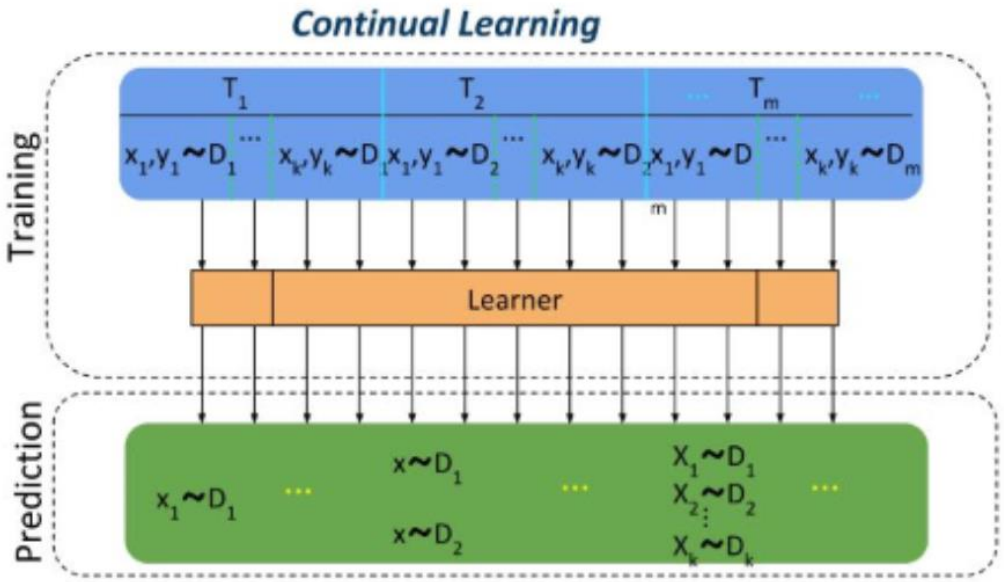
Forget-free Continual Learning with Winning Subnetworks

Haeyong Kang [*1] Rusty John Lloyd Mina [*1] Sultan Rizky Hikmawan Madjid [1] Jaehong Yoon [1]
Mark Hasegawa-Johnson [2] Sung Ju Hwang [1 3] Chang D. Yoo [1]

ICML 2022

Incremental learning



Incremental learning (Continual learning , lifelong learning):

- 独立同分布(i.i.d.)假设
- 稳定性-可塑性困境

Task incremental learning:

$$\{y^{(t)}\} \neq \{y^{(t+1)}\}$$

Task known & Task unknown

Methods:

- Replay
- Regularization-based
- Parameter isolation

Motivation

Limitation: memory usage increases as new tasks arrive.

- Replay: additional room to store the replay buffer.
- Parameter isolation: addition room for models to deal with new task.

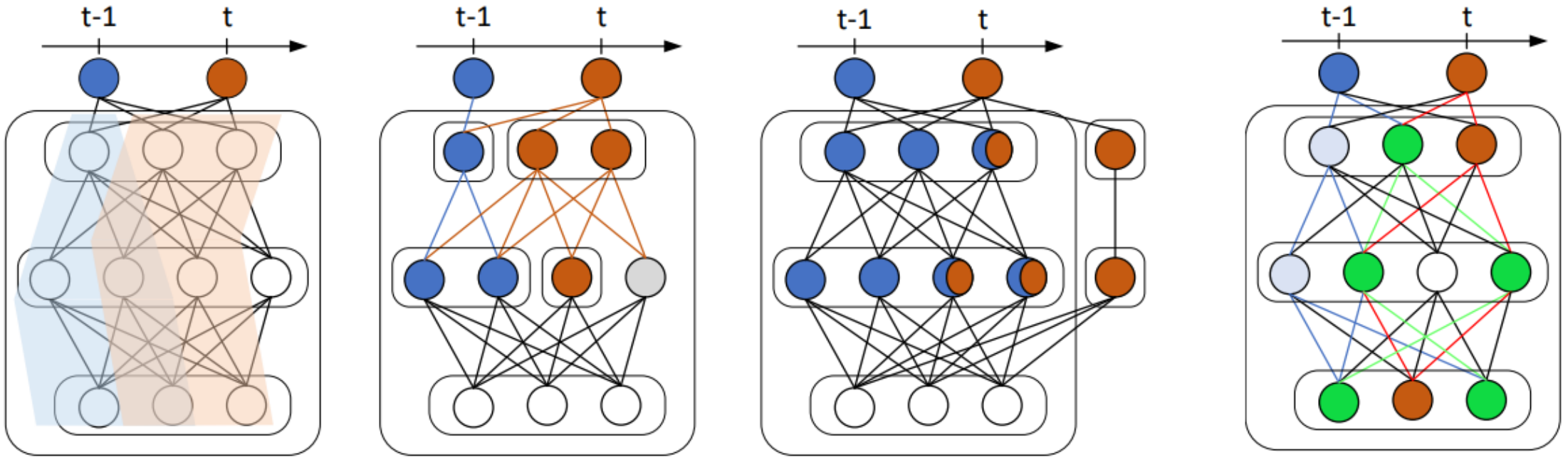
How can we build a memory-efficient CL model that does not exceed the capacity of the backbone network?

Lottery Ticket Hypothesis(LTH):

- Model-purning method
- Over-parameter
- Winner ticket: subnetworks
- “Subnetworks of large network—when trained in isolation—converge in a comparable number of iterations to comparable accuracy”

[1] The Lottery Ticket Hypothesis :Finding Sparse Trainable Neural Networks. ICLR 2019 (best paper)

Motivation



(a) Fixed Backbone Network

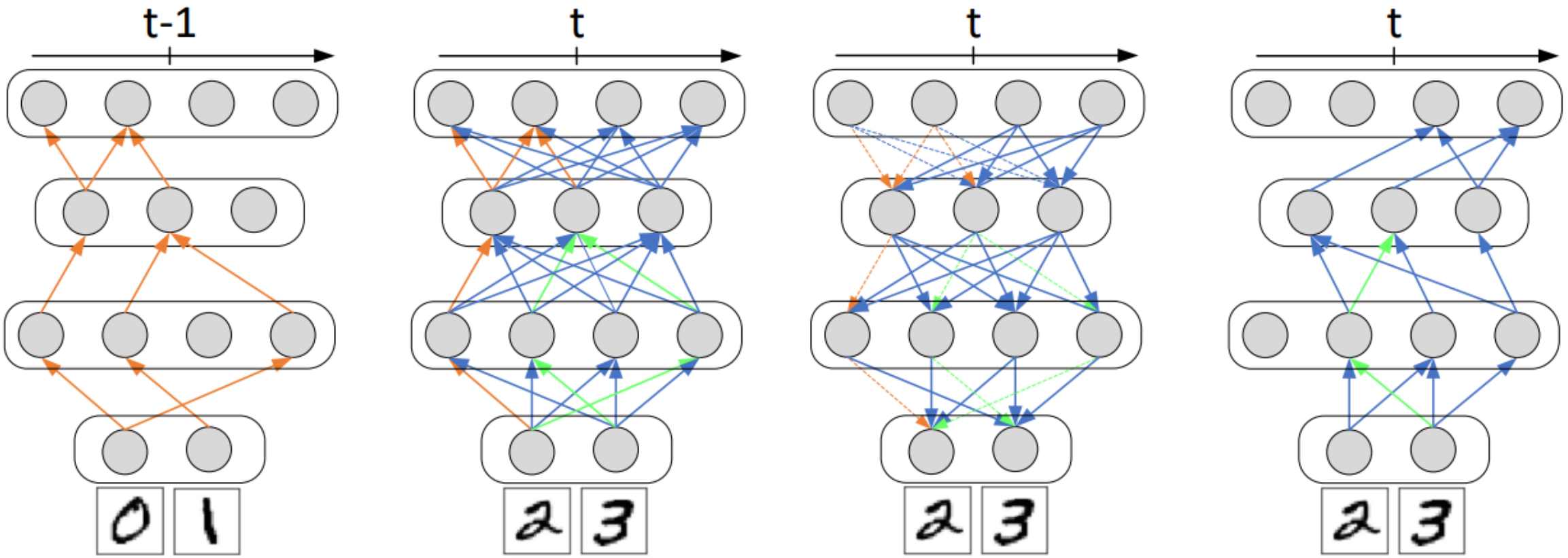
(b) Biased Transfer

(c) Selective Reuse Expansion beyond Dense Network

(d) Selective Reuse Expansion within Network (Our WSN)

Figure 1. **Concept Comparison:** (a) Piggyback (Mallya et al., 2018), and SupSup (Wortsman et al., 2020) find the optimal binary mask on a fixed backbone network a given task (b) PackNet (Mallya & Lazebnik, 2018) and CLNP (Golkar et al., 2019) forces the model to reuse all features and weights from previous subnetworks which causes bias in the transfer of knowledge (c) APD (Yoon et al., 2020) selectively reuse and dynamically expand the dense network (d) Our WSN selectively reuse and dynamically expand subnetworks within a dense network. Green edges are reused weights

Method



(a) selected weights $\hat{\boldsymbol{\theta}}_{t-1}$ at prior task

(b) forward pass using reused weights

(c) backward pass only on non-used weights

(d) selected weights $\hat{\boldsymbol{\theta}}_t$ with subsets of reused weights

Figure 2. **An illustration of Winning SubNetworks (WSN):** (a) The top-c% weights $\hat{\boldsymbol{\theta}}_{t-1}$ at prior task are obtained, (b) In the forward pass of a new task, WSN reuses weights selected from prior tasks, (c) In the backward pass, WSN updates only non-used weights, and (d) after several iterations of (b) and (c), we acquire again the top-c% weights $\hat{\boldsymbol{\theta}}_t$ including subsets of reused weights. for the new task.

Overview & Details

Algorithm 1 Winning SubNetworks (WSN)

input $\{\mathcal{D}_t\}_{t=1}^{\mathcal{T}}$, model weights $\boldsymbol{\theta}$, score weights $\mathbf{s}$, binary mask $\mathbf{M}_0 = \mathbf{0}^{|\boldsymbol{\theta}|}$, layer-wise capacity c

Randomly initialize $\boldsymbol{\theta}$ and $\mathbf{s}$.
for task $t = 1, ..., \mathcal{T}$ **do**
  for batch $\mathbf{b}_t \sim \mathcal{D}_t$ **do**
    Obtain mask $\mathbf{m}_t$ of the top-$c\%$ scores $\mathbf{s}$ at each layer
    Compute $\mathcal{L}\left(\boldsymbol{\theta} \odot \mathbf{m}_t; \mathbf{b}_t\right)$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} \odot (\mathbf{1} - \mathbf{M}_{t-1})\right)$ ▷ Weight update
    $\mathbf{s} \leftarrow \mathbf{s} - \eta(\frac{\partial \mathcal{L}}{\partial \mathbf{s}})$ ▷ Weight score update
  end for
  $\mathbf{M}_t \leftarrow \mathbf{M}_{t-1} \vee \mathbf{m}_t$ ▷ Accumulate binary mask
end for

- Freeze parameters selected by pervious steps.
- Compress binary mask M with **Huffman encoder**.

```
output = model(data)
loss = criterion(output, target)

loss.backward()

getattr(getattr(model, module_name),
module_attr).grad[consolidated_masks[ke
y] == 1] = 0

optimizer.step()
```

Experiments

Method	Permuted MNIST			5 Datasets			Omniglot Rotation		
	ACC (%)	CAP (%)	BWT	ACC (%)	CAP (%)	BWT	ACC (%)	CAP (%)	BWT
STL	97.37 (± 0.01)	1,000.0	-	93.44 (± 0.12)	500.0	-	82.13 (± 0.08)*	10,000.0	-
FINETUNE	78.22 (± 0.84)	100.0	-0.21 (± 0.01)	80.06 (± 0.74)	100.0	-0.17 (± 0.01)	44.48 (± 1.68)	100.0	-0.45 (± 0.02)
EWC (Kirkpatrick et al., 2017)	92.01 (± 0.56)	100.0	-0.03 (± 0.00)	88.64 (± 0.26)†	100.0†	-0.04 (± 0.01)†	68.66 (± 1.92)*	100.0*	-
HAT (Serrà et al., 2018)	-	-	-	91.32 (± 0.18)†	100.0†	-0.03 (± 0.00)†	-	-	-
GPM (Saha et al., 2021)	94.96 (± 0.07)	100.0	-0.02 (± 0.01)	91.22 (± 0.20)†	100.0	-0.01 (± 0.00)†	85.24 (± 0.37)	100.0	-0.01 (± 0.00)
PackNet (Mallya & Lazebnik, 2018)	96.37 (± 0.04)	96.38	**0.0**	92.81 (± 0.12)	82.86	**0.0**	30.70 (± 1.50)	399.2	**0.0**
SupSup (Wortsman et al., 2020)	96.31 (± 0.09)	122.89 (± 0.07)	**0.0**	93.28 (± 0.21)	104.27 (± 0.21)	**0.0**	58.14 (± 2.42)	407.12 (± 0.17)	**0.0**
WSN, $c = 0.03$	94.84 (± 0.11)	**19.87 (± 0.16)**	**0.0**	90.57 (± 0.65)	**12.11 (± 0.06)**	**0.0**	80.68 (± 2.60)	**75.87 (± 1.24)**	**0.0**
WSN, $c = 0.05$	95.65 (± 0.03)	26.49 (± 0.16)	**0.0**	91.61 (± 0.21)	17.26 (± 0.25)	**0.0**	**87.28 (± 0.72)**	79.85 (± 1.19)	**0.0**
WSN, $c = 0.1$	96.14 (± 0.03)	40.41 (± 0.54)	**0.0**	92.67 (± 0.12)	28.01 (± 0.28)	**0.0**	83.10 (± 1.56)	83.08 (± 1.61)	**0.0**
WSN, $c = 0.3$	**96.41 (± 0.07)**	77.73 (± 0.36)	**0.0**	93.22 (± 0.32)	62.30 (± 0.69)	**0.0**	81.89 (± 1.15)	102.2 (± 0.89)	**0.0**
WSN, $c = 0.5$	96.24 (± 0.11)	98.10 (± 0.25)	**0.0**	**93.41 (± 0.13)**	86.10 (± 0.57)	**0.0**	79.80 (± 2.16)	121.2 (± 0.50)	**0.0**
MTL	96.70 (± 0.02)†	100.0	-	91.54 (± 0.28)†	100.0	-	81.23 (± 0.52)	100.0	-

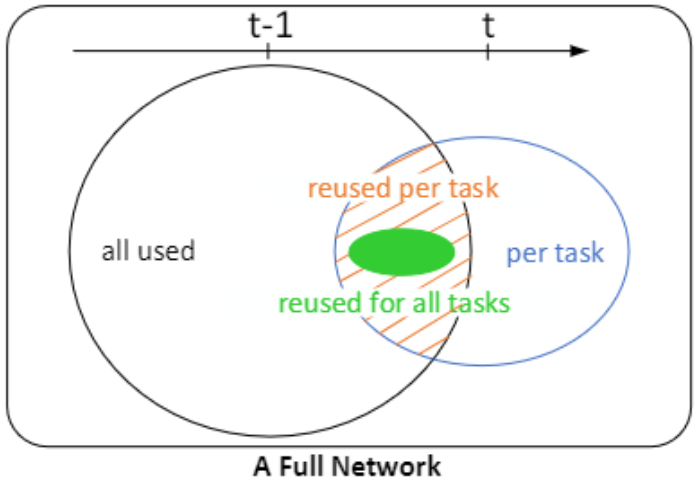
- ACC: accuracy
- CAP: capacity of parameters(%)
- BWT: the ACC drop during continual learning(i.e., forgetting)

Experiments

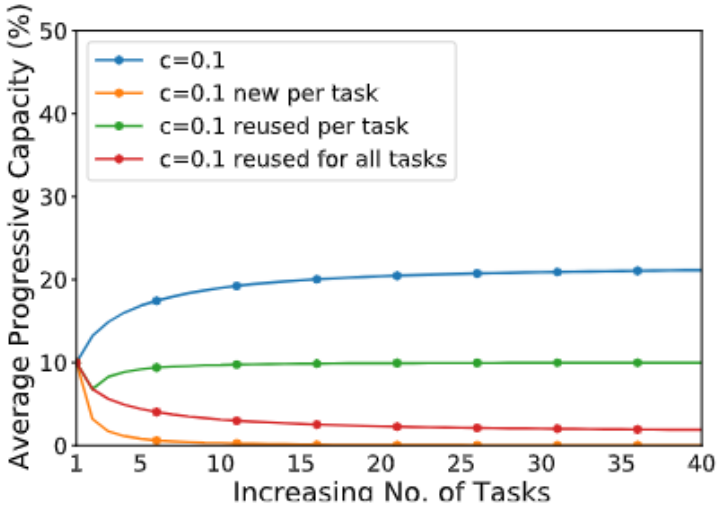
Table 2. Performance comparisons of the proposed method and other state-of-the-art including baselines - PackNet (Mallya & Lazebnik, 2018) and SupSup (Wortsman et al., 2020) - on various benchmark datasets. We report the mean and standard deviation of the average accuracy (ACC), average capacity (CAP), and average backward transfer (BWT) across 5 independent runs with 5 seeds under the same experimental setup (Deng et al., 2021). The best results are highlighted in bold. Also, † denotes results reported from Deng et al. (2021).

Method	**CIFAR-100 Split**			**CIFAR-100 Superclass**			**TinyImageNet**		
	ACC (%)	CAP (%)	BWT (%)	ACC (%)	CAP (%)	BWT (%)	ACC (%)	CAP (%)	BWT (%)
La-MaML (Gupta et al., 2020)	71.37 (± 0.67)†	100.0	-5.39 (± 0.53)†	54.44 (± 1.36)†	100.0	-6.65 (± 0.85)†	66.90 (± 1.65)†	100.0	-9.13 (± 0.90)†
GPM (Saha et al., 2021)	73.18 (± 0.52)†	100.0	-1.17 (± 0.27)†	57.33 (± 0.37)†	100.0	-0.37 (± 0.12)†	67.39 (± 0.47)†	100.0	**1.45** (± **0.22**)†
FS-DGPM (Deng et al., 2021)	74.33 (± 0.31)†	100.0	-2.71 (± 0.17)†	58.81 (± 0.34)†	100.0	-2.97 (± 0.35)†	70.41 (± 1.30)†	100.0	-2.11 (± 0.84)†
PackNet (Mallya & Lazebnik, 2018)	72.39 (± 0.37)	96.38 (± 0.00)	**0.0**	58.78 (± 0.52)	126.65 (± 0.00)	**0.0**	55.46 (± 1.22)	188.67 (± 0.00)	**0.0**
SupSup (Wortsman et al., 2020)	75.47 (± 0.30)	129.00 (± 0.03)	**0.0**	61.70 (± 0.31)	162.49 (± 0.00)	**0.0**	59.60 (± 1.05)	214.52 (± 0.89)	**0.0**
WSN, $c = 0.03$	70.65 (± 0.36)	**18.56** (± **0.25**)	**0.0**	54.99 (± 0.71)	**22.30** (± **0.22**)	**0.0**	68.72 (± 1.63)	**37.19** (± **0.21**)	**0.0**
WSN, $c = 0.05$	72.44 (± 0.27)	25.09 (± 0.42)	**0.0**	57.99 (± 1.34)	27.37 (± 0.33)	**0.0**	71.22 (± 0.94)	41.98 (± 0.52)	**0.0**
WSN, $c = 0.1$	74.55 (± 0.47)	39.87 (± 0.62)	**0.0**	60.45 (± 0.37)	38.55 (± 0.20)	**0.0**	**71.96** (± **1.41**)	48.65 (± 3.03)	**0.0**
WSN, $c = 0.3$	75.98 (± 0.68)	80.26 (± 1.53)	**0.0**	61.47 (± 0.30)	63.47 (± 1.33)	**0.0**	70.92 (± 1.37)	73.44 (± 2.35)	**0.0**
WSN, $c = 0.5$	**76.38** (± **0.34**)	99.13 (± 0.48)	**0.0**	**61.79** (± **0.23**)	80.93 (± 1.58)	**0.0**	69.06 (± 0.82)	92.03 (± 1.80)	**0.0**
Multitask	79.75 (± 0.38)†	100.0	-	61.00 (± 0.20)†	100.0	-	77.10 (± 1.06)†	100.0	-

Experiment

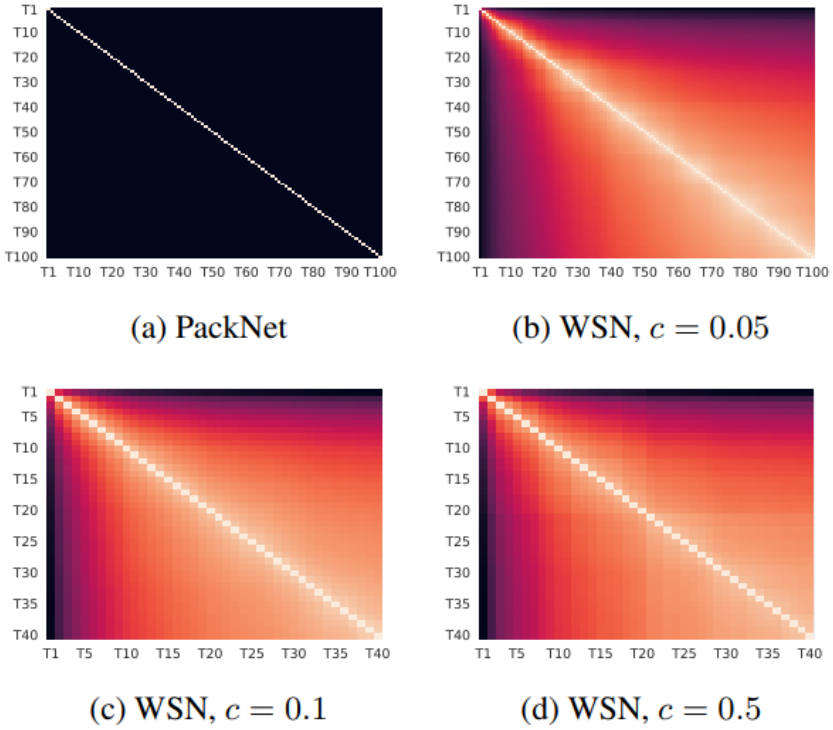


(a) A Diagram of Capacities



(b) Capacities over c

- "reused for all tasks" keeps in a low level, while "reused per tasks" is more.
- WSN tends uses diverse weights to solve the sequential tasks within all used weights.



(a) PackNet (b) WSN, $c = 0.05$

(c) WSN, $c = 0.1$ (d) WSN, $c = 0.5$

Paramenter Correlations between tasks.