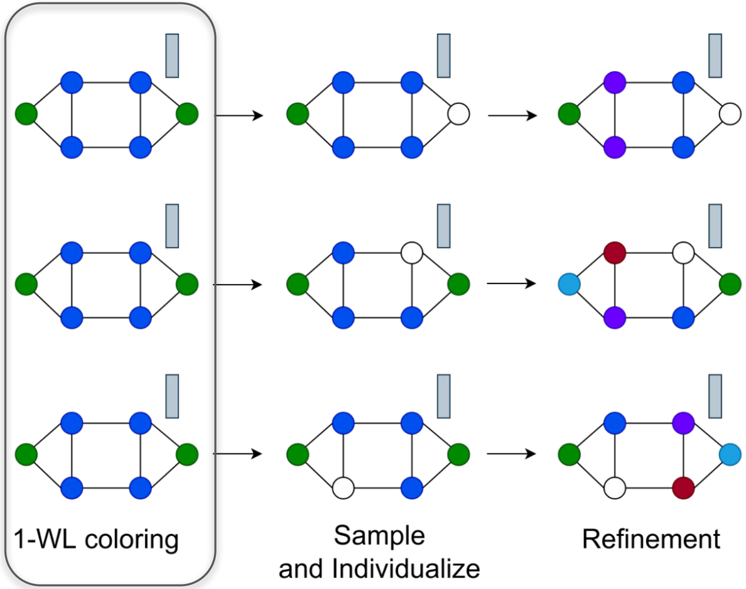
Transition to next state of colorings

- Sample a node and individualize

$$v \sim P(\mathcal{V}|\boldsymbol{H}_t^k;\theta)$$

$$\boldsymbol{M}_t^k = \boldsymbol{1}\boldsymbol{1}^\top; \quad \boldsymbol{M}_{t\ v,:}^k = MLP_{trans}(h_{vt}^k); \quad \boldsymbol{H}_t^k = \boldsymbol{H}_t^k \odot \boldsymbol{M}_t^k;$$

PF-GNN



Transition to next state of colorings

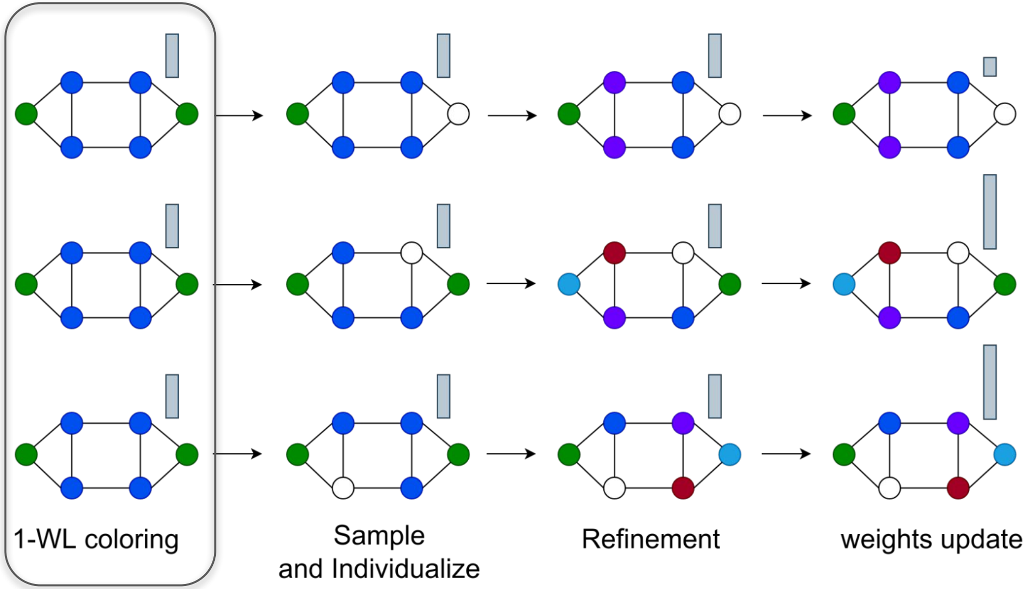
- Sample a node and individualize
- Refine the colorings

$$v \sim P(\mathcal{V}|\boldsymbol{H}_t^k;\theta)$$

$$\boldsymbol{M}_t^k = \mathbf{1}\mathbf{1}^\top;\ \ \boldsymbol{M}_{t\ v,:}^k = MLP_{trans}(h_{vt}^k);\ \ \boldsymbol{H}_t^k = \boldsymbol{H}_t^k \odot \boldsymbol{M}_t^k;$$

$$\boldsymbol{H}_{t+1}^k = GNN_t(\boldsymbol{H}_t^k)$$

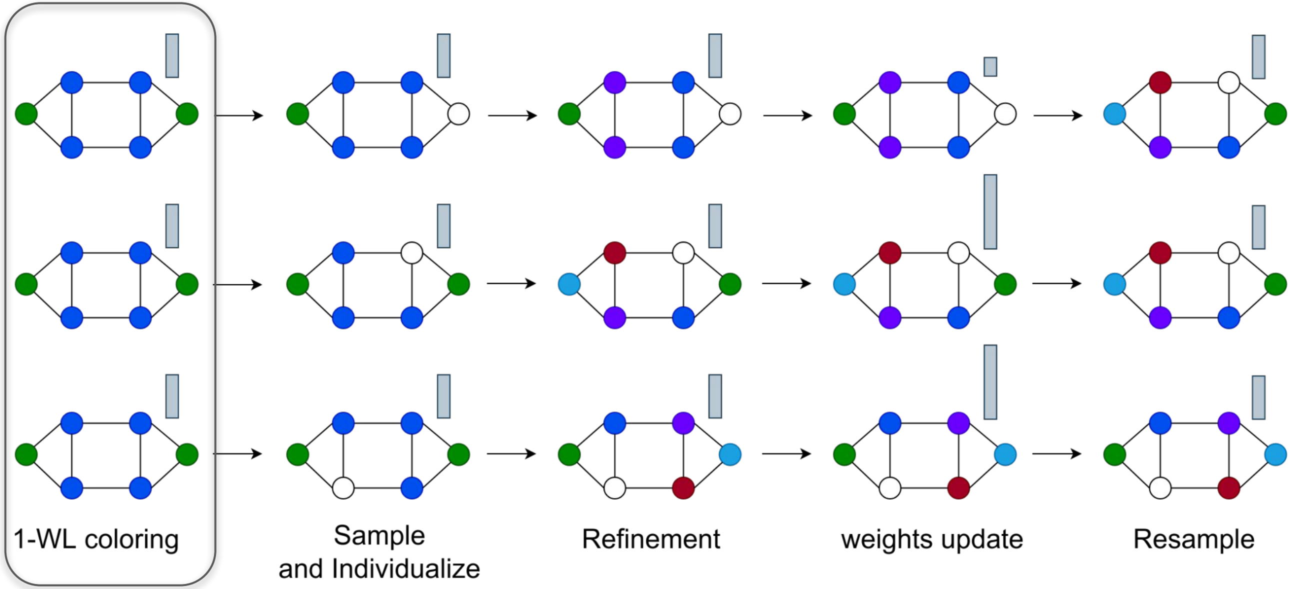
PF-GNN



Weights update with discriminative observation function

$$w_{t+1}^k = \frac{f_{obs}(\boldsymbol{H}_{t+1}^k; \theta_o) \cdot w_t^k}{\sum_k f_{obs}(\boldsymbol{H}_{t+1}^k; \theta_o) \cdot w_t^k}$$

PF-GNN

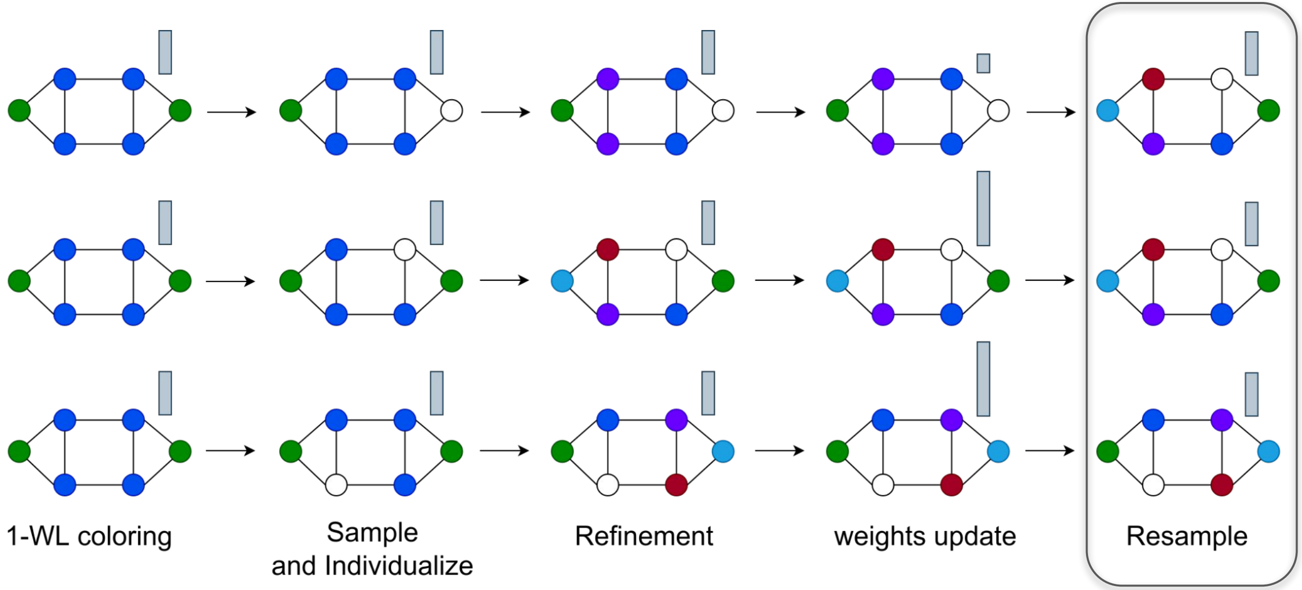


Particle Resampling

$$q_t(k) = \alpha w_t^k + (1-\alpha)1/K$$

$$w_t'^k = \frac{p_t(k)}{q_t(k)} = \frac{w_t^k}{\alpha w_t^k + (1-\alpha)1/K}$$

PF-GNN



Updated belief

$$b_{t+1}(\mathcal{G}) \approx \langle(\mathcal{G}, \boldsymbol{H}_{t+1}^k), w_{t+1}^k\rangle_{k=1:K}$$

PF-GNN

Minimize expected loss on sampled paths

$$Loss(\mathcal{G}, y) = \sum_{\mathcal{I}} P(\mathcal{I}|\mathcal{G}, \pi^{\mathcal{I}}_{t=1:T}; \theta) L(\tilde{y}(\pi^{\mathcal{I}}_{t=1:T}), y; ; \theta)$$

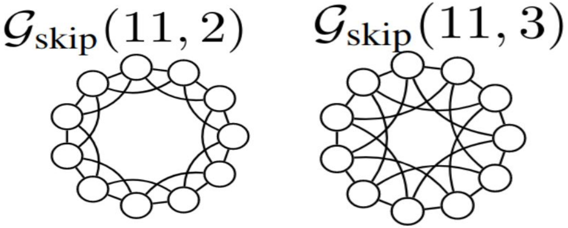
Train with policy gradient loss

$$\nabla Loss(\mathcal{G}, y) = \sum_{\mathcal{I}} \nabla L(\tilde{y}(\pi^{\mathcal{I}}_{i=1:T}), y; \theta) + \sum_{\mathcal{I}} \Big(\nabla \log P(\mathcal{I}|\mathcal{G}, \pi^{\mathcal{I}}_{i=1:T}; \theta)\Big) L(\tilde{y}(\pi^{\mathcal{I}}_{i=1:T}), y; \theta)$$

Experiments

Graph Classification on Circulant Skip Link Graphs

	GCN	GAT	GIN*	RING-GNN	RP-GIN	3-WL GNN	PF-GNN
MEAN	10	10	10	10	37.6	97.8	**100.0**
MEDIAN	10	10	10	10	43.3	-	**100.0**
MAX	10	10	10	10	53.3	**100.0**	**100.0**
MIN	10	10	10	10	10	30	**100.0**
STD	0	0	0	0	12.9	10.9	0



Graph Isomorphism detection

DATASET	MODEL	CHEBNET	PPGN (3-WL)	GNNML3	GCN	GAT	GIN	GNNML1
SR25	BACKBONE	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0
	+PF-GNN	-	-	-	**100.0**±0.0	**100.0**±0.0	**100.0**±0.0	**100.0**±0.0
EXP	BACKBONE	0.0±0.0	0.0±0.0	**100.0**±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0
	+PF-GNN	-	-	-	**100.0**±0.0	**100.0**±0.0	**100.0**±0.0	**100.0**±0.0

Experiments

Graph prediction on Real-world datasets

Predicting molecular properties

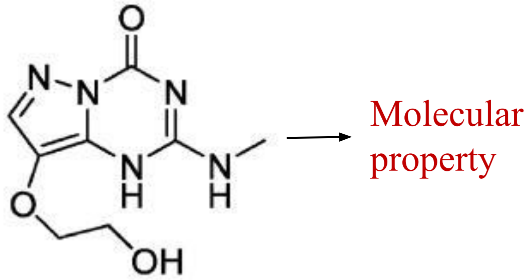
QM9

Method	MAE
GINE-ϵ	0.081 ±0.003
MPNN	0.034 ±0.001
1-2-GNN	0.068 ±0.001
1-3-GNN	0.088 ±0.007
1-2-3-GNN	0.062 ±0.001
3-IGN	0.046 ±0.001
δ-2-LGNN	0.029 ±0.001
Dimenet	0.019 ±0.001
PF-GNN	**0.017** ±0.001

OGB-molhiv

Method	ROC-AUC
GIN	75.58±1.40
GCN	76.06±0.97
DeeperGCN	78.58±1.17
PNA	79.05±1.32
DGN	79.70±0.97
GSN	77.99±1.00
Directional GSN	**80.39**±0.90
Graphormer	80.51±0.50
PF-GNN	80.15±0.68

Molecular graphs



Takeaway

We use **Particle filters** to approximate the search-tree of IR based solvers.

With **PF-GNN**, we get

- Principled approximation of universal representations on graphs
- No exponential runtime
- No preprocessing required

Thank you!