JUNE 18-22, 2023
CVPR
VANCOUVER, CANADA

Paper Tag: THU-AM-347

Partial Network Cloning

Jingwen Ye, Songhua Liu, Xinchao Wang
National University of Singapore

NUS National University of Singapore

http://www.lv-nus.org/

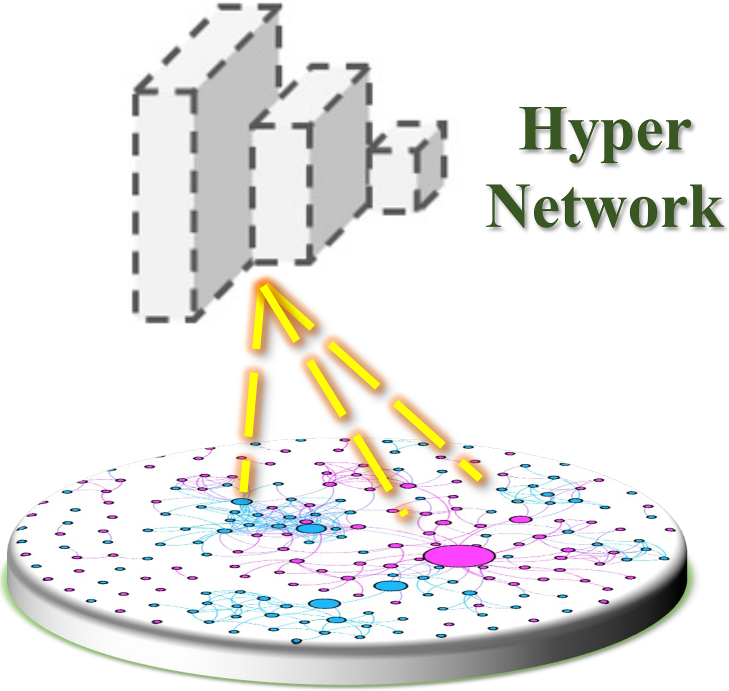
Quick Review

[Goal] Build a new network by **connecting** instead of creating.

$$\mathcal{M}_c \leftarrow Clone(\mathcal{M}_t, M, \mathcal{M}_s, R)$$

Two Steps

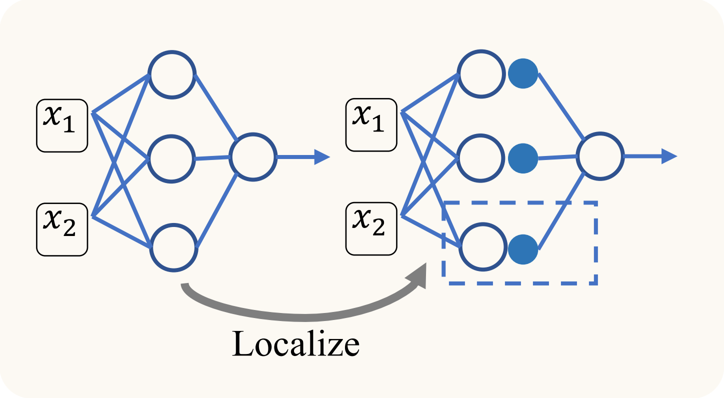
$$\mathcal{M}_f^\rho \leftarrow Local(\mathcal{M}_s^\rho, M^\rho),$$
$$\mathcal{M}_c \leftarrow Insert_{\rho=0}^{P}(\mathcal{M}_t, \mathcal{M}_f^\rho, R^\rho)$$



$\mathcal{M}_s, \mathcal{M}_t$: Pretrained Networks
M: Masking parameters
P: Position parameters

Quick Review

Step I: Localize with Pruning

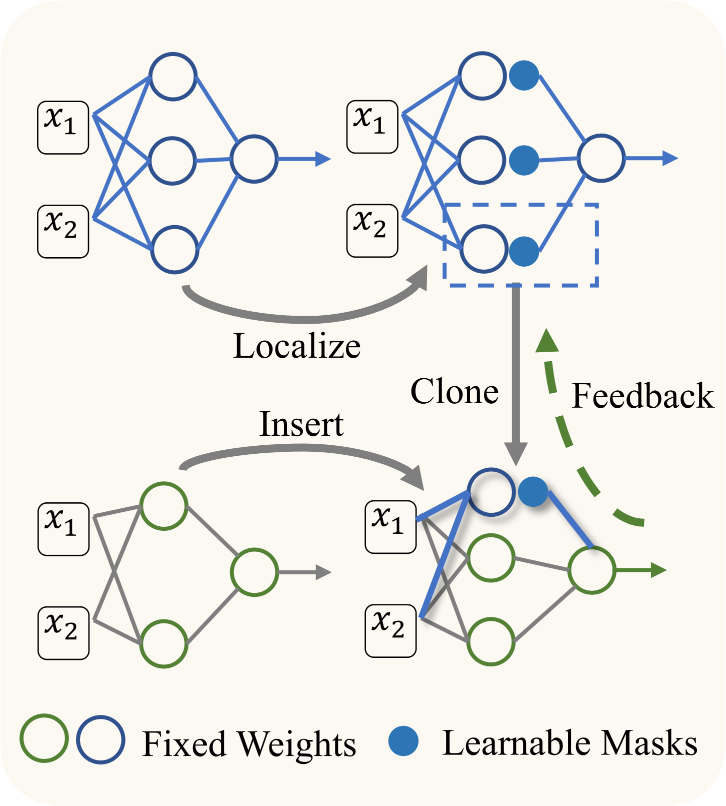


To model the source $\mathcal{M}_s$ in the $\mathcal{D}_t$ neighborhood, and then use the local model set as the surrogate:

$$\mathcal{G} = \{g_i\}^{(N)} \approx \mathcal{M}_s | \mathcal{D}_t$$

Quick Review

Step II: Insert with adaptation



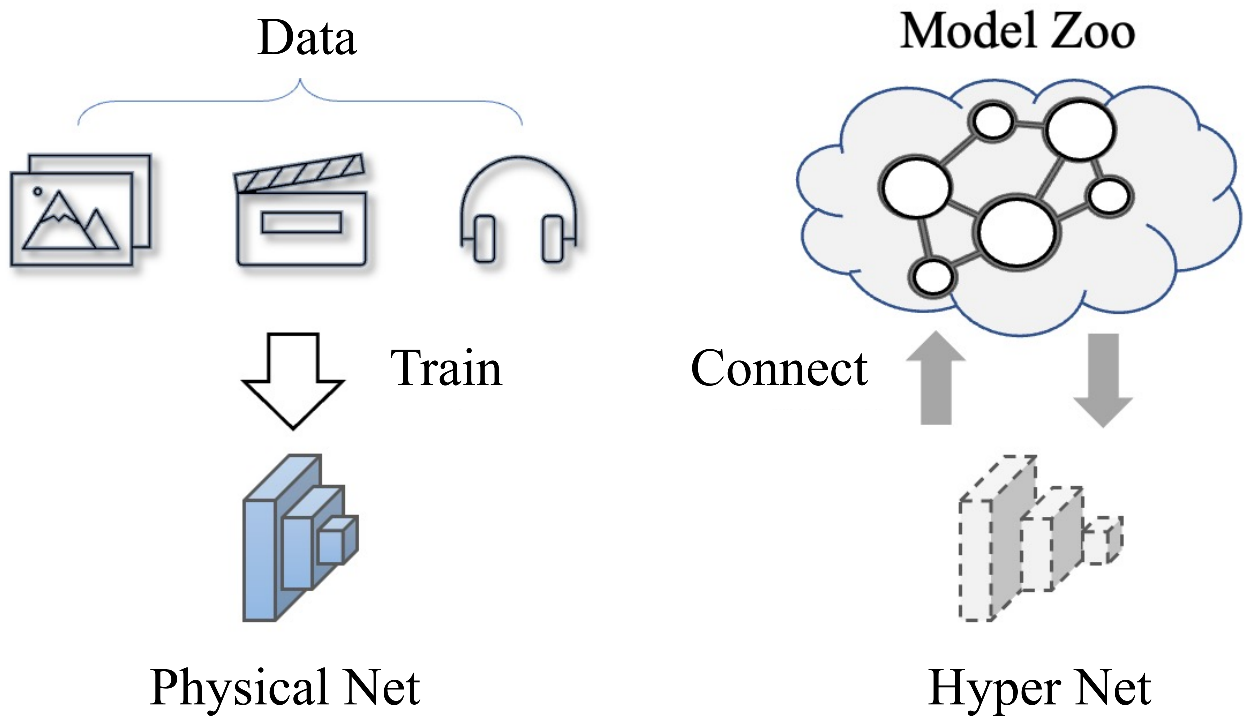
The learning-to-insert process with R is simplified as finding the best position:

$$\mathcal{M}_c^R \leftarrow \mathcal{M}_t\left(W_t^{[0:R]}\right) \circ \left\{\mathcal{M}_{t'}\left(W_t^{[R:L]}\right) \mathcal{M}_f\right\}$$

$$\min_{\mathcal{F}_c,\mathcal{A}} \mathcal{L}_{kd} \circ f_t\big[\mathcal{F}_c\big(\mathcal{A}; \mathcal{M}_c^R(B \cdot x)\big),$$
$$\mathcal{G}(B)\big] + \mathcal{L}_{kd} \circ \overline{f_t}\big[\mathcal{F}_c\big(\mathcal{A}; \mathcal{M}_c^R(B \cdot x)\big), \mathcal{M}_t(B \cdot x)\big]$$

$$R: (L-1) \rightarrow 0$$

Background

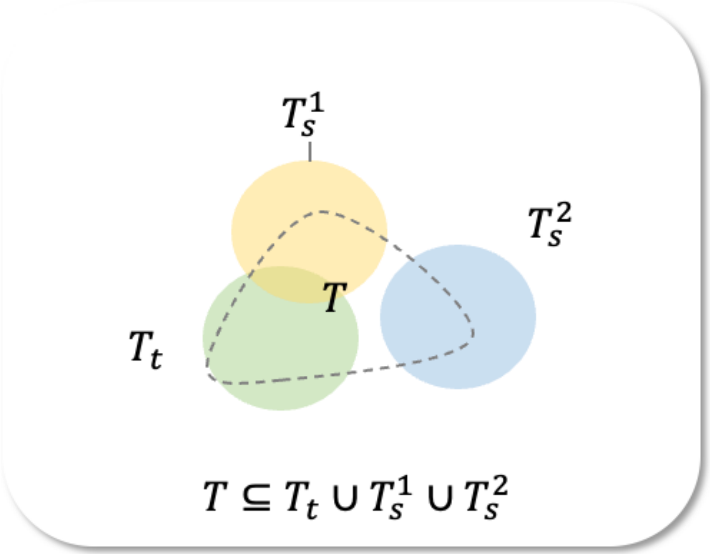


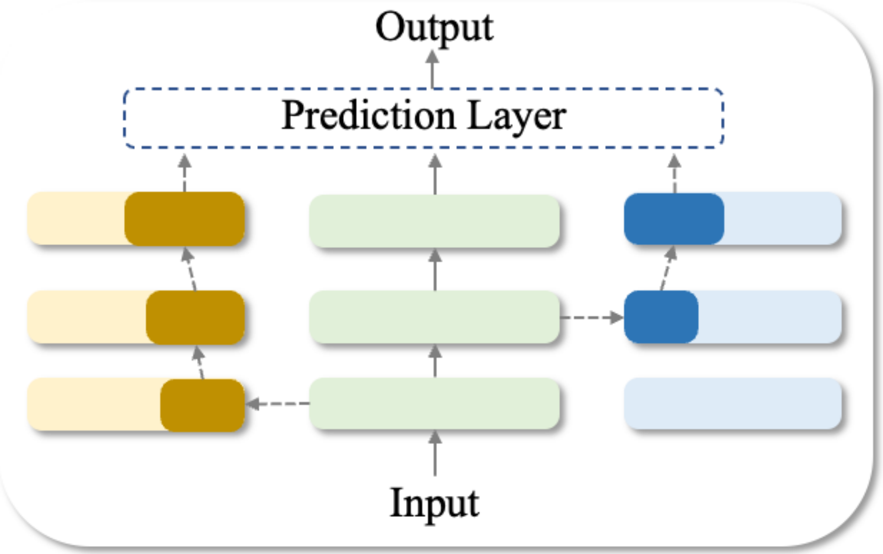
Main Idea

JUNE 18-22, 2023 CVPR VANCOUVER, CANADA

Three steps to build a hyper network:

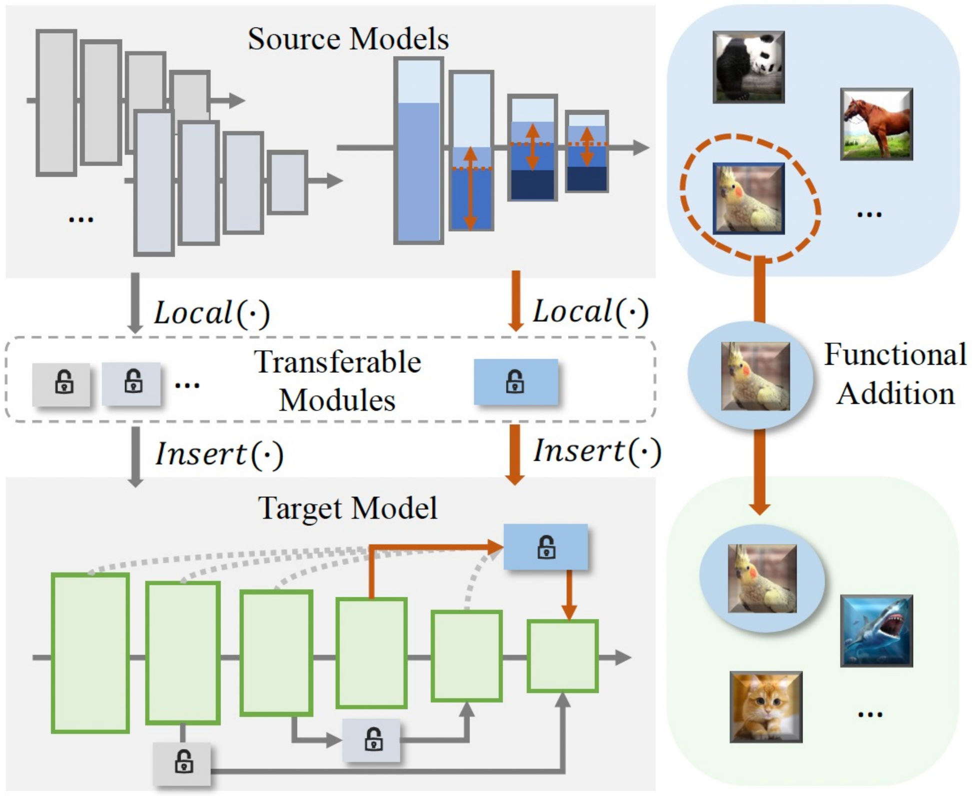
Step I: Determine target network $\mathcal{M}_t$;
Step II: Clone from the source networks $\mathcal{M}_s$;
Step III: Finetune the prediction layers;





Main Idea

<u>The key to PNC is to learn an optimal transferable module!</u>



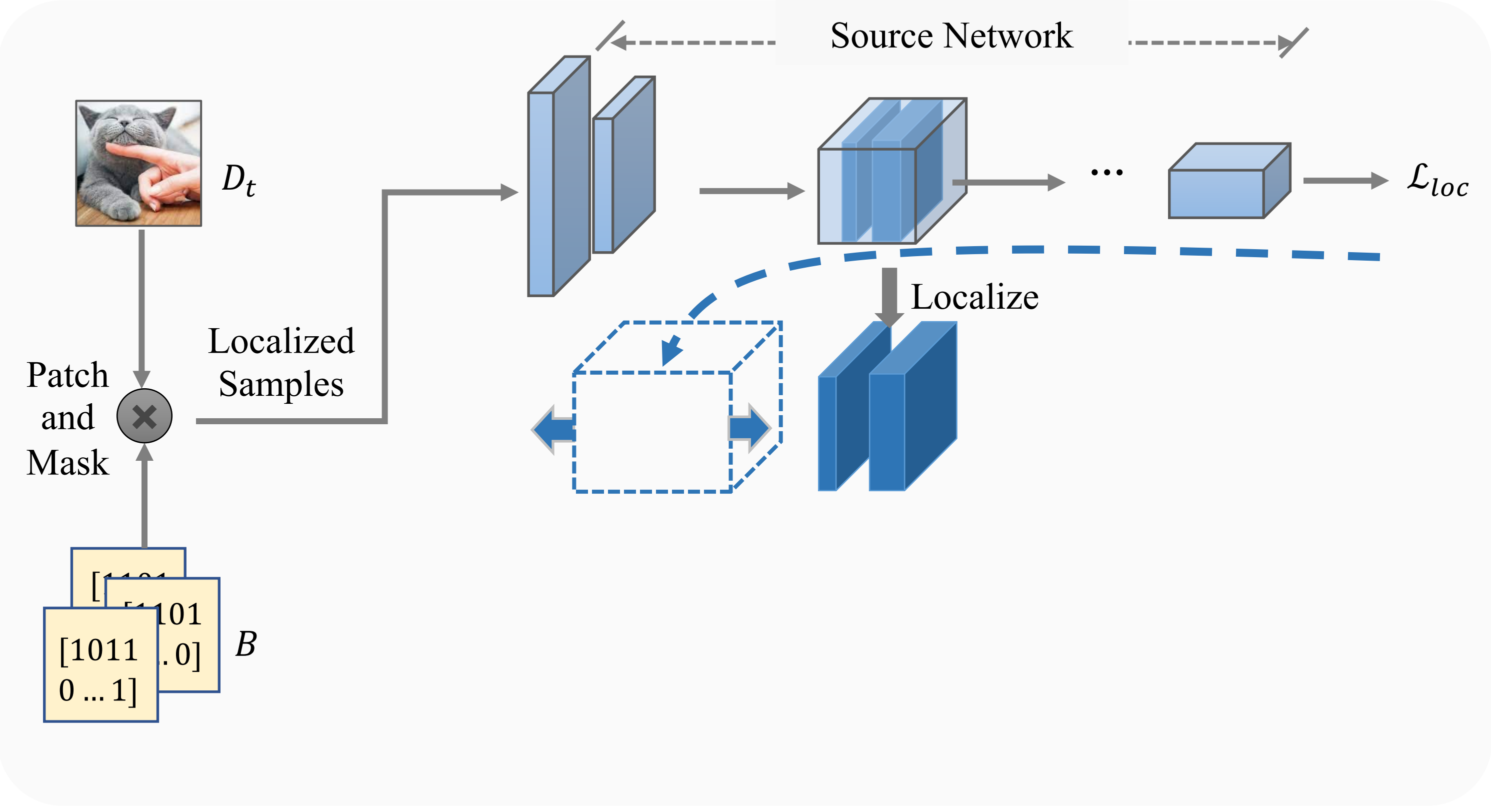
- **Transferablity:** The extracted transferable module should contain the explicit knowledge of the to-be-cloned task T_s, which could be transferred effectively to the downstream networks;
- **Locality:** The influence on the cloned model $\mathcal{M}_c$ out of the target data D_t should be minimized;
- **Efficiency:** Functional cloning should be efficient in terms of runtime and memory;
- **Sustainability:** The process of cloning wouldn't do harm to the model zoo, meaning that no modification the pre-trained models are allowed and the cloned model could be fully recovered.

Main Idea

- Localize with pruning $\mathcal{M}_f^\rho \leftarrow Local(\mathcal{M}_s^\rho, M^\rho)$
- Insert with adaptation $\mathcal{M}_c \leftarrow Insert_{\rho=0}^{P}(\mathcal{M}_t, \mathcal{M}_f^\rho, R^\rho)$

Method

➢ Localize with pruning: $\mathcal{M}_f^{\rho} \leftarrow Local(\mathcal{M}_s^{\rho}, M^{\rho})$



- The localization can be denoted as:

$$\mathcal{M}_f \leftarrow M \cdot \mathcal{M}_s \Leftrightarrow \{m^l | \cdot w_s^l \ 0 \leq l < L\}$$

- We use the local model set as the surrogate:

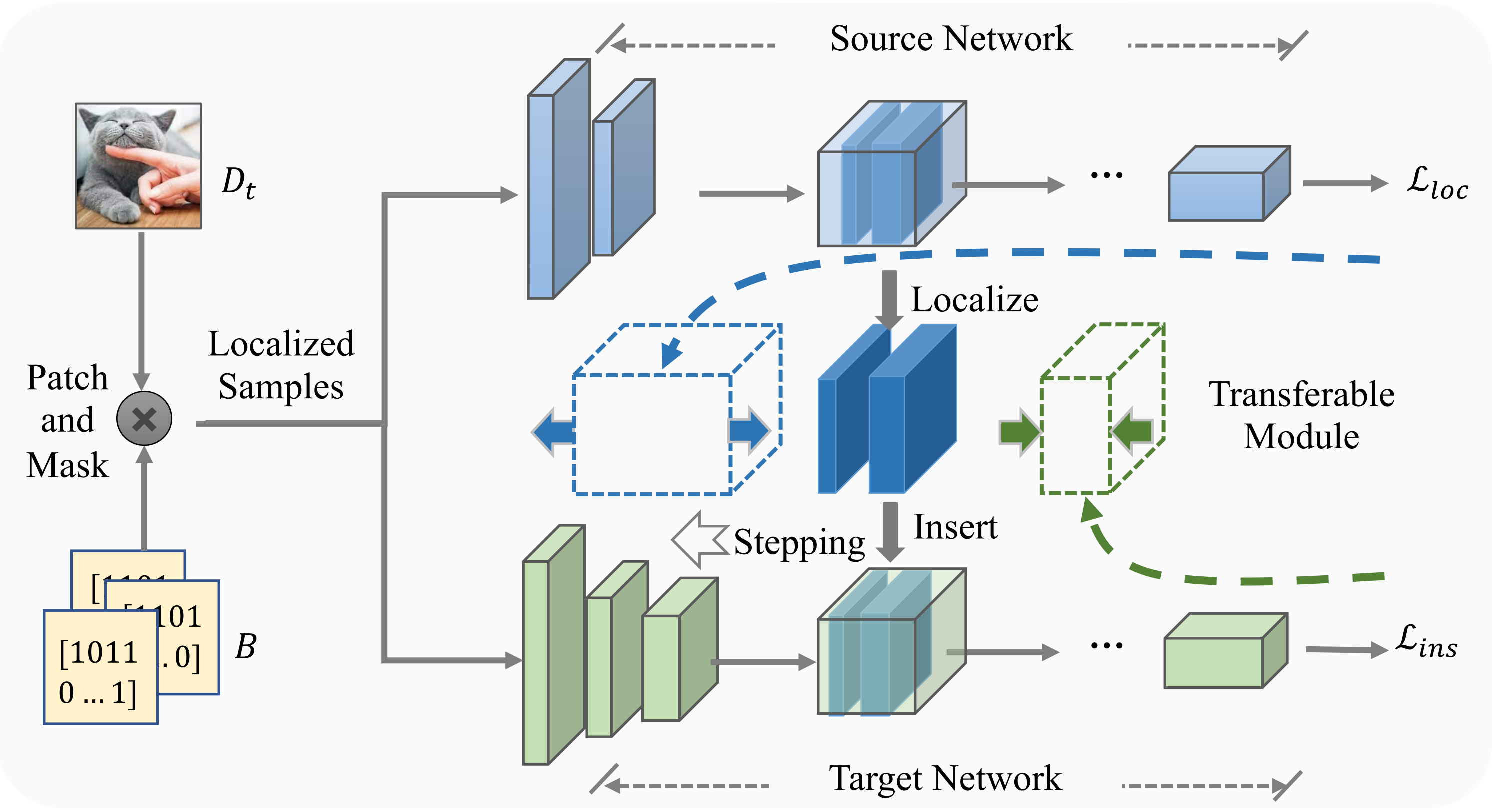
$$\mathcal{G} = \{g_i\}^{(N)} \approx \mathcal{M}_s | \mathcal{D}_t$$

- The localization process could be optimized as:

$$\min_M \sum_{g_i \in \mathcal{G}} \sum_{b \in B} \| f_t[\mathcal{M}_s(M \cdot W_s; b \cdot x)] - f_t[g_i(b)] \|^2$$

Method

➢ Insert with adaptation: $\mathcal{M}_c \leftarrow Insert_{\rho=0}^{P}(\mathcal{M}_t, \mathcal{M}_f^{\rho}, R^{\rho})$



- The process is simplified as finding the best position to insert the transferable module:

$$\mathcal{M}_c^R \leftarrow \mathcal{M}_t\left(W_t^{[0:R]}\right) \circ \left\{\mathcal{M}_{t'}\left(W_t^{[R:L]}\right)\mathcal{M}_f\right\}$$

$$\min_{\mathcal{F}_c,\mathcal{A}} \mathcal{L}_{kd} \circ f_t\left[\mathcal{F}_c\left(\mathcal{A}; \mathcal{M}_c^R(B \cdot x)\right), \mathcal{G}(B)\right] + \mathcal{L}_{kd} \circ \bar{f}_t\left[\mathcal{F}_c\left(\mathcal{A}; \mathcal{M}_c^R(B \cdot x)\right), \mathcal{M}_t(B \cdot x)\right]$$

$$R: (L-1) \rightarrow 0$$

✓ *While training, R is firstly set to be L−1 and then moving layer by layer to R = 0;*

✓ In each moving step, we finetune the adapter and the corresponding fully connected layers.

Cloning in various usages

[Scenario I] Partial network cloning is a better form for information transmission.

When there is a request for transferring the networks, it is better to transfer the cloned network obtained by PNC as **to reduce latency and transmission loss.**

[Scenario II] Partial network cloning enables model zoo online usage.

In some resource limited situation, the users could **flexibly utilize model zoo online** without downloading it on local.

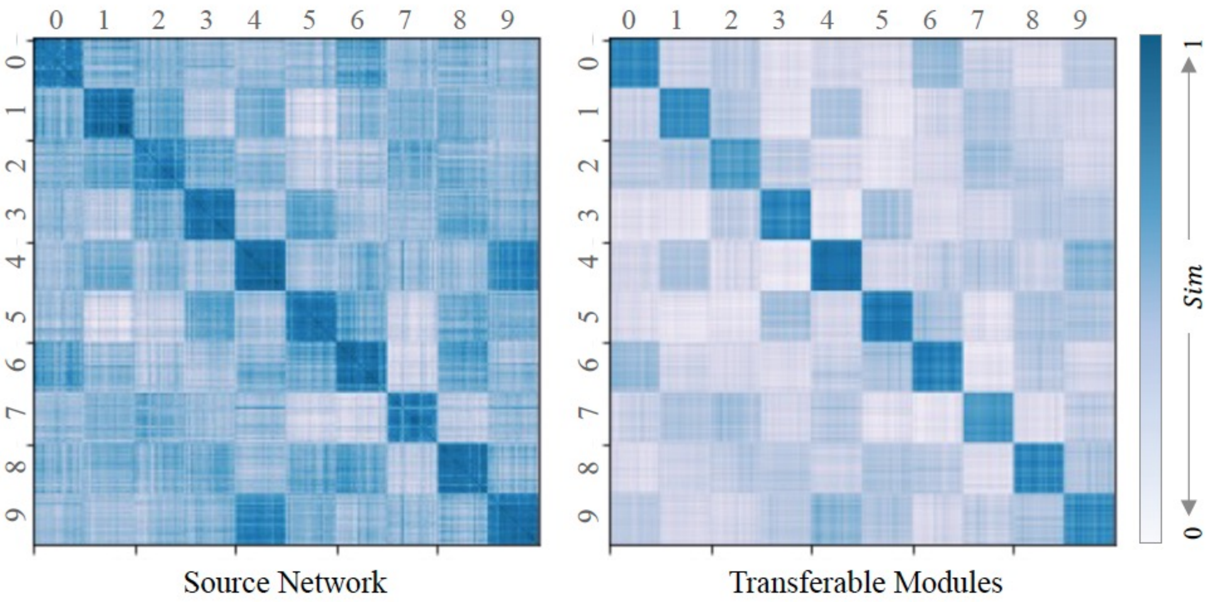
Experiments

Method	Acc on MNIST (LeNet5, #3 Steps) Ori.-S	Tar.-S	Avg.-S	Ori.-M	Tar.-M	Avg.-M	Acc on CIFAR-10 (ResNet-18, #5 Steps) Ori.-S	Tar.-S	Avg.-S	Ori.-M	Tar.-M	Avg.-M
Pre-trained	99.7	99.5	99.7	99.7	99.5	99.6	95.9	97.2	96.1	95.9	97.6	96.5
Joint+Full Set	99.8	98.3	99.6	99.7	99.3	99.5	95.2	96.8	95.5	94.4	95.1	94.7
Continual	83.4-10.1	100.0+17.3	86.2-5.5	65.1-27.9	98.8+16.8	77.7-11.2	67.7+2.8	97.2+2.6	75.3-14.8	92.8+18.7	78.2+16.6	87.3-2.1
Direct Ensemble	94.6+1.1	56.1-26.4	88.2-3.5	94.6+1.6	81.9-0.1	89.8+0.9	90.5+25.6	39.3-55.3	82.0+12.1	90.5+16.4	43.8-17.8	73.0+3.6
Continual+KD	*93.5*	*82.7*	*91.7*	*93.0*	*82.0*	*88.9*	*64.9*	*94.6*	*69.9*	*74.1*	*61.6*	*69.4*
PNC-F (w/o Local)	87.7-5.8	100.0+17.3	90.0-1.7	90.9-2.1	98.2+16.2	93.6+4.7	88.6+23.7	**97.3**+2.7	90.1+20.2	85.5+11.4	95.8+34.2	89.4+20.0
PNC-F (w/o Insert)	86.9-6.6	100.0+17.3	89.1-2.6	90.4-2.6	97.7+15.7	93.1+4.2	86.1+21.2	96.8+2.2	87.9+18.0	86.0+11.9	**96.2**+34.6	89.8+30.4
PNC-F (full)	88.5-5.0	**99.7**+17.0	90.4-2.6	91.1-1.9	98.8+16.8	94.0+5.1	83.0+18.1	96.5+1.9	85.3+15.4	85.4+11.3	95.5+33.9	89.2+19.8
PNC (w/o Local)	93.6+0.1	96.2+13.5	94.0+2.3	92.9-0.1	94.0+12.0	93.3+4.4	90.5+25.6	93.9-0.7	91.7+21.8	87.1+13.0	94.6+33.1	89.9+29.8
PNC (w/o Insert)	92.8-0.7	99.5+16.8	93.9+2.2	91.9-1.1	97.3+15.3	93.9+5.0	89.5+24.6	94.4-0.2	90.3+20.4	89.2+15.1	94.7+33.2	91.3+21.9
PNC (Ours, full)	**96.4**+2.9	**99.7**+17.0	**97.0**+5.3	**96.2**+3.2	**97.8**15.8	**96.8**+7.9	**94.9**+30.0	95.5+0.9	**95.0**+25.1	**93.7**+19.6	94.5+32.9	**94.0**+24.6

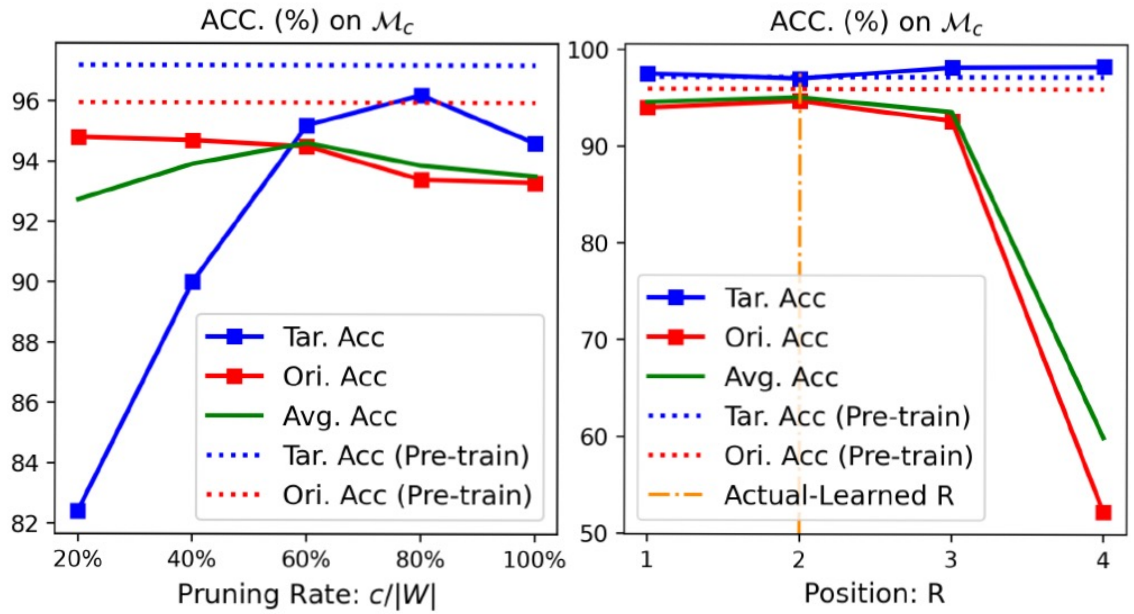
Method	Acc on CIFAR-100 (ResNet-50, #5 Steps) Ori.-S	Tar.-S	Avg.-S	Ori.-M	Tar.-M	Avg.-M	Acc on Tiny-ImageNet (ResNet-18, #5 Steps) Ori.-S	Tar.-S	Avg.-S	Ori.-M	Tar.-M	Avg.-M
Pre-trained	80.0	80.3	80.1	80.0	77.2	79.0	71.3	67.6	70.7	71.3	68.9	70.4
Joint+Full Set	78.0	74.9	77.5	76.3	77.9	76.9	63.1	60.8	62.7	63.7	61.6	62.9
Direct Ensemble	59.3-6.2	46.4-26.3	57.2-9.6	56.0-18.4	46.4-26.6	52.4-21.5	58.0+0.8	35.9-20.5	54.3-2.8	50.6-9.3	30.2-27.9	43.0-16.3
Continual	52.3-13.2	**79.4**+6.7	56.8-9.9	58.8-15.6	**78.0**+5.0	66.0-7.9	54.6-2.6	**70.1**+13.7	57.2+0.1	55.9-4.0	**64.9**+6.8	59.3+0.1
Continual + KD	*65.5*	*72.7*	66.7	*74.4*	*73.0*	73.9	*57.2*	*56.4*	57.1	*59.9*	*58.1*	59.2
PNC (w/o Local)	72.2+6.7	70.4-2.3	71.9+5.2	75.7+1.3	68.3-4.7	72.9-1.0	**65.6**+8.4	52.5-3.9	**63.4**+6.4	56.4-3.5	55.9-2.2	56.2-3.0
PNC (w/o Insert)	63.2-2.3	76.1+3.4	65.4-1.3	66.1-8.3	76.0+3.0	69.8-4.1	60.7+3.5	63.5+7.1	61.2+4.1	58.8-1.1	60.9+2.8	59.6+0.4
PNC (Ours, full)	**76.7**+11.2	74.9+2.2	**76.4**+9.7	**76.9**+2.5	76.5+3.5	**76.8**+2.9	63.2+6.0	60.7+4.3	62.8+5.7	**63.5**+3.6	60.4+2.3	**62.3**+3.1

 # Overall performance on partial network cloning on MNIST, CIFAR10, CIFAR100 and Tiny-ImageNet datasets

Experiments



The similarity matrix maps.



#The performance with different scales

JUNE 18-22, 2023 CVPR VANCOUVER, CANADA

Thanks for Watching !

Presenter: Jingwen Ye

Feel free to contact me: *jingweny@nus.edu.sg*

National University of Singapore

http://www.lv-nus.org/