Graph Neural Networks are Inherently Good Generalizers: Insights by Bridging GNNs and MLPs

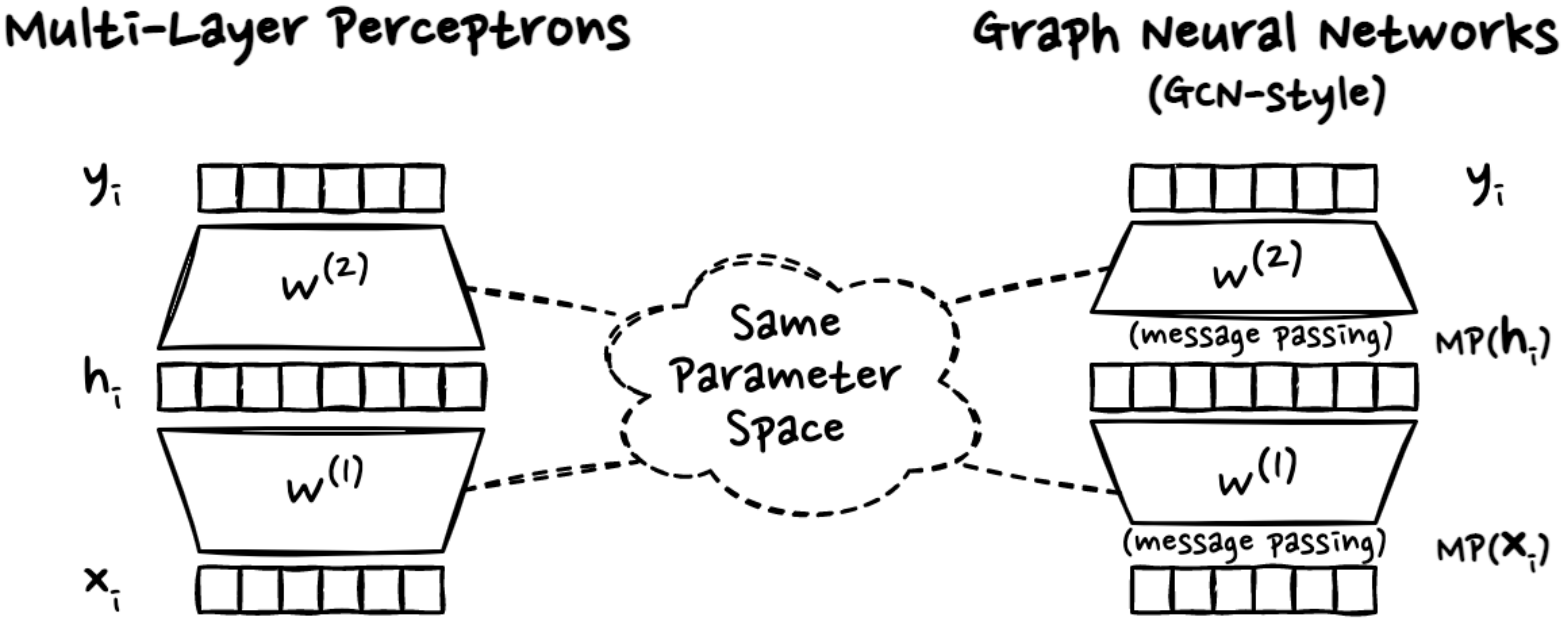
Chenxiao Yang, Qitian Wu, Jiahua Wang, Junchi Yan

Department of Computer Science and Engineering, Shanghai Jiao Tong University

MLP v.s. GNN : Architectural Connection

GNN Layer : **(MP):** $\tilde{\mathbf{h}}_u^{(l-1)} = \sum_{v \in \mathcal{N}_u \cup \{u\}} a_G(u, v) \cdot \mathbf{h}_u^{(l-1)},$ **(FF):** $\mathbf{h}_u^{(l)} = \psi^{(l)}\left(\tilde{\mathbf{h}}_u^{(l-1)}\right)$

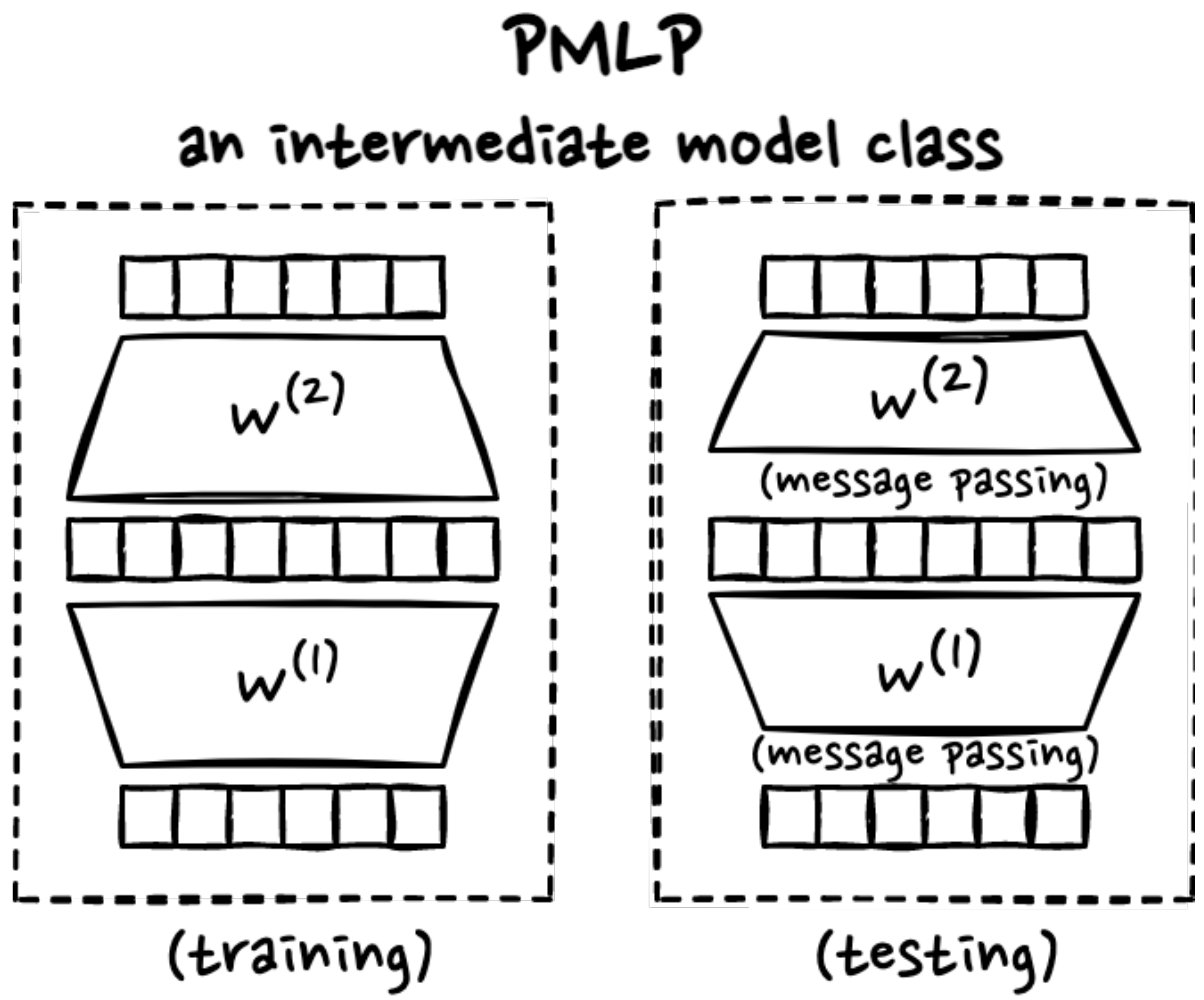
MLP Layer : ~~**(MP):**~~ $\cancel{\tilde{\mathbf{h}}_u^{(l-1)} = \sum_{v \in \mathcal{N}_u \cup \{u\}} a_G(u, v) \cdot \mathbf{h}_u^{(l-1)}}$ **(FF):** $\mathbf{h}_u^{(l)} = \psi^{(l)}\left(\mathbf{h}_u^{(l-1)}\right)$



Observation

GNN and MLP share the same feed-forward backbone and parameter (sub)space.

MLP - (?) - GNN : Propagational MLP (PMLP)



PMLP : Training with **MLP**

Inference with **GNN**

Instantiations of PMLP

Model	Train and Valid	Inference
MLP		MLP
PMLP_{GCN}		GCN: $\psi^{(l)}(\text{MP}(\{\mathbf{h}_v^{(l-1)}\}_{v\in\mathcal{N}_u\cup\{u\}}))$
PMLP_{SGC}	MLP: $\hat{y}_u = \psi(\mathbf{x}_u)$	SGC: $\psi(\text{Multi-MP}(\{\mathbf{x}_v\}_{v\in\mathcal{V}}))$
PMLP_{APPNP}		APPNP: $\text{Multi-MP}(\psi(\{\mathbf{x}_v\}_{v\in\mathcal{V}}))$
PMLP_{GCNII}	ResNet	GCNII
PMLP_{JKNet}	MLP+JK	JKNet

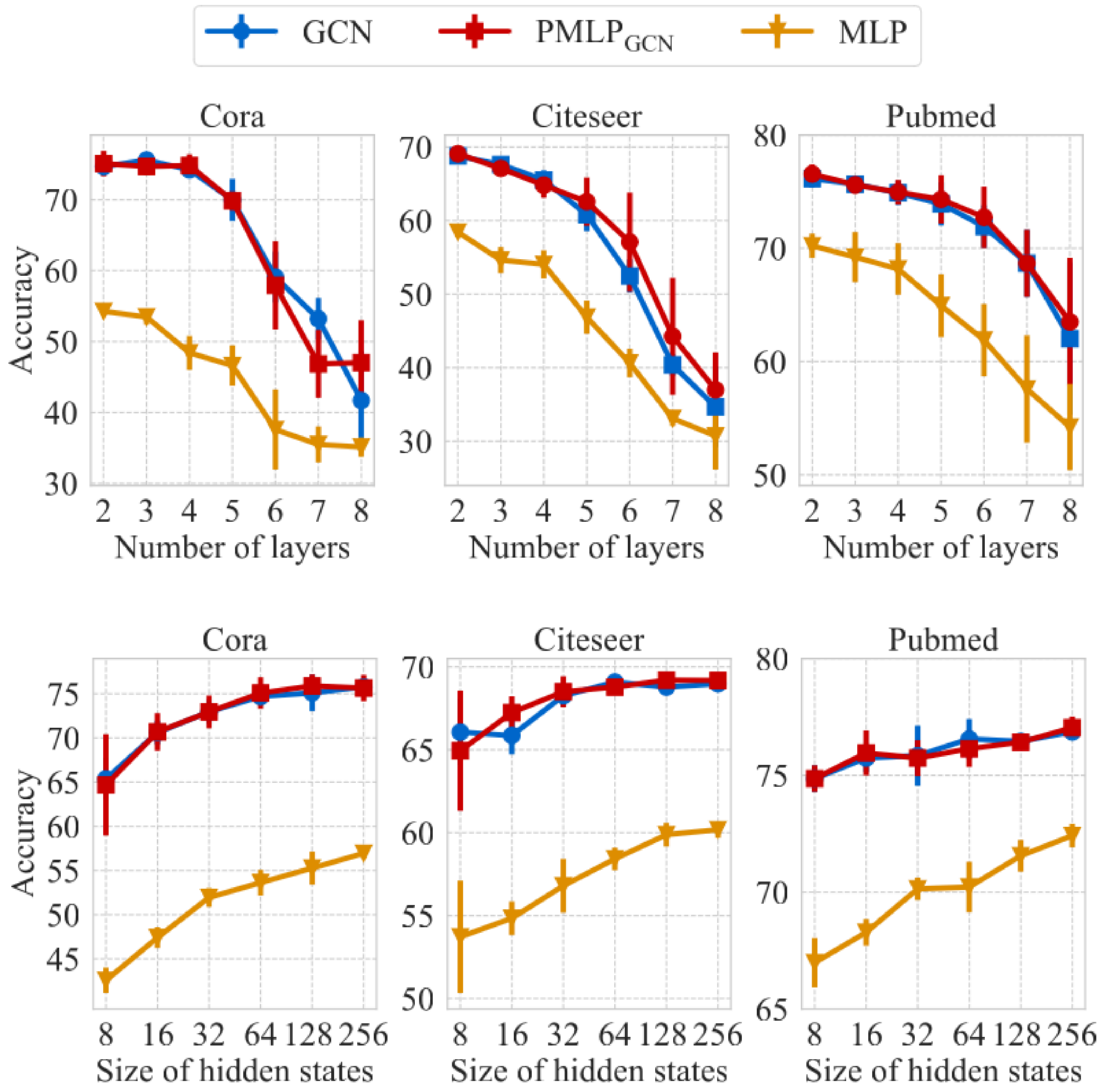
⋮

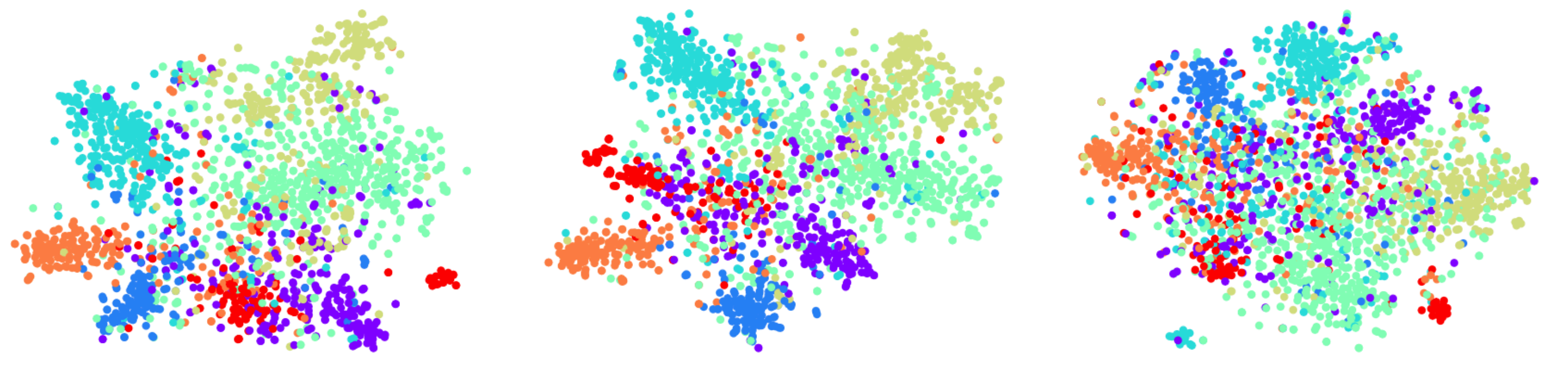
(could be extended to other GNN architectures with some modifications)

⋮

Empirical Evaluation

	Dataset #Nodes	Cora 2,708	Citeseer 3,327	Pubmed 19,717	A-Photo 7,650	A-Computer 13,752	Coauthor-CS 18,333	Coauthor-Physics 34,493
GNNs	GCN	74.82 ± 1.09	67.60 ± 0.96	76.56 ± 0.85	89.69 ± 0.87	78.79 ± 1.62	91.79 ± 0.35	91.22 ± 0.18
	SGC	73.96 ± 0.59	67.34 ± 0.54	76.00 ± 0.59	83.42 ± 2.47	77.10 ± 2.54	91.24 ± 0.59	89.18 ± 0.46
	APPNP	75.02 ± 2.17	66.58 ± 0.77	76.48 ± 0.49	89.51 ± 0.86	78.29 ± 0.55	91.64 ± 0.34	91.80 ± 0.77
MLPs	MLP	55.30 ± 0.58	56.20 ± 1.27	70.76 ± 0.78	75.61 ± 0.63	63.07 ± 1.67	87.51 ± 0.51	85.09 ± 4.11
	PMLP$_{GCN}$	75.86 ± 0.93	68.00 ± 0.70	76.06 ± 0.55	89.10 ± 0.88	78.05 ± 1.21	91.76 ± 0.27	91.35 ± 0.82
	Δ_{GNN}	+1.39%	+0.59%	−0.65%	−0.66%	−0.94%	−0.03%	+0.14%
	Δ_{MLP}	+37.18%	+21.00%	+7.49%	+17.84%	+23.75%	+4.86%	+7.36%
	PMLP$_{SGC}$	75.04 ± 0.95	67.66 ± 0.64	76.02 ± 0.57	86.50 ± 1.40	74.72 ± 3.86	91.09 ± 0.50	89.34 ± 1.40
	Δ_{GNN}	+1.46%	+0.48%	+0.03%	+3.69%	−3.09%	−0.16%	+0.18%
	Δ_{MLP}	+35.70%	+20.39%	+7.43%	+14.40%	+18.47%	+4.09%	+4.99%
	PMLP$_{APP}$	75.84 ± 1.36	67.52 ± 0.82	76.30 ± 1.44	88.47 ± 1.64	78.07 ± 2.10	91.64 ± 0.46	91.96 ± 0.51
	Δ_{GNN}	+1.09%	+1.41%	−0.24%	−1.16%	−0.28%	+0.00%	+0.17%
	Δ_{MLP}	+37.14%	+20.14%	+7.83%	+17.01%	+23.78%	+4.72%	+8.07%

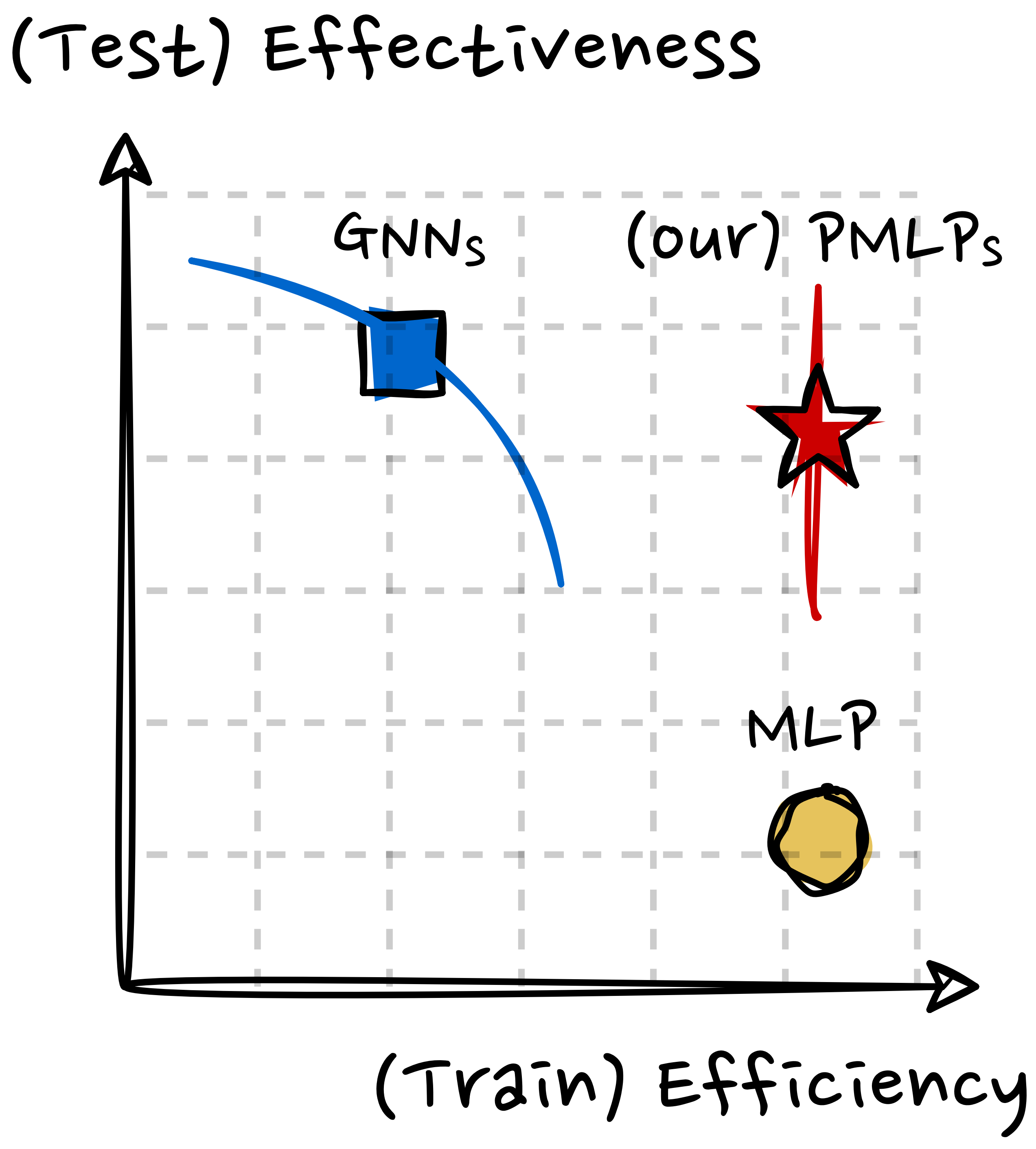




(a) GCN (b) PMLP$_{GCN}$ (c) MLP

1. PMLPs **significantly outperform** MLP.
2. PMLPs **perform on par with** or exceed GNNs in inductive learning setting.
3. PMLPs **speed up training** up to 65 times.
4. PMLPs are more **robust to graph structural noises**.

GNNs are Inherently Good Generalizers



What lies between MLP and GNN?

MLP $\xrightarrow{\text{large gap}}$ PMLP :

- Same optimization process and learning dynamics
- Same model expressivity and capability to fit the data

+ Different generalizability due to the GNN architecture in inference (key factor)

PMLP $\xrightarrow{\text{small gap}}$ GNN :

- Same GNN architecture in inference

+ Different optimization process and model expressivity

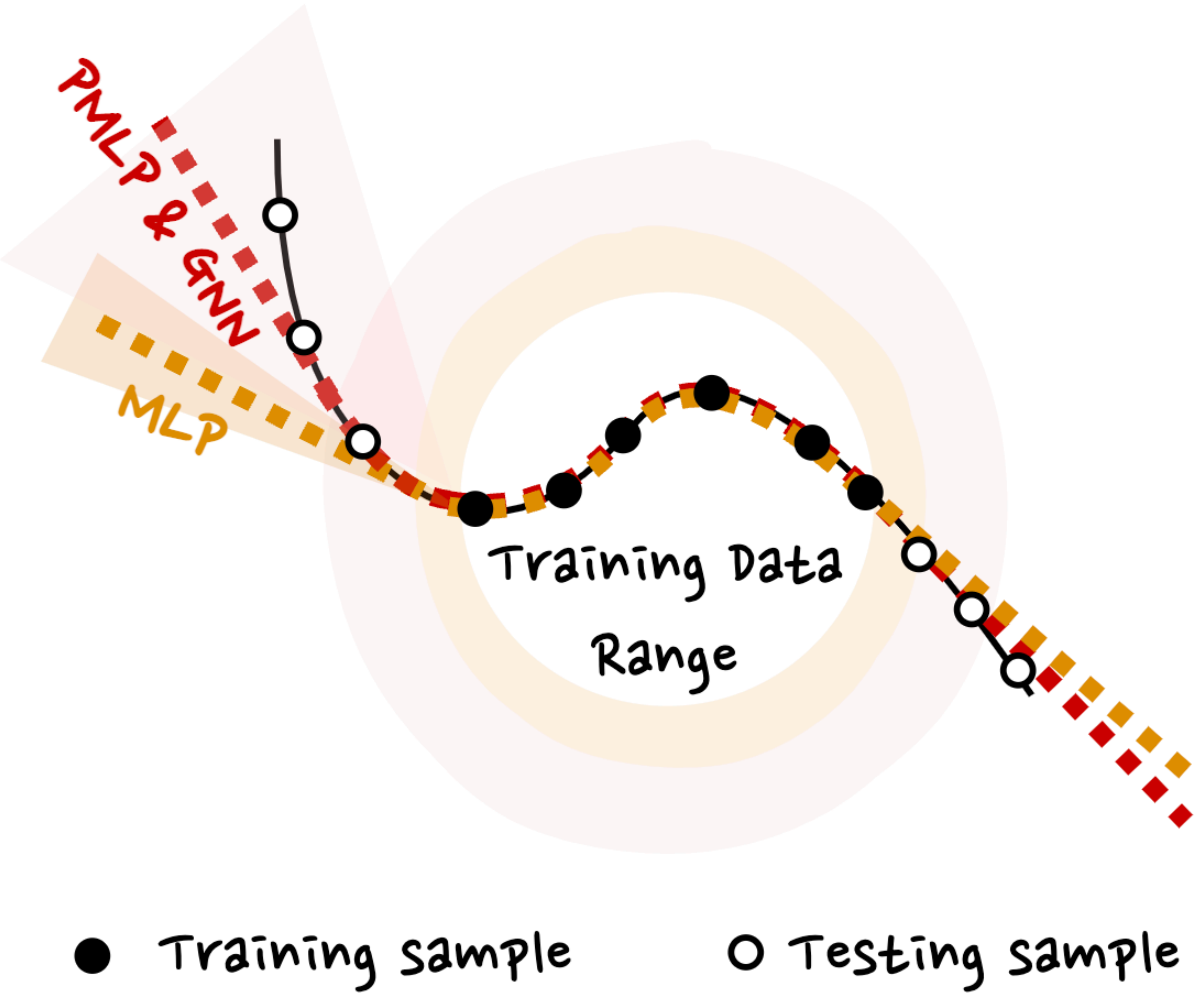
PMLPs pinpoints the major source of performance gap between GNNs and MLPs in node classification stems from **the inherent generalization capability of GNN architectures**.

Effects of Model Architectures in Extrapolation

Comparison in NTK perspective

The key factor

$$f_{mlp}(\mathbf{x}) = \mathbf{w}_{mlp}^{*\top} \phi_{mlp}(\mathbf{x}) \qquad f_{pmlp}(\mathbf{x}) = \mathbf{w}_{mlp}^{*\top} \phi_{gnn}(\mathbf{x}) \qquad f_{gnn}(\mathbf{x}) = \mathbf{w}_{gnn}^{*\top} \phi_{gnn}(\mathbf{x})$$



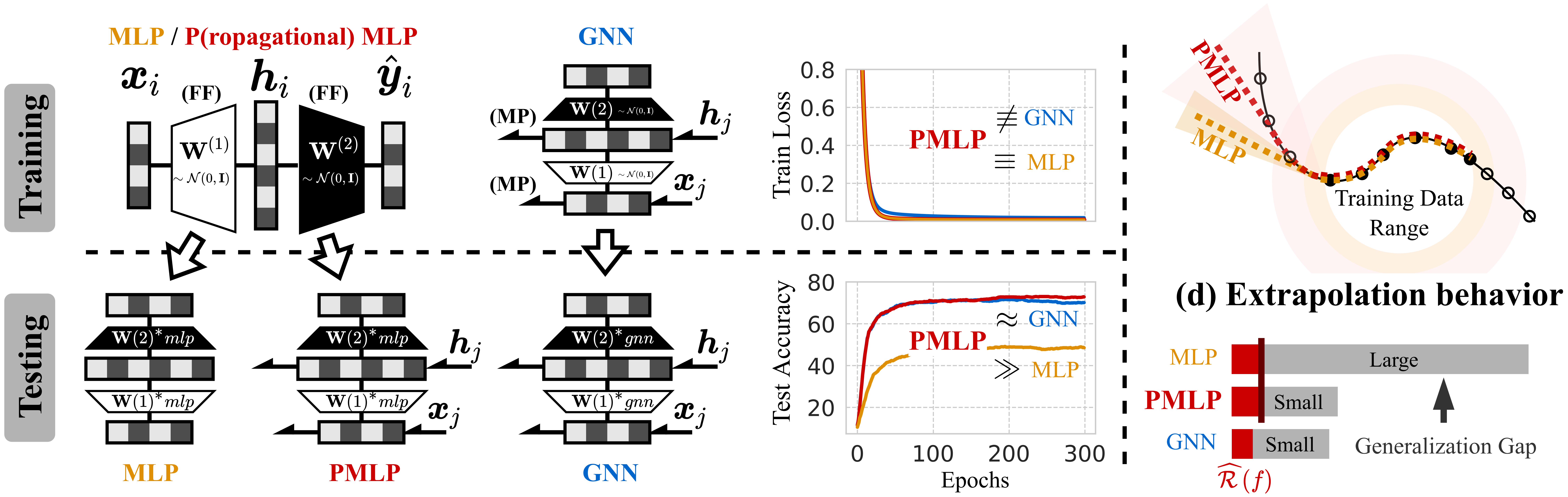
Theorem 4

Alike MLP, both PMLP and GNN (in GCN-style) eventually converge to linear functions when testing samples are far away from the training data support.

Theorem 5

PMLP and GNN's convergence rates (to linear models) are smaller than MLP due to message passing at each layer. This indicates they are less more vulnerable to linearalization, and prone to generalize to testing samples near the training data.

Summary and Takeaways



(a) Illustration of model architecture **(b) Learning behavior** **(c) Intrinsic generalizability of GNN**

1. Vanilla MLP with post-training message passing can be as effective as GNN.
2. GNN inherently extrapolates better than MLP because of its architecture.
3. It is important to understand GNN generalization for understanding its success.