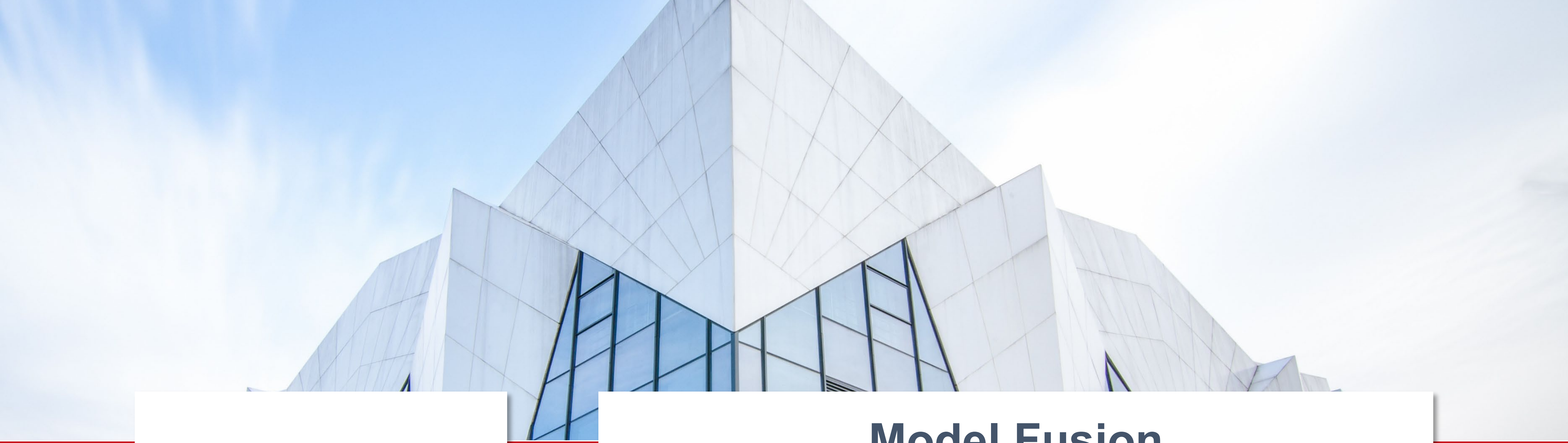
Deep Neural Network Fusion via Graph Matching with Applications to Model Ensemble and Federated Learning

Chang Liu, Chenfei Lou, Runzhong Wang, Yuhan Xi, Li Shen, Junchi Yan

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

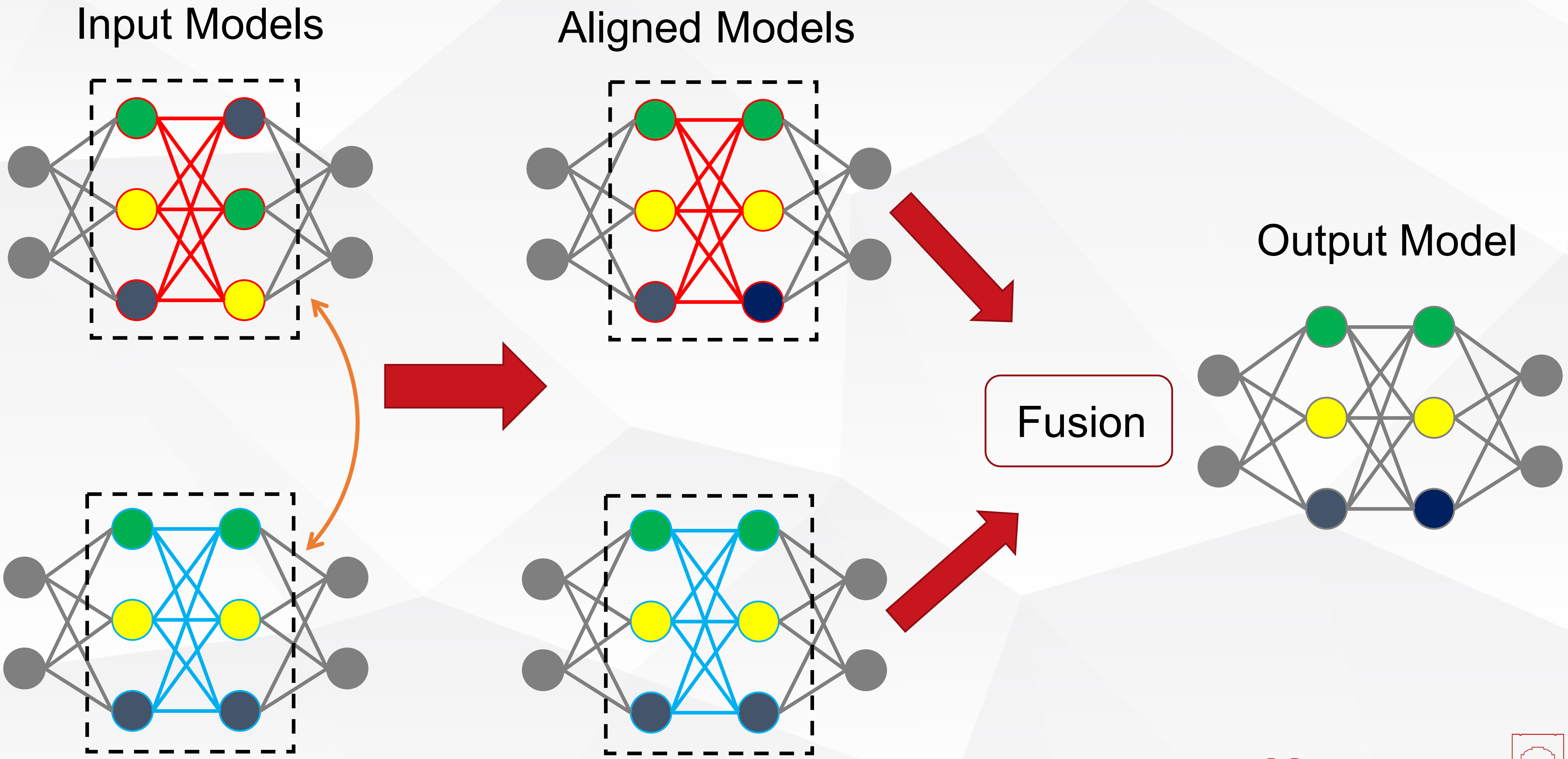
京东探索研究院
JD EXPLORE ACADEMY



01

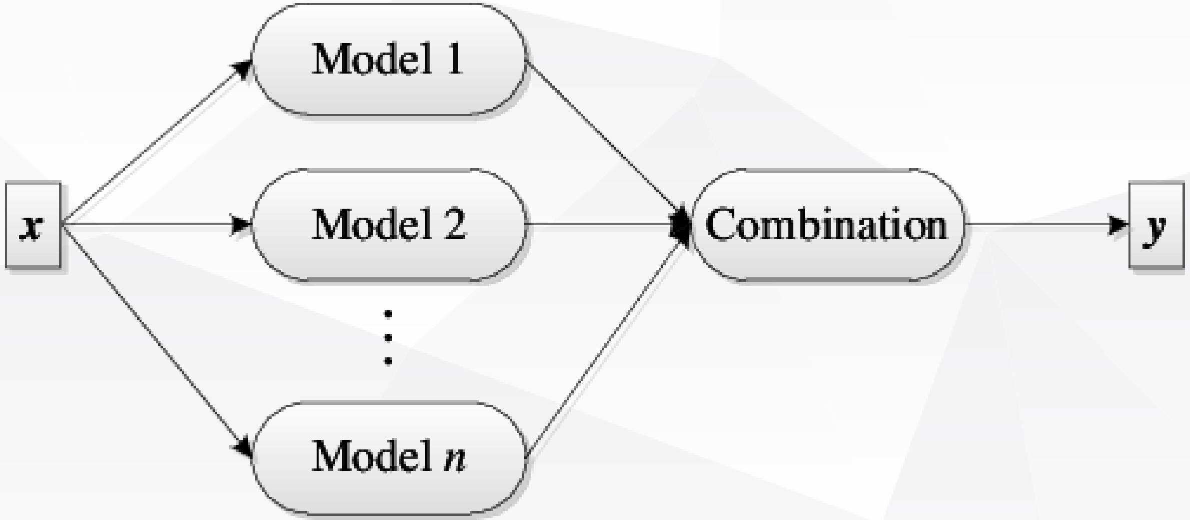
Model Fusion

What is Model Fusion



Why Model Fusion

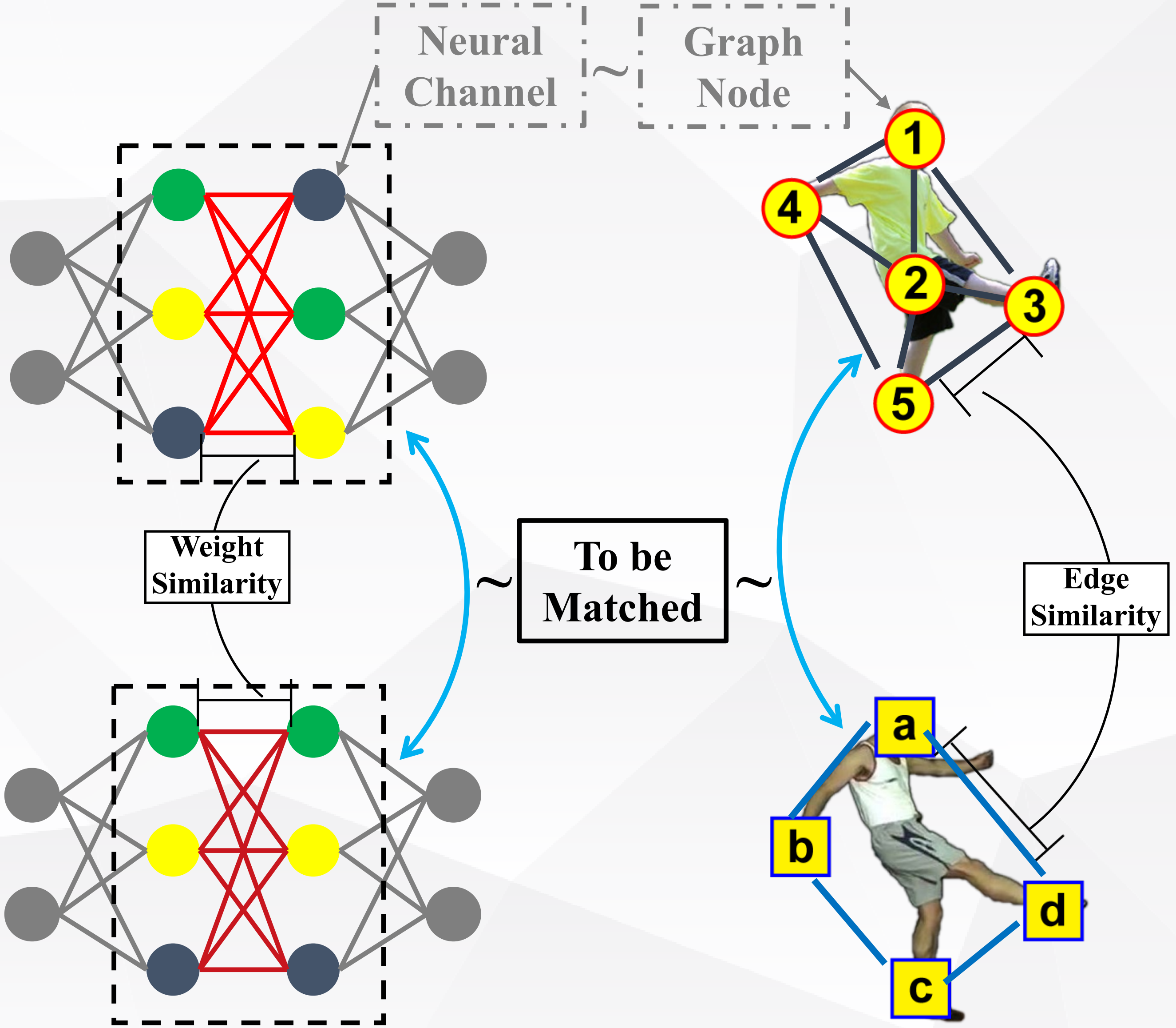
1. Compact Model Ensemble
   - Prediction based ensemble : maintain all individual models.
   - Model fusion based ensemble : maintain only one model instead of all.
2. Federated Learning
   - Each client use their data to train their local models.
   - The global server aggregate the local model in the communication round.



02

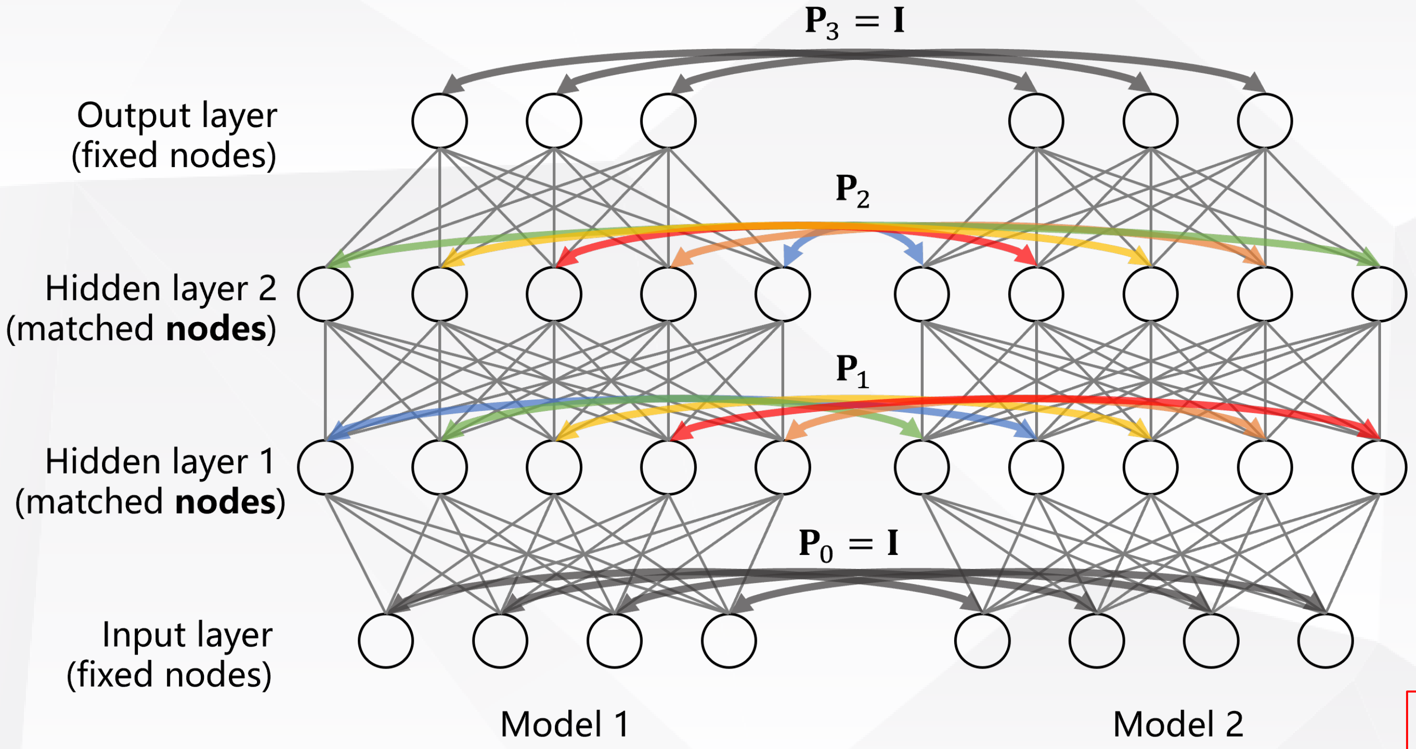
Model Fusion via Graph Matching

Model Fusion via Graph Matching

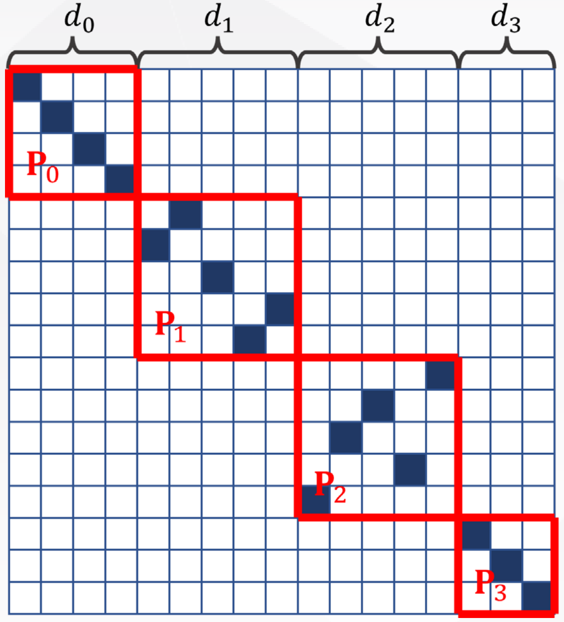


Model Fusion via Graph Matching

- Transfer Model Fusion to a Graph Matching Formulation



The Structure of P



Optimize Goal

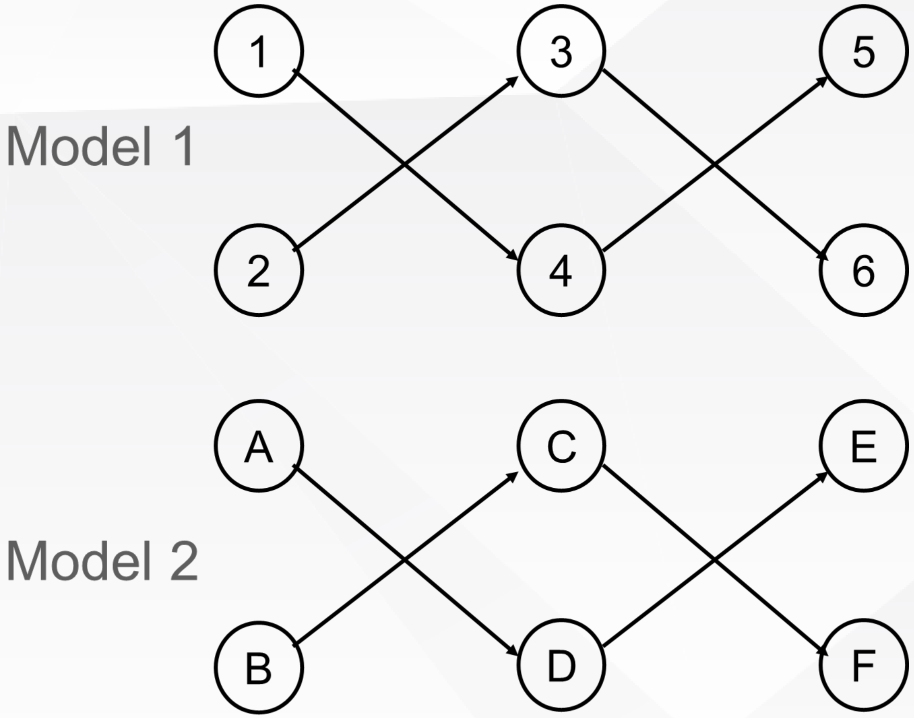
$$\max_{P} \sum_{i=0}^{d_\Sigma-1} \sum_{j=0}^{d_\Sigma-1} \sum_{a=0}^{d_\Sigma-1} \sum_{b=0}^{d_\Sigma-1} P_{[i,j]} K_{[i,j,a,b]} P_{[a,b]}$$

Subject To

$$P_0 = I; P_3 = I; \forall j \sum_{i=0}^{d_1-1} P_{1[i,j]} = 1, \forall i \sum_{j=0}^{d_1-1} P_{1[i,j]} = 1;$$

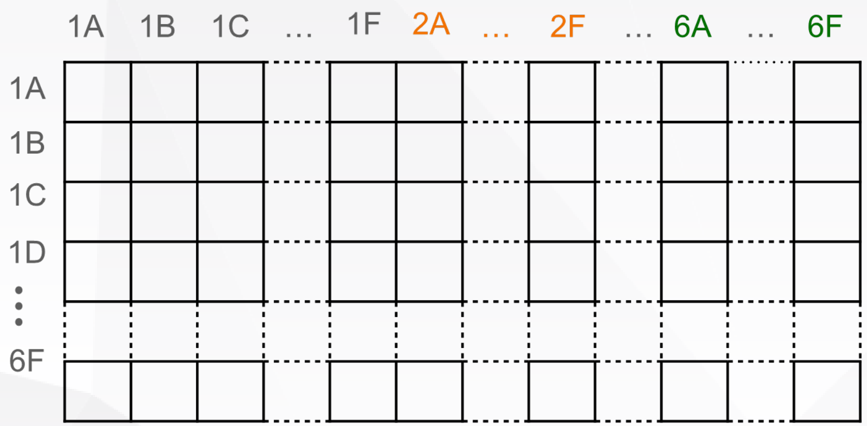
$$\forall j \sum_{i=0}^{d_2-1} P_{2[i,j]} = 1, \forall i \sum_{j=0}^{d_2-1} P_{2[i,j]} = 1.$$

Challenge



Traditional affinity matrix:

size = $((2+2+2)\times(2+2+2))^2 = 1296$

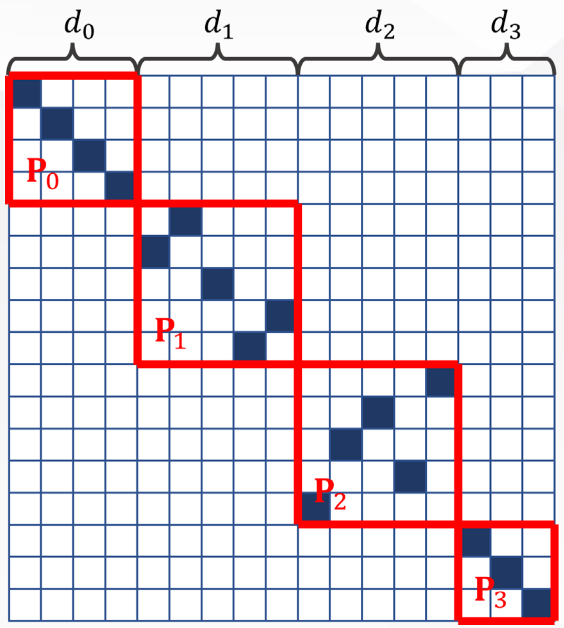


Scalability issue :

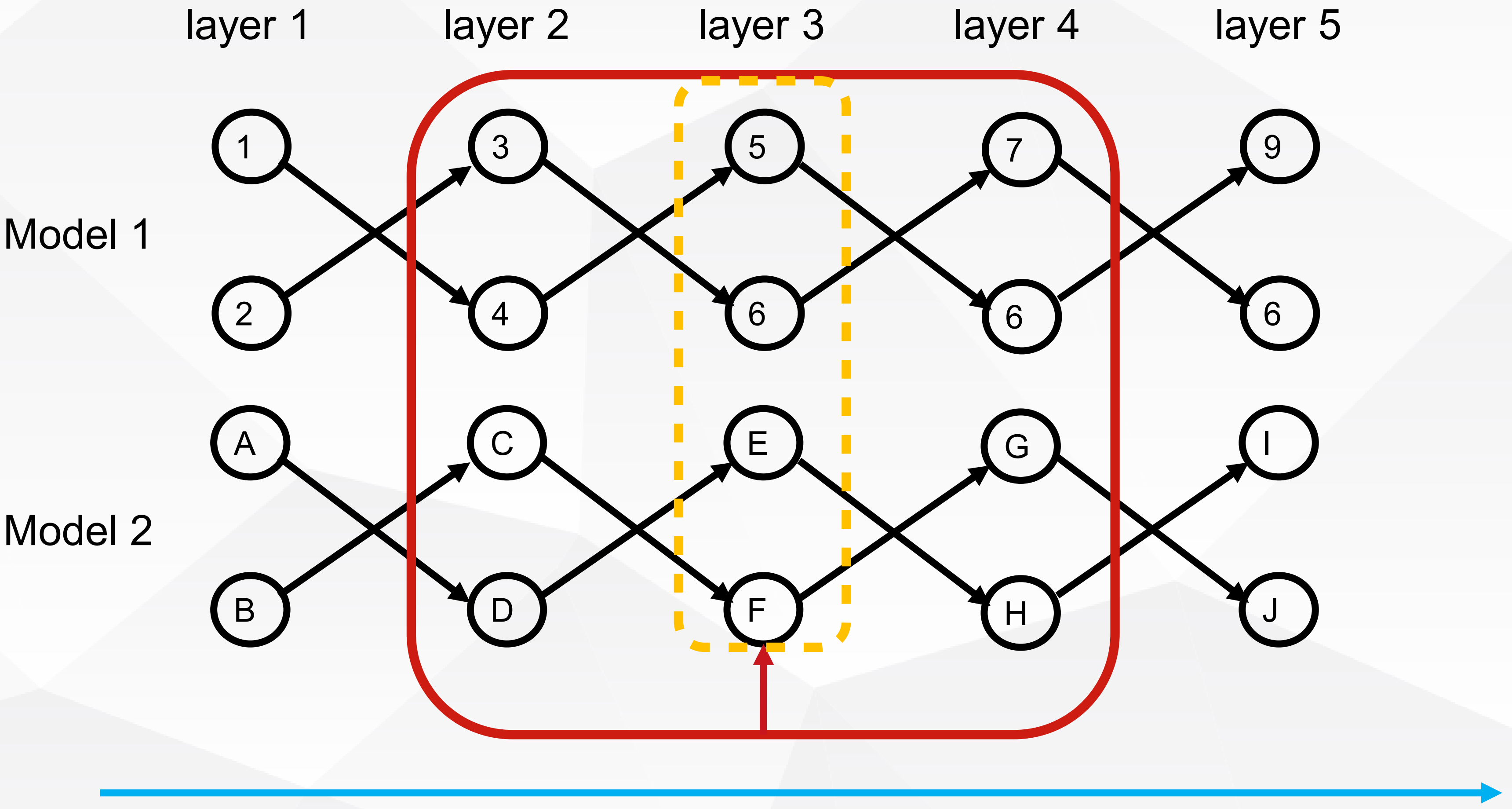
What if we change 2 to 512?

size = $((512+512+512)\times(512+512+512))^2 \approx 5 \times 10^{12}$

Graduated Assignment based Model Fusion (GAMF)



The Structure of P



Graduated Assignment based Model Fusion (GAMF)

Algorithm 1: Graduated Assignment Model Fusion (Two Neural Nets)

Input: weights $\{\mathbf{W}_i^{(1)}\}, \{\mathbf{W}_i^{(2)}\}$; initial annealing τ_0; descent factor γ; minimum τ_{min}; Gaussian kernel σ.

Randomly initialize $\{\mathbf{P}_i\}$; projector $\leftarrow$ Sinkhorn; $\tau \leftarrow \tau_0$;
while *True* **do**
    while $\{\mathbf{P}_i\}$ *not converged* **do**
        $\forall i = 1, 2, \ldots :$
        $$\mathbf{R}_{i[a,b]} = \sum_j \exp\left(-\frac{\left|(\mathbf{P}_{i-1}^\top \mathbf{W}_i^{(1)})_{[j,a]} - \mathbf{W}^{(2)}_{i[j,b]}\right|^2}{\sigma}\right) + \sum_j \exp\left(-\frac{\left|(\mathbf{W}^{(1)}_{i+1} \mathbf{P}_{i+1})_{[a,j]} - \mathbf{W}^{(2)}_{i+1[b,j]}\right|^2}{\sigma}\right);$$
        $\mathbf{P}_i = \text{projector}(\mathbf{R}_i, \tau)$;
    # graduated assignment control
    if projector == Sinkhorn *AND* $\tau \geq \tau_{min}$ **then**
        $\tau \leftarrow \tau \times \gamma$;
    else if projector == Sinkhorn *AND* $\tau < \tau_{min}$ **then**
        projector $\leftarrow$ Hungarian;
    else
        break;

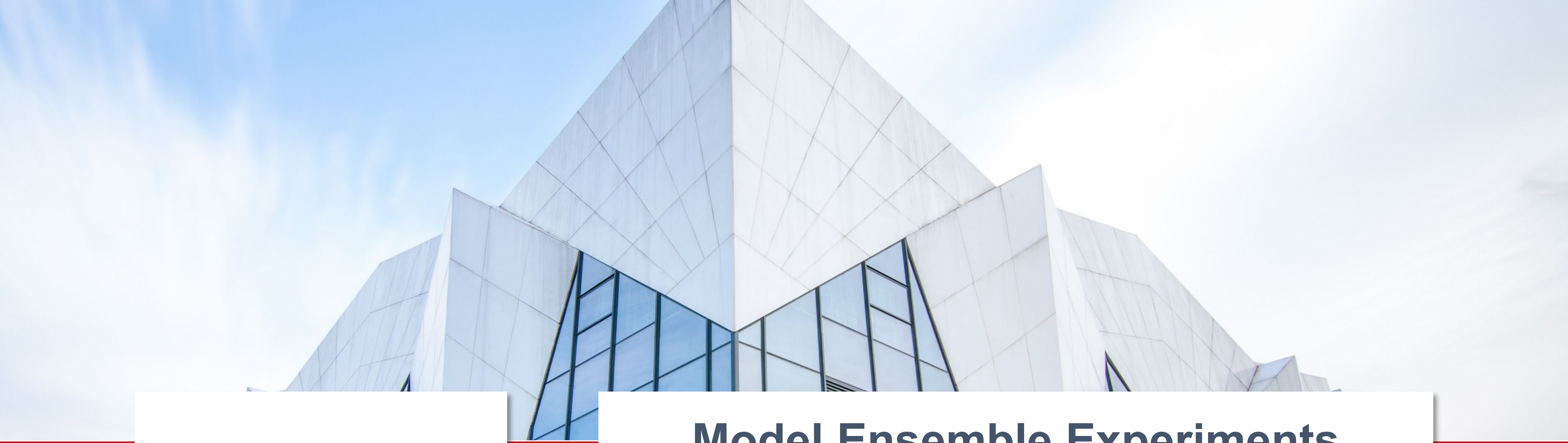
Output: The set of permutation matrices $\{\mathbf{P}_i\}$.

Algorithm 2: Graduated Assignment Model Fusion (Multiple Neural Nets)

Input: weight matrices $\{\mathbf{W}_i^{(k)}\}$; initial annealing τ_0; descent factor γ; minimum τ_{min}; Gaussian kernel parameter σ.

Randomly initialize $\{\mathbf{U}_i^{(k)}\}$; projector $\leftarrow$ Sinkhorn; $\tau \leftarrow \tau_0$;
while *True* **do**
    while $\{\mathbf{U}_i^{(k)}\}$ *not converged* **do**
        $\forall i = 1, 2, \ldots; \forall k = 1, 2, \ldots :$
        $$\mathbf{R}^{(k)}_{i[a,b]} = \sum_{k' \neq k}\left[\sum_j \exp\left(-\frac{\left|(\mathbf{U}_{i-1}^{(k')\top} \mathbf{W}_i^{(k')})_{[j,a]} - (\mathbf{U}_{i-1}^{(k)\top} \mathbf{W}_i^{(k)})_{[j,b]}\right|^2}{\sigma}\right) + \sum_j \exp\left(-\frac{\left|(\mathbf{U}_{i+1}^{(k')\top} \mathbf{W}_{i+1}^{(k')})_{[a,j]} - (\mathbf{U}_{i+1}^{(k)\top} \mathbf{W}_{i+1}^{(k)})_{[b,j]}\right|^2}{\sigma}\right)\right];$$
        $\mathbf{U}_i^{(k)} = \text{projector}(\mathbf{R}_i^{(k)}, \tau)$;
    # graduated assignment control
    if projector == Sinkhorn *AND* $\tau \geq \tau_{min}$ **then**
        $\tau \leftarrow \tau \times \gamma$;
    else if projector == Sinkhorn *AND* $\tau < \tau_{min}$ **then**
        projector $\leftarrow$ Hungarian;
    else
        break;

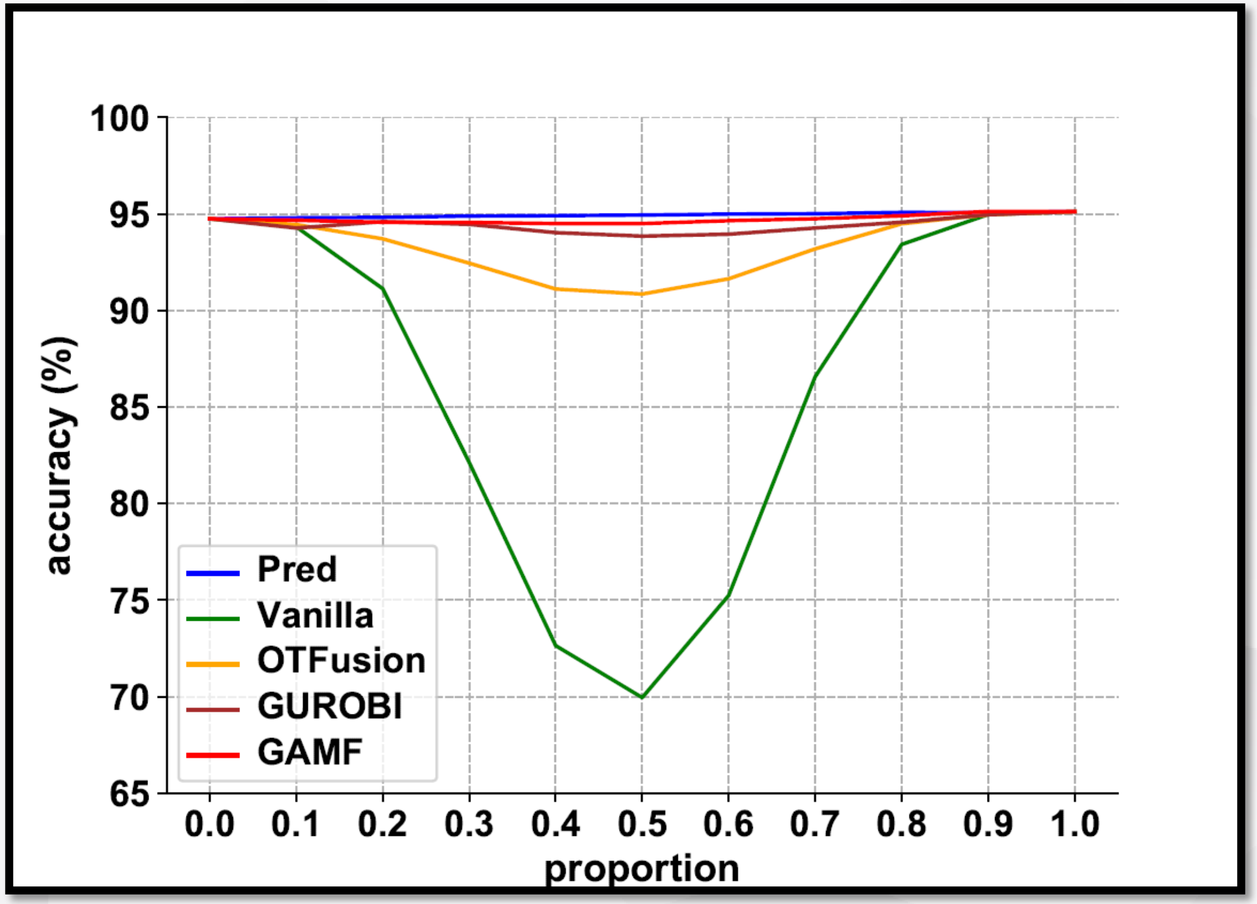
Output: The set of permutation matrices $\{\mathbf{U}_i^{(k)}\}$.



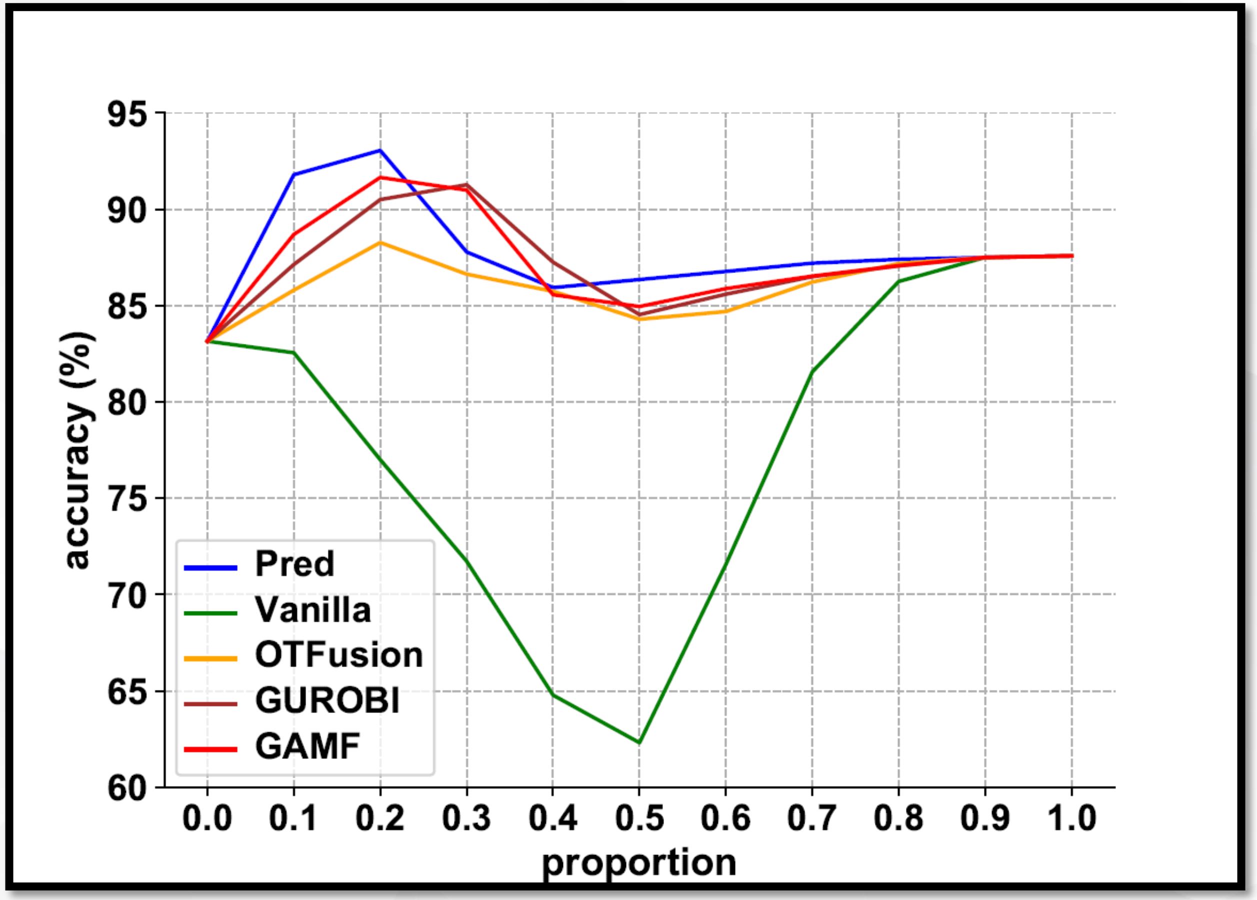
03

Model Ensemble Experiments

MNIST



Homogeneous
Data



Heterogeneous
Data

CIFAR-10

	Data Partition	# of Models ($= N$)	Individual Models	Pred ($N\times$ size)	Vanilla ($1\times$ size)	OTFusion ($1\times$ size)	GAMF ($1\times$ size)
One-shot	Homogeneous	2	[61.32, 62.64]	67.28	16.85	39.04	**49.79**
Finetune			[61.46, 62.94]	–	62.53	63.67	**65.37**
One-shot	Heterogeneous	2	[58.81, 60.70]	67.31	17.52	32.00	**47.91**
Finetune			[63.44, 63.79]	–	58.73	62.29	**64.15**
One-shot	Homogeneous	4	[61.32, 62.64, 63.03, 61.58]	68.97	13.21	14.13	**33.51**
Finetune			[62.02, 61.28, 62.34, 61.55]	–	64.59	64.90	**66.35**
One-shot	Heterogeneous	4	[56.94, 54.15, 57.55, 59.00]	67.81	12.43	27.10	**41.25**
Finetune			[63.58, 61.72, 62.98, 63.79]	–	59.1	63.63	**64.33**

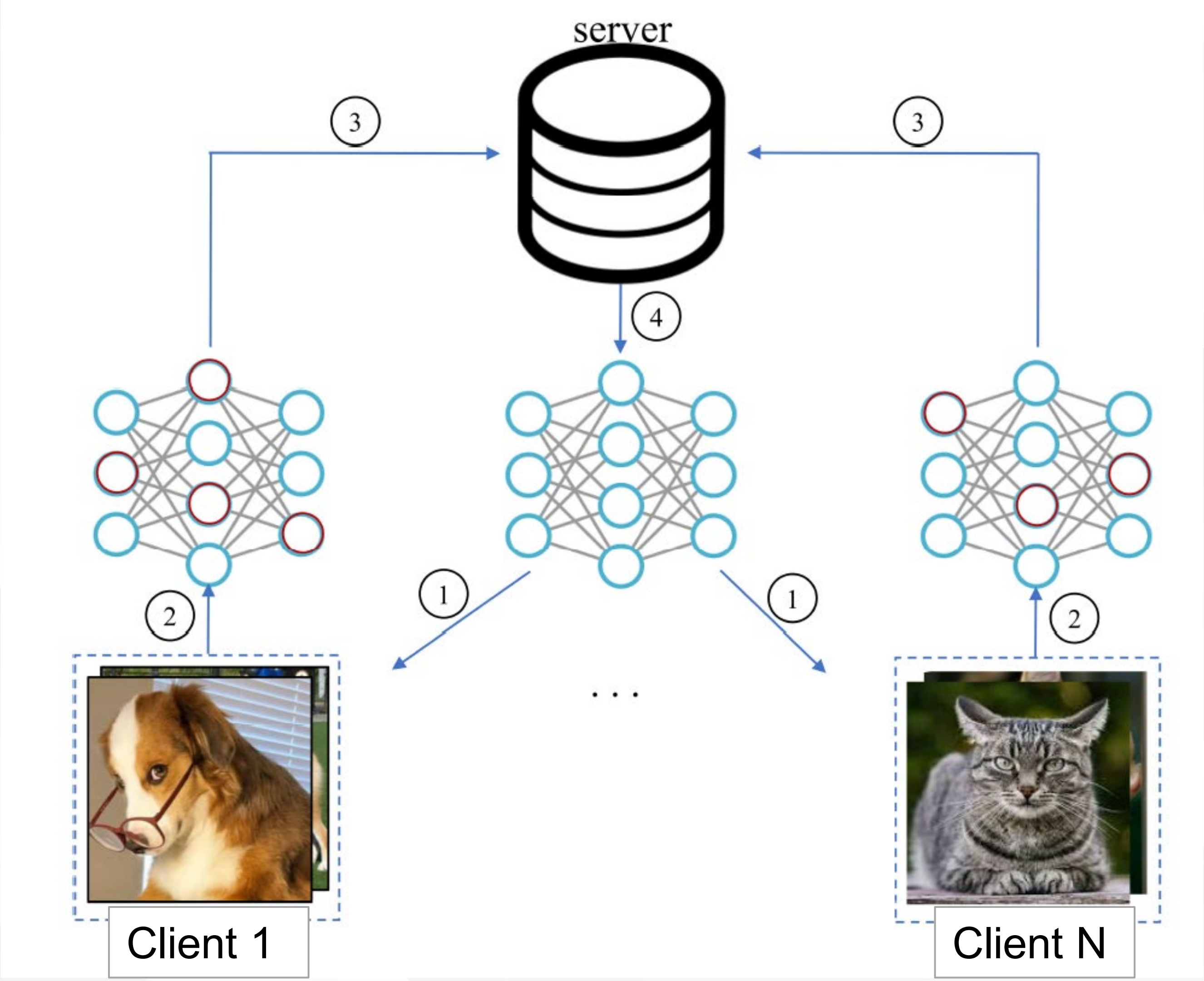
	Data Partition	# of Models ($= N$)	Individual Models	Pred ($N\times$ size)	Vanilla ($1\times$ size)	OTFusion ($1\times$ size)	GAMF ($1\times$ size)
One-shot	Homogeneous	2	[90.31, 90.50]	91.34	17.01	85.98	**87.02**
Finetune			[90.29, 90.53]	–	90.41	90.68	**90.75**
One-shot	Heterogeneous	2	[69.29, 71.89]	75.46	9.84	9.87	**36.73**
Finetune			[71.37, 75.96]	–	60.34	62.08	**79.40**
One-shot	Homogeneous	4	[90.31, 90.50, 90.47, 90.56]	91.91	9.99	**73.56**	73.42
Finetune			[90.29, 90.53, 90.45, 90.55]	–	69.33	**90.89**	90.87
One-shot	Heterogeneous	4	[73.88, 70.73, 72.50, 71.53]	79.87	9.24	9.99	**12.35**
Finetune			[76.76, 75.96, 77.25, 75.24]	–	43.63	48.21	**50.54**

04

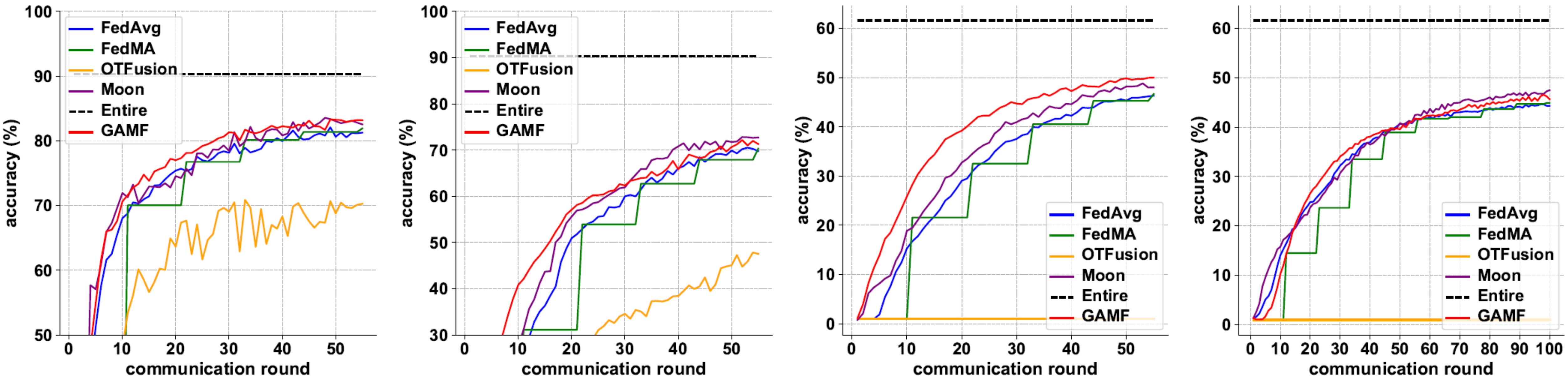
Federated Learning Experiments

Federated Learning

1. Server sends the global model to the clients.
2. Clients update the model with local data.
3. Clients send their local models to the server.
4. Server update the global model by aggregating all local models.



Results



(a) CIFAR-10; 5 clients (b) CIFAR-10; 10 clients (c) CIFAR-100; 5 clients (d) CIFAR-100; 10 clients

1. Server sends the global model to the clients.
2. Clients update the model with local data. (Moon)
3. Clients send their local models to the server.
4. Server update the global model by aggregating all local models. (GAMF) (OTFusion) (FedMA)

Results

	CIFAR-10; 5 clients	CIFAR-10; 10 clients	CIFAR-100; 5 clients	CIFAR-100; 10 clients	Tiny-Imagenet
FedAvg [3]	81.01% ± 0.31%	69.99% ± 0.40%	45.94% ± 0.32%	44.42% ± 0.13%	22.87% ± 0.11%
OTFusion [4]	69.83% ± 0.55%	46.40% ± 1.01%	1.00% ± 0.00%	1.00% ± 0.00%	0.50% ± 0.00%
FedMA [5]	81.46% ± 0.20%	70.29% ± 0.69%	47.50% ± 0.52%	44.95% ± 0.19%	23.19% ± 0.16%
Moon [2]	82.78% ± 0.57%	72.42% ± 0.45%	48.24% ± 0.28%	46.99% ± 0.28%	23.49% ± 0.10%
GAMF (ours)	82.82% ± 0.58%	72.39% ± 0.54%	**49.80%** ± 0.25%	45.99% ± 0.41%	23.96% ± 0.12%
GAMF + Moon	**84.92%** ± 0.39%	**73.43%** ± 0.59%	48.72% ± 0.78%	**48.24%** ± 0.39%	**24.61%** ± 0.11%

Table 1. The top-1 accuracy of the compared methods on CIFAR-10, CIFAR-100, and Tiny-Imagenet.

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

Thanks~

Reference:

1. Li, Q., He, B., & Song, D.X. (2021). Model-Contrastive Federated Learning. *CVPR 2021*, 10708-10717
2. Singh, S. P., & Jaggi, M. (2020). Model fusion via optimal transport. Advances in Neural Information Processing Systems, 33, 22045-22055.
3. Wang, H., Yurochkin, M., Sun, Y., Papailiopoulos, D., & Khazaeni, Y. (2020). Federated Learning with Matched Averaging. ArXiv, abs/2002.06440.
4. McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In Artificial intelligence and statistics (pp. 1273-1282). PMLR.

饮水思源 爱国荣校