Boost then Convolve: Gradient Boosting Meets Graph Neural Networks

Sergey Ivanov
Research Scientist, Criteo

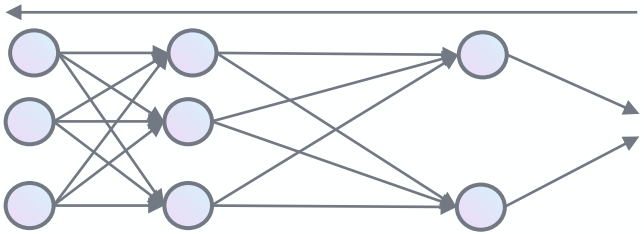
Outline

- Background
  - GNNs
  - Gradient Boosted Decision Trees
- Boost-GNN
  - Architecture
  - Training/Inference
- Experiments

Two worlds of ML models

Neural Networks

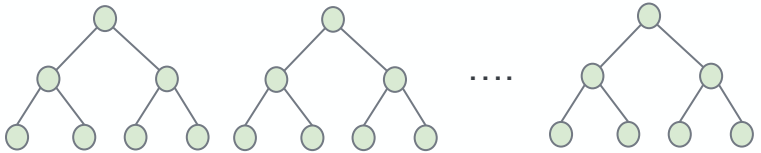
SGD



- Automated feature engineering
- Transfer learning
- End-to-end training

Gradient Boosted Decision Trees

FGD



- Built-in input preprocessing
- Efficient for heterogeneous data
- Outcome interpretation

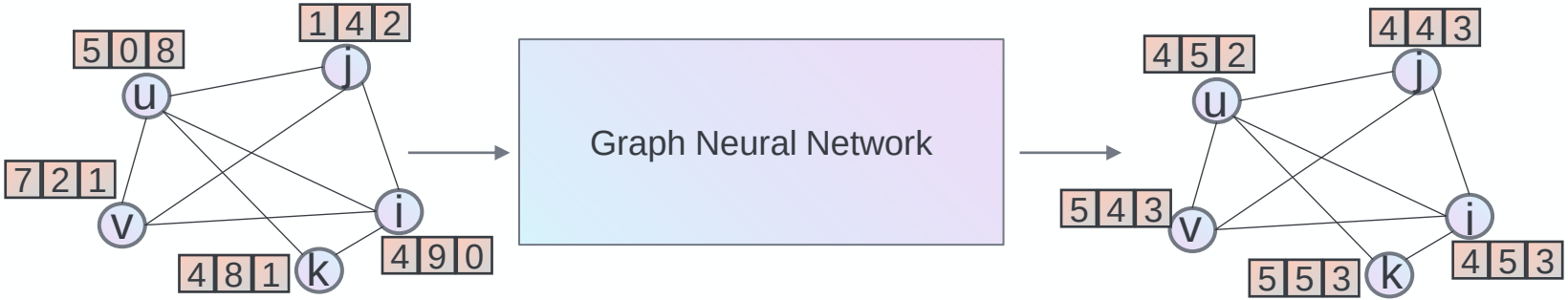
Is it possible to get the best of **both** worlds?

Graph Neural Networks

criteo

Graph Neural Network

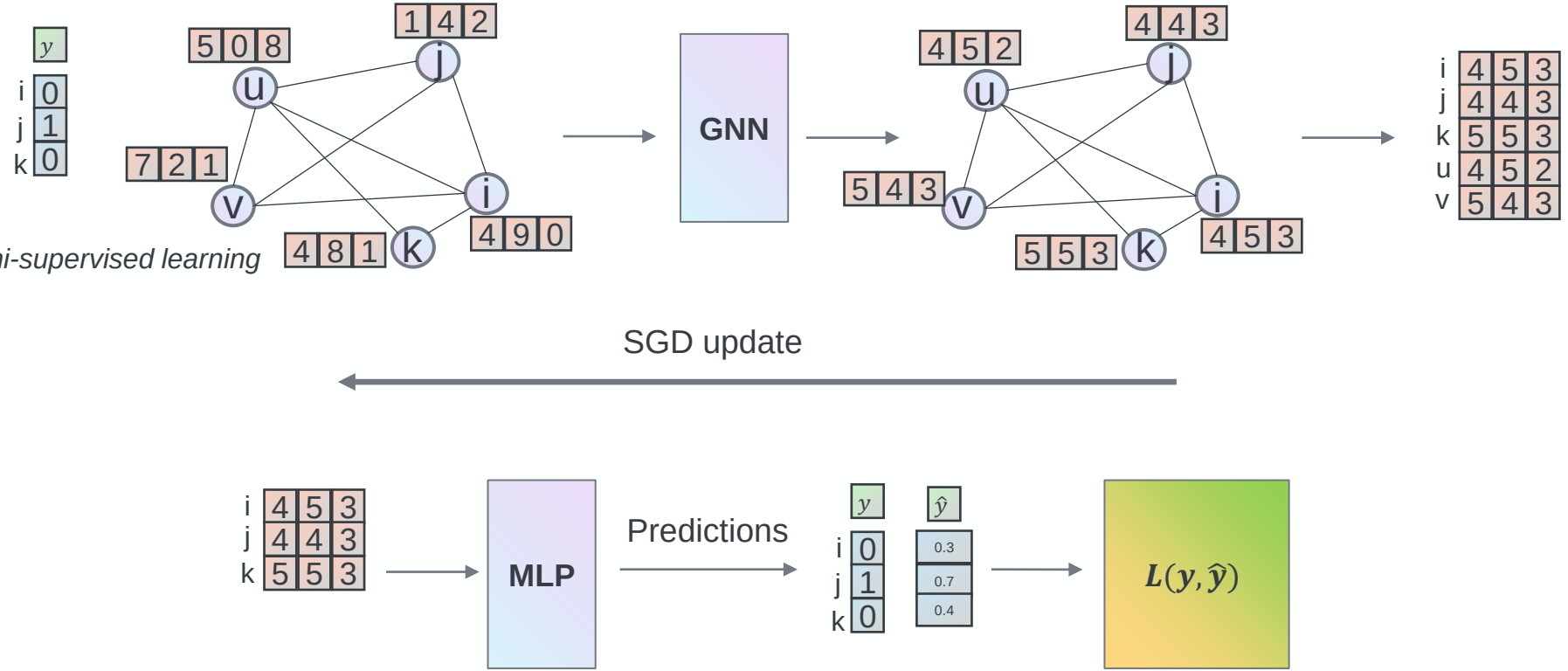
GNN aggregates neighborhood features and applies some (non-linear) function on top.



$$\mathbf{x}_v^t = \text{COMBINE}^t\left(\mathbf{x}_v^{t-1}, \text{AGGREGATE}^t\left(\left\{(\mathbf{x}_w^{t-1}, \mathbf{x}_v^{t-1}) : (w, v) \in E\right\}\right)\right)$$

Graph Neural Network training

GNN is differentiable parametric function => SGD

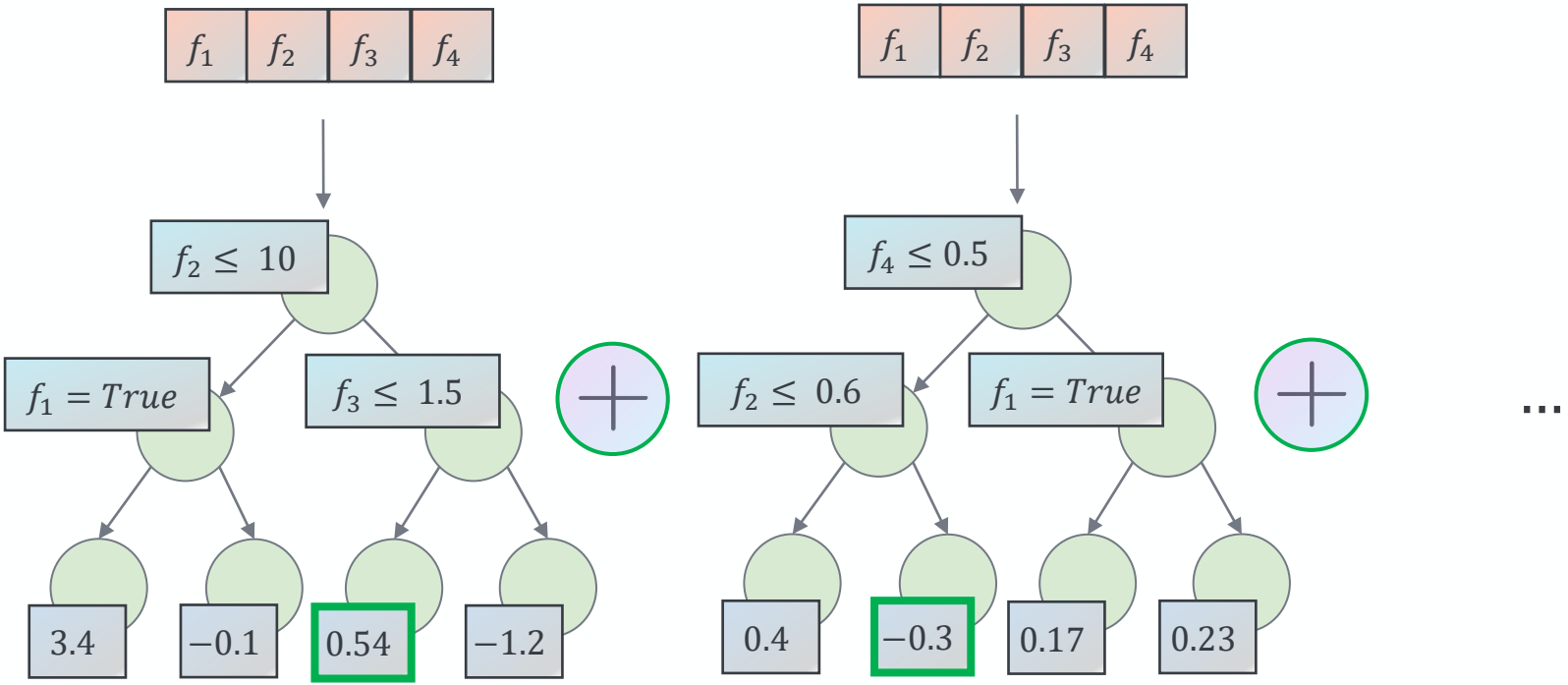


Gradient Boosted Decision Trees

criteo

Gradient Boosted Decision Trees

GBDT maps features to predictions.



$$F_m(x) = F_{m-1}(x) + \arg\min_{h_m \in \mathcal{H}} \left[\sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + h_m(x_i)) \right]$$

Decision Tree training

How a single DT makes prediction?

f_1	f_2	f_3	y
4	7	3	0
4	8	3	1
5	9	3	0
4	7	2	1
5	8	3	0

$$\hat{y} = \frac{\sum y_i}{N}$$

$$H = \frac{\sum (y_i - \hat{y})^2}{N}$$

$$L(f_i, s_j) = \frac{n_{left}}{N_m} H(T_{left}) + \frac{n_{right}}{N_m} H(T_{right})$$

$(f_1, 4)$
$(f_1, 5)$
$(f_2, 7)$
$(f_2, 8)$
$(f_2, 9)$
$(f_3, 2)$
$(f_3, 3)$

$f_1 \leq 4$

T_{left}

4	7	3	0
4	8	3	1
4	7	2	1

$$\hat{y} = \frac{\sum y_i}{N} = \frac{2}{3}$$

$$H(T_{left}) = \frac{2}{9}$$

T_{right}

5	9	3	0
5	8	3	0

$$\hat{y} = \frac{\sum y_i}{N} = \frac{0}{2} = 0$$

$$H(T_{right}) = 0$$

$$L(f_4, 4) = \frac{3}{5} * \frac{2}{9} + \frac{2}{5} * 0 = \frac{2}{15}$$

$$(f_1, 4) \rightarrow \frac{2}{15}$$

Decision Tree training

How a single DT makes prediction?

f_1	f_2	f_3
4	7	3
4	8	3
5	9	3
4	7	2
5	8	3

y
0
1
0
1
0

$$\hat{y} = \frac{\sum y_i}{N}$$

$$H = \frac{\sum (y_i - \hat{y})^2}{N}$$

$$L(f_i, s_j) = \frac{n_{left}}{N_m} H(T_{left}) + \frac{n_{right}}{N_m} H(T_{right})$$

$(f_1, 4)$
$(f_1, 5)$
$(f_2, 7)$
$(f_2, 8)$
$(f_2, 9)$
$(f_3, 2)$
$(f_3, 3)$

$f_1 \leq 4$

$f_1 \leq 5$

$f_2 \leq 7$

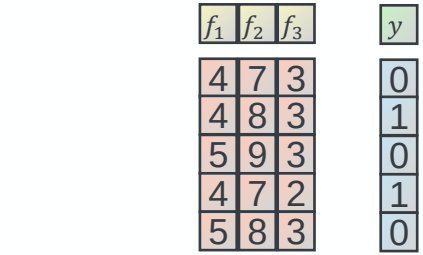
...

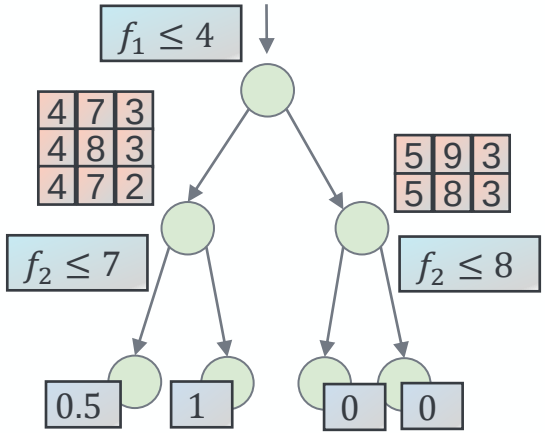
$$(f_1, 4) \rightarrow \frac{2}{15}$$

$$(f_1, 5) \rightarrow \frac{6}{25}$$

$$(f_2, 7) \rightarrow \frac{17}{36}$$

$\dots$



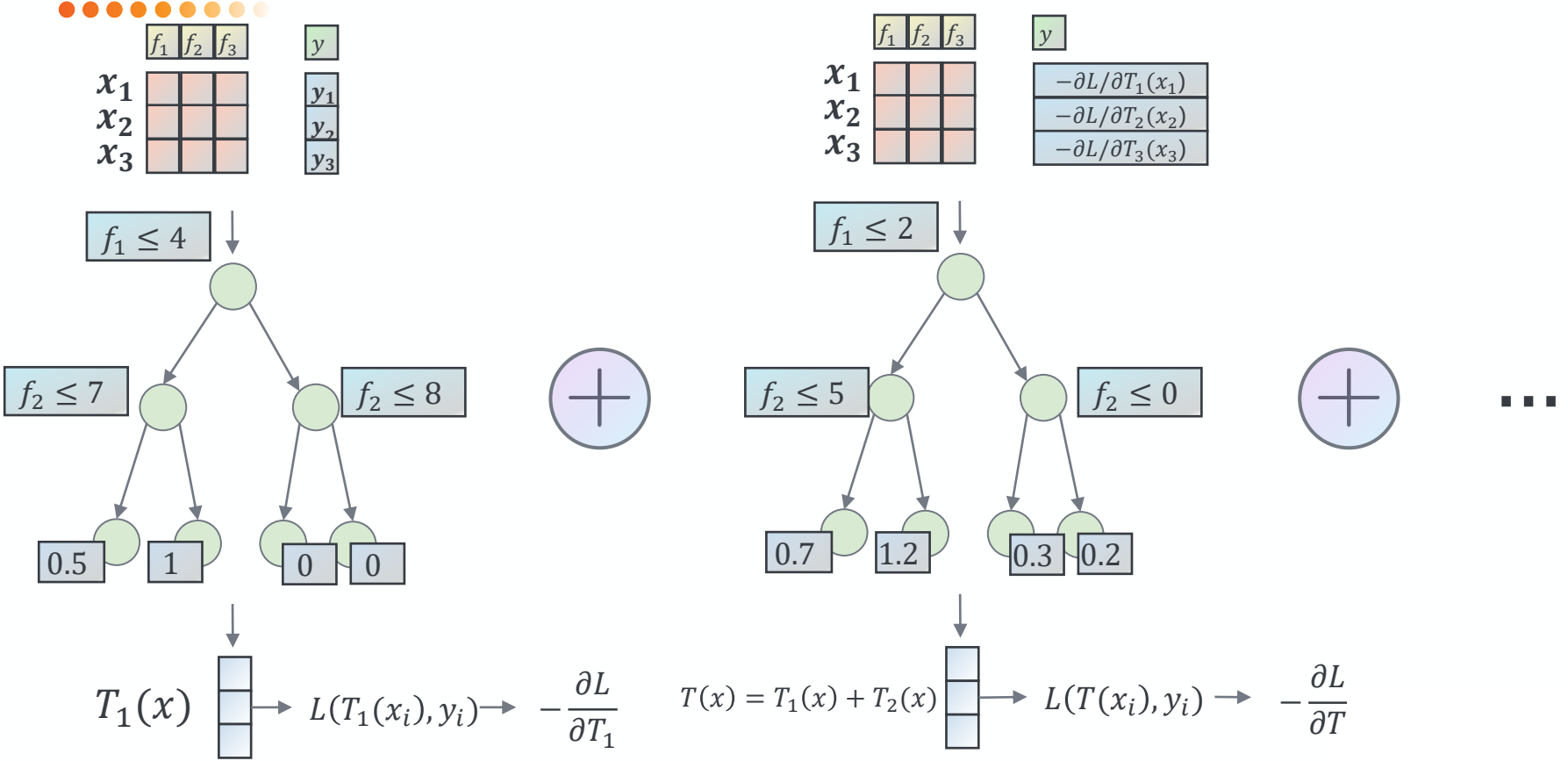


Leaves take the average target labels of all observations that end up in that leaf.

GBDT training

How do we combine trees?

We build each new tree approximating the error of the previous trees.



Note on gradient descent

Gradient Descent in space of parameters (GNN)

$$f_\theta : (G, X) \mapsto (G, X') \quad \Rightarrow \quad \theta_{new} \leftarrow \theta_{old} - \eta \frac{\partial L}{\partial \theta_{old}} \quad \Rightarrow \quad L_{new} < L_{old}$$

Gradient Descent in space of functions (GBDT)

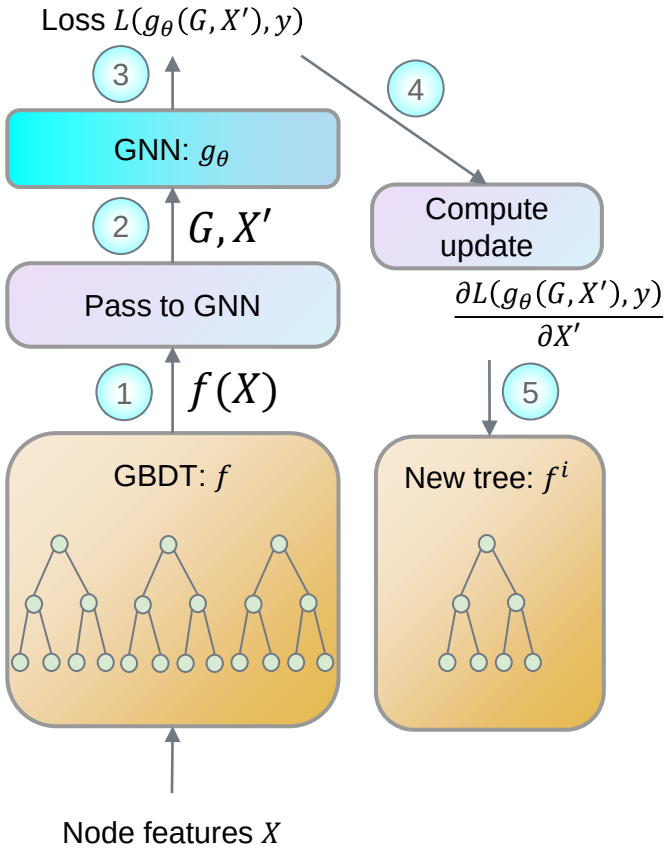
$$f = \sum f_i \quad \Rightarrow f_0 = c, \qquad f_{i+1} \leftarrow \sum_{j=0}^{i} f_j - \eta \frac{\partial L}{\partial \sum_{j=0}^{i} f_j} \quad \Rightarrow \quad L_{new} < L_{old}$$

Boost-GNN

criteo

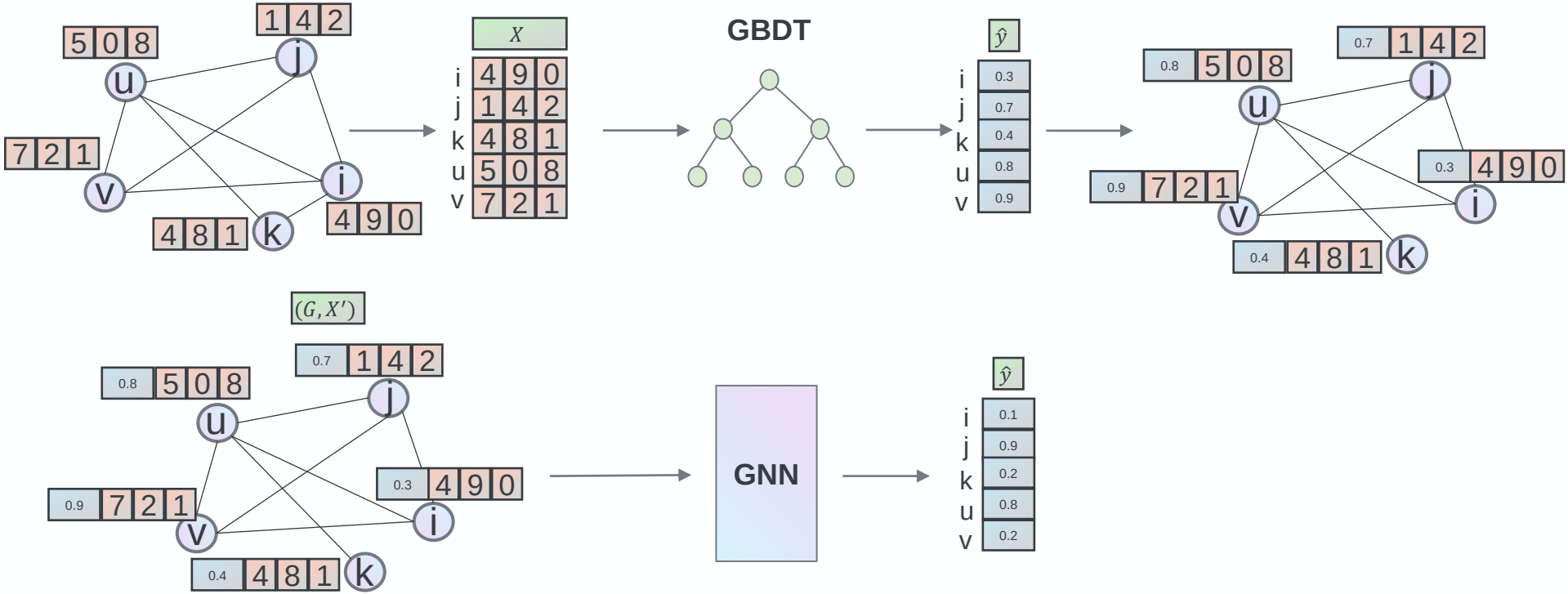
Architecture

- Data preprocessing inherited by GBDT model (missing values, cat features, etc.)
- Pluggable with any GBDT or GNN model
- End-to-end training
- Interpretation 1: GNN with embedding layer
- Interpretation 2: GBDT with parametric loss function

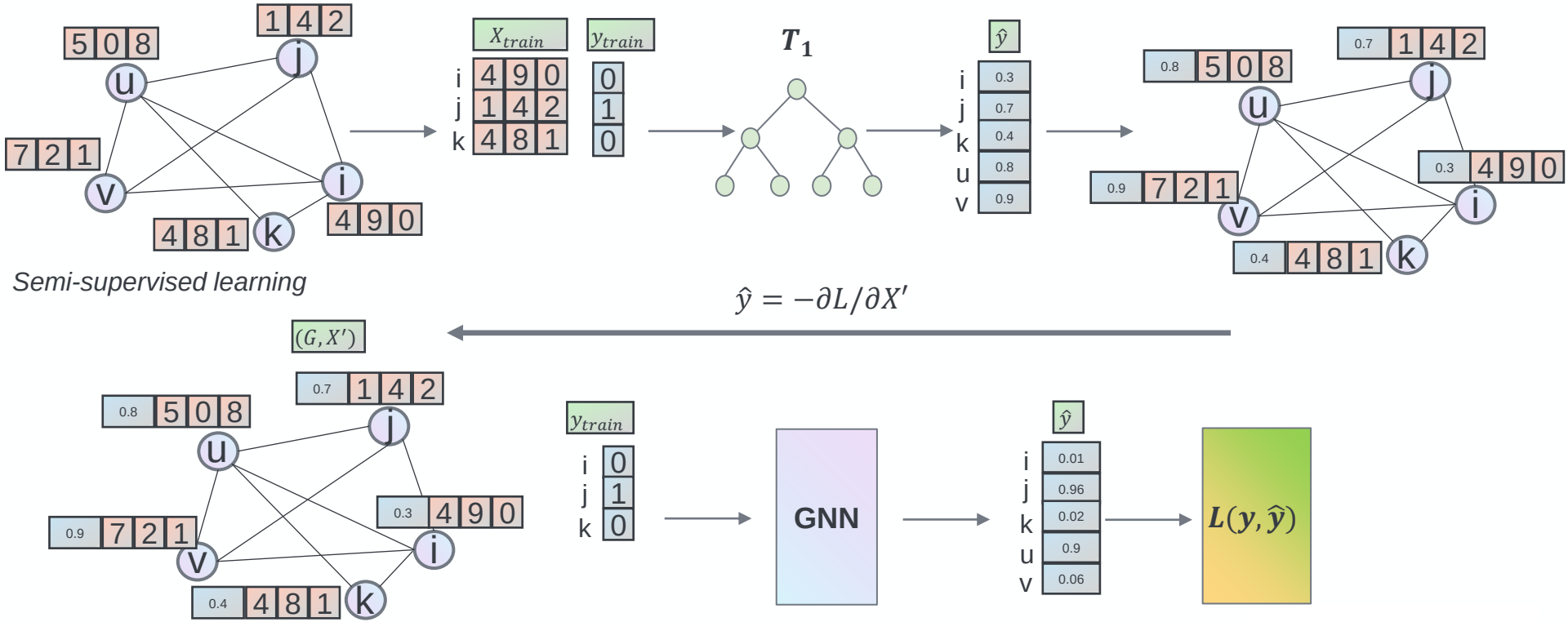


BGNN inference

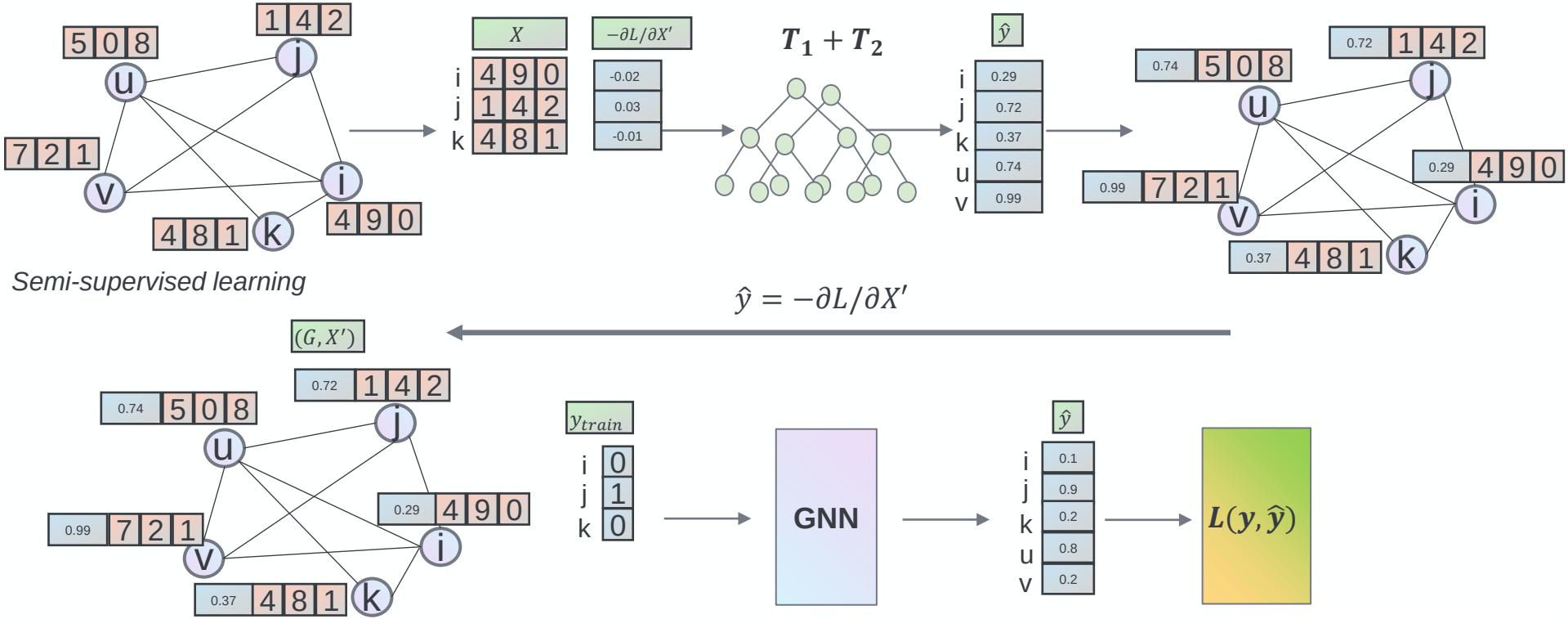
First, boost features with GBDT, then apply GNN on new features.



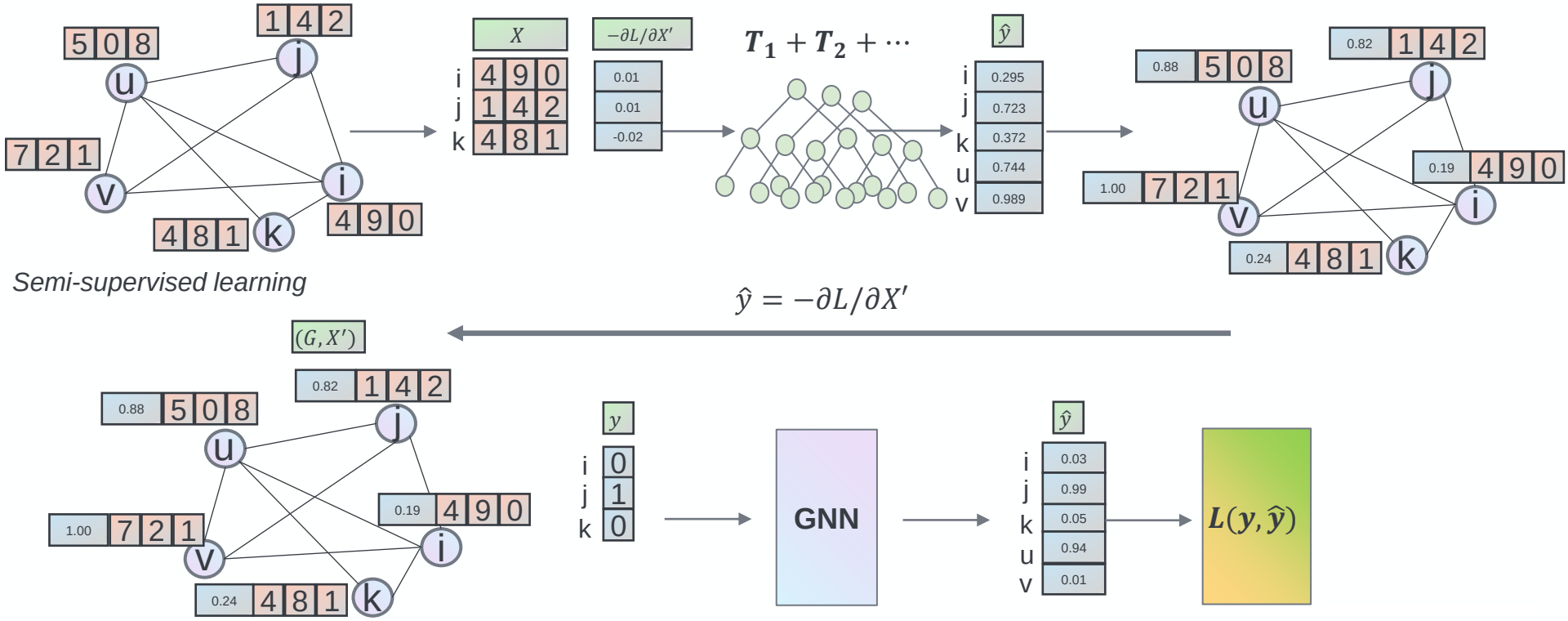
BGNN training (1st tree)



BGNN training (2nd tree)



BGNN training



Experiments

Experiments

Q1: Does combination of GBDT with GNN helps on node prediction tasks?

Q2: Is our BGNN architecture (end-to-end) better than just stacking GBDT with GNN?

		Heterogeneous								*Homogeneous*	
	Method	House RMSE	House Gap %	County RMSE	County Gap %	VK RMSE	VK Gap %	Avazu RMSE	Avazu Gap %	Wiki RMSE	Wiki Gap %
GBDT	**LightGBM**	0.63 ± 0.01	15.3	1.39 ± 0.07	-4.32	7.16 ± 0.2	-0.82	0.1172 ± 0.02	3.36	46359 ± 4508	0.97
	CatBoost	0.63 ± 0.01	15.98	1.4 ± 0.07	-3.93	7.2 ± 0.21	-0.33	0.1171 ± 0.02	3.27	49915 ± 3643	8.71
GNN	**GAT**	0.54 ± 0.01	0	1.45 ± 0.06	0	7.22 ± 0.19	0	0.1134 ± 0.01	0	**45916 ± 4527**	**0**
	GCN	0.63 ± 0.01	16.77	1.48 ± 0.08	2.06	7.25 ± 0.19	0.34	0.1141 ± 0.02	0.58	**44936 ± 4083**	**-2.14**
	AGNN	0.59 ± 0.01	8.01	1.45 ± 0.08	-0.19	7.26 ± 0.2	0.54	0.1134 ± 0.02	-0.02	45982 ± 3058	0.14
	APPNP	0.69 ± 0.01	27.11	1.5 ± 0.11	3.39	13.23 ± 0.12	83.19	0.1127 ± 0.01	-0.65	53426 ± 4159	16.36
NN	**FCNN**	0.68 ± 0.02	25.49	1.48 ± 0.07	1.56	7.29 ± 0.21	1.02	0.118 ± 0.02	4.07	51662 ± 2983	12.51
	FCNN-GNN	0.53 ± 0.01	-2.48	1.39 ± 0.06	-4.68	7.22 ± 0.2	0.01	0.1114 ± 0.02	-1.82	48491 ± 7889	5.61
Ours	**Res-GNN**	**0.51 ± 0.01**	**-6.39**	**1.33 ± 0.08**	**-8.35**	**7.07 ± 0.2**	**-2.04**	**0.1095 ± 0.01**	**-3.42**	46747 ± 4639	1.81
	BGNN	**0.5 ± 0.01**	**-8.15**	**1.26 ± 0.08**	**-13.67**	**6.95 ± 0.21**	**-3.8**	**0.109 ± 0.01**	**-3.9**	49222 ± 3743	7.2

Experiments

Q3: Can BGNN architecture be beneficial to different GNN architectures?

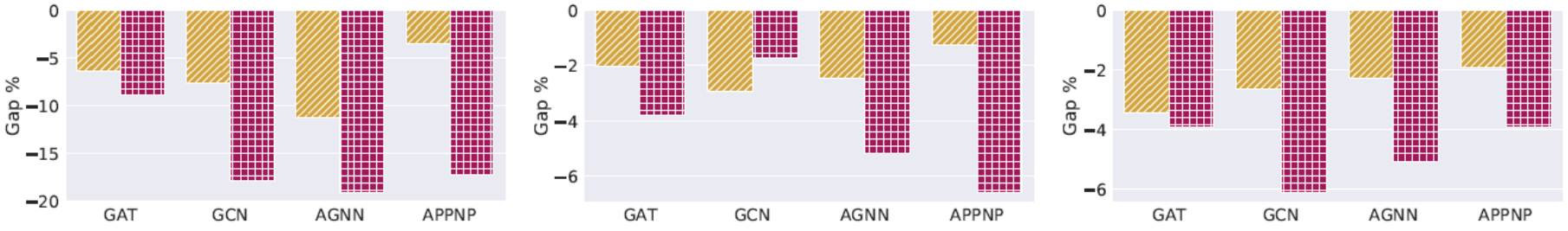


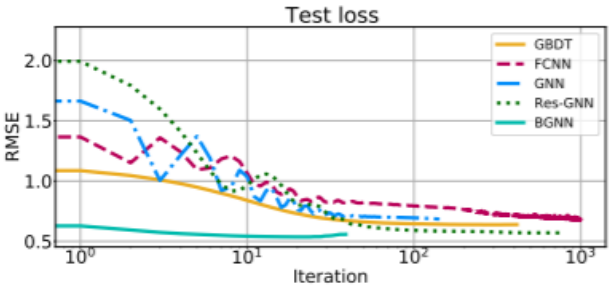
Figure 2: Relative difference for **Res-GNN** (yellow, diagonal) and **BGNN** (red, squared) for different GNN architectures w.r.t. GNN RMSE. The smaller, the better.

Experiments

Q4: Is running time an issue for BGNN?

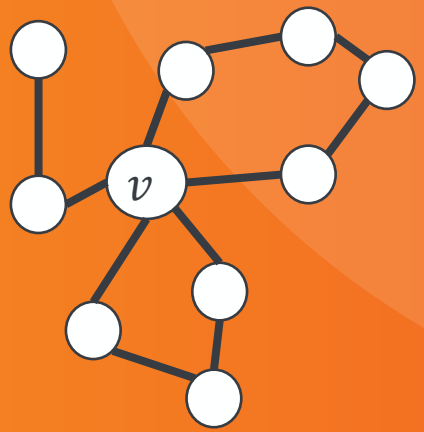
Summary of running time (s) in node regression task.

	Method	House	County	VK	Wiki	Avazu
GBDT	**LightGBM**	4 ± 1	2 ± 1	24 ± 4	10 ± 1	2 ± 2
	CatBoost	3 ± 0	1 ± 0	5 ± 3	3 ± 2	0 ± 0
GNN	**GAT**	35 ± 2	19 ± 6	42 ± 4	15 ± 1	9 ± 2
	GCN	28 ± 0	18 ± 7	38 ± 0	13 ± 3	12 ± 6
	AGNN	38 ± 5	28 ± 3	48 ± 3	19 ± 5	14 ± 8
	APPNP	68 ± 1	34 ± 10	81 ± 3	49 ± 26	24 ± 15
NN	**FCNN**	16 ± 5	2 ± 1	109 ± 35	12 ± 2	2 ± 0
	FCNN-GNN	39 ± 1	21 ± 6	48 ± 2	16 ± 1	14 ± 3
Ours	**Res-GNN**	36 ± 7	7 ± 3	41 ± 7	31 ± 9	7 ± 2
	BGNN	20 ± 4	2 ± 0	16 ± 0	21 ± 7	5 ± 1

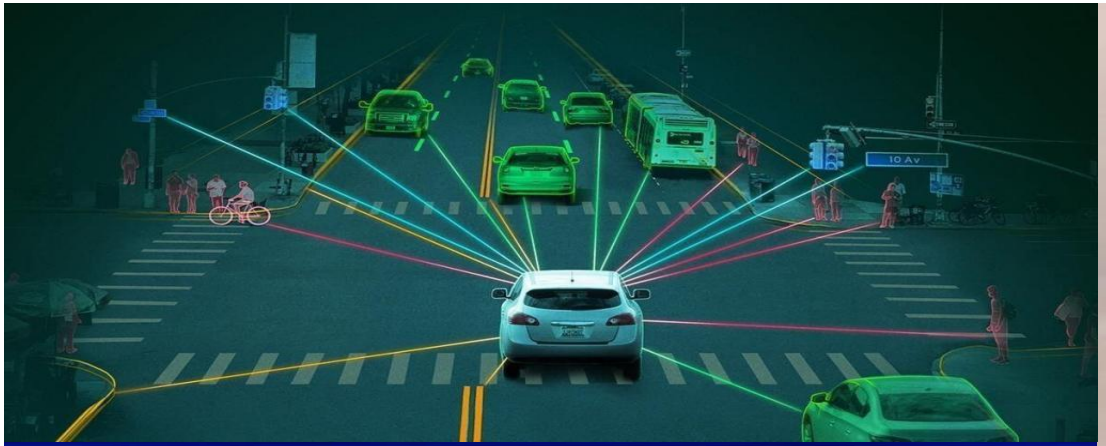


(a) House

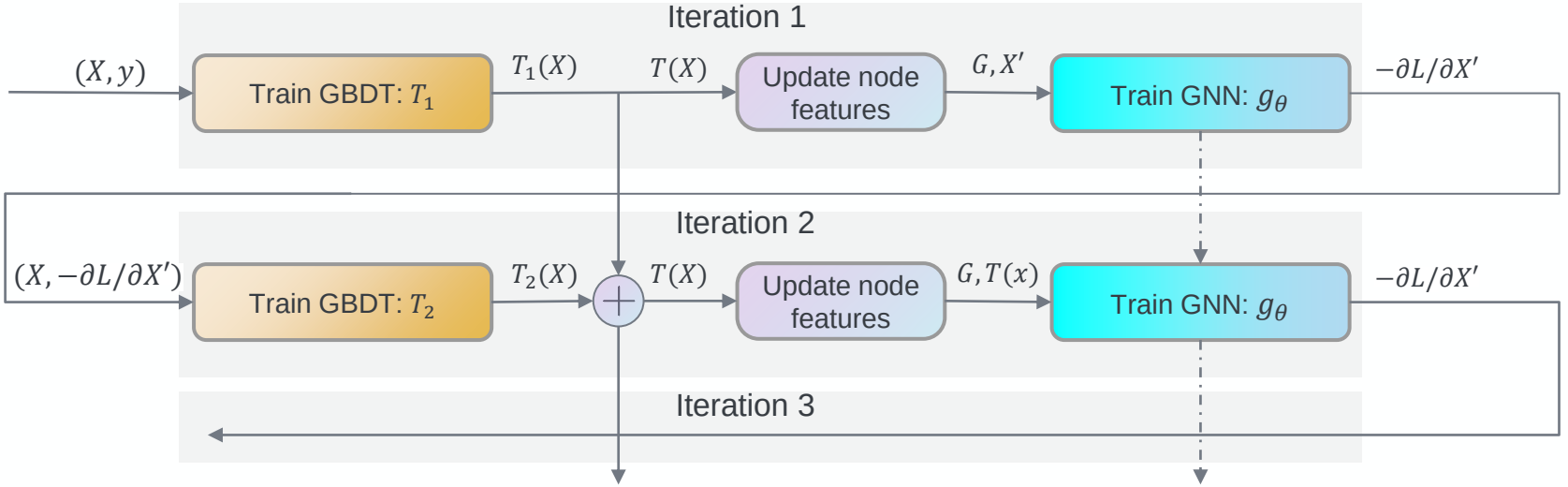
Thank you!



Graphs and ML



Boosted Graph Neural Network training



- First tree approximates original target
- Each new tree approximates a gradient update of the loss function $-\partial L_{gnn}/\partial T(x)$

Datasets

Node regression

	House	County	VK	Wiki	Avazu
Setting	Balanced	Balanced	Balanced	Imbalanced	Imbalanced
# Nodes	20640	3217	54028	5201	1297
# Edges	182146	12684	213644	198493	54364
# Features/Node	6	7	14	3148	9
Mean Target	2.06	5.44	35.47	27923.86	0.08
Min Target	0.14	1.7	13.48	16	0
Max Target	5.00	24.1	118.39	849131	1
Median Target	1.79	5	33.83	9225	0

Node classification

	SLAP	DBLP	OGB-ArXiv
# Nodes	20419	14475	169343
# Edges	172248	40269	1166243
# Features	2701	5002	128
Classes	15	4	40
Min Class	103	745	29
Max Class	534	1197	27321

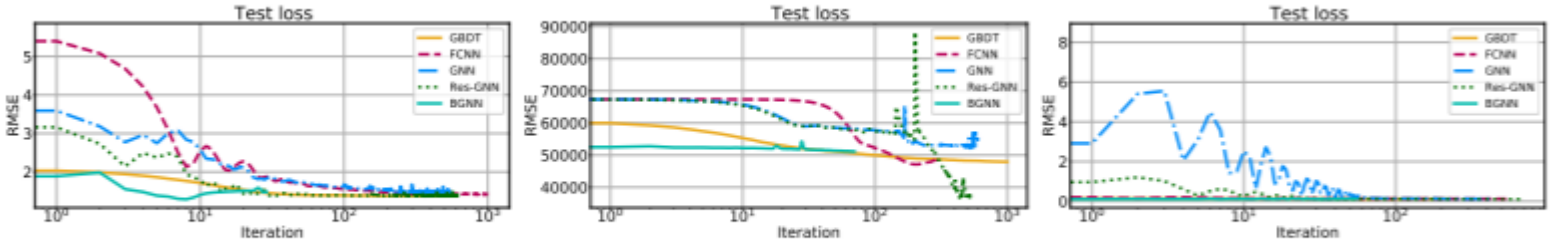
Node classification

		Heterogeneous								*Homogeneous*	
	Method	House_class RMSE	House_class Gap %	VK_class RMSE	VK_class Gap %	Slap RMSE	Slap Gap %	DBLP RMSE	DBLP Gap %	OGB-ArXiv RMSE	OGB-ArXiv Gap %
GBDT	**LightGBM**	0.52 ± 0.01	-16.82	0.57 ± 0.01	-1.26	0.922 ± 0.01	15.12	0.759 ± 0.03	-5.42	0.45	-36.35
	CatBoost	0.55 ± 0	-11.98	0.579 ± 0.01	0.26	**0.963 ± 0**	**20.3**	**0.913 ± 0.01**	**13.73**	0.51	-26.97
GNN	**GAT**	0.625 ± 0	0	0.577 ± 0	0	0.801 ± 0.01	0	0.802 ± 0.01	0	**0.7**	**0**
	GCN	0.6 ± 0	-3.98	0.574 ± 0	-0.6	0.878 ± 0.01	9.72	0.428 ± 0.04	-46.6	-	-
	AGNN	0.614 ± 0.01	-1.73	0.572 ± 0	-0.79	0.892 ± 0.01	11.47	0.794 ± 0.01	-1.02	-	-
	APPNP	0.619 ± 0	-0.89	0.573 ± 0	-0.67	0.895 ± 0.01	11.79	0.83 ± 0.02	3.47	-	-
NN	**FCNN**	0.534 ± 0.01	-14.53	0.567 ± 0.01	-1.72	0.759 ± 0.04	-5.24	0.623 ± 0.02	-22.3	0.5	-28.91
	FCNN-GNN	**0.64 ± 0**	**2.36**	0.589 ± 0	2.13	0.89 ± 0.01	11.11	0.81 ± 0.01	0.94	**0.71**	**0.54**
Ours	**Res-GNN**	0.625 ± 0.01	-0.06	**0.603 ± 0**	**4.45**	0.905 ± 0.01	13.06	**0.892 ± 0.01**	**11.11**	0.7	-0.33
	BGNN	**0.682 ± 0**	**9.18**	**0.683 ± 0**	**18.3**	**0.95 ± 0**	**18.61**	0.889 ± 0.01	10.77	0.67	-4.36

Improvement for different GNN models

	Method	House RMSE	House Time (s)	County RMSE	County Time (s)	VK RMSE	VK Time (s)	Wiki RMSE	Wiki Time (s)	Avazu RMSE	Avazu Time (s)
GAT	GNN	0.54 ± 0.01	35 ± 2	1.45 ± 0.06	19 ± 6	7.22 ± 0.19	42 ± 4	45916 ± 4527	15 ± 1	0.11 ± 0.01	9 ± 2
	Res-GNN	0.51 ± 0.01	36 ± 7	1.33 ± 0.08	7 ± 3	7.07 ± 0.2	41 ± 7	46747 ± 4639	31 ± 9	0.11 ± 0.01	7 ± 2
	BGNN	0.5 ± 0.01	20 ± 4	1.26 ± 0.08	2 ± 0	6.95 ± 0.21	16 ± 0	49222 ± 3743	21 ± 7	0.11 ± 0.01	5 ± 1
GCN	GNN	0.63 ± 0.01	28 ± 0	1.48 ± 0.08	18 ± 7	7.25 ± 0.19	38 ± 0	44936 ± 4083	13 ± 3	0.11 ± 0.02	12 ± 6
	Res-GNN	0.59 ± 0.01	25 ± 2	1.35 ± 0.09	11 ± 5	7.03 ± 0.2	52 ± 6	44876 ± 3777	21 ± 5	0.11 ± 0.02	9 ± 6
	BGNN	0.54 ± 0.01	41 ± 15	1.33 ± 0.13	12 ± 8	7.12 ± 0.21	76 ± 6	47426 ± 4112	22 ± 11	0.11 ± 0.01	4 ± 1
AGNN	GNN	0.59 ± 0.01	38 ± 5	1.45 ± 0.08	28 ± 3	7.26 ± 0.2	48 ± 3	45982 ± 3058	19 ± 5	0.11 ± 0.02	14 ± 8
	Res-GNN	0.52 ± 0.01	33 ± 4	1.3 ± 0.07	16 ± 4	7.08 ± 0.2	51 ± 15	46010 ± 2355	24 ± 3	0.11 ± 0.02	7 ± 2
	BGNN	0.49 ± 0.01	34 ± 4	1.28 ± 0.08	3 ± 1	6.89 ± 0.21	25 ± 4	53080 ± 5117	47 ± 37	0.11 ± 0.02	5 ± 1
APPNP	GNN	0.69 ± 0.01	68 ± 1	1.5 ± 0.11	34 ± 10	13.23 ± 0.12	81 ± 3	53426 ± 4159	49 ± 26	0.11 ± 0.01	24 ± 15
	Res-GNN	0.67 ± 0.01	58 ± 12	1.41 ± 0.12	19 ± 10	13.06 ± 0.17	76 ± 11	53206 ± 4593	66 ± 27	0.11 ± 0.01	15 ± 10
	BGNN	0.59 ± 0.01	21 ± 7	1.33 ± 0.1	17 ± 6	12.36 ± 0.14	50 ± 6	54359 ± 4734	30 ± 13	0.11 ± 0.01	6 ± 1

Convergence of the model



(a) **County** (b) **Wiki** (c) **Avazu**