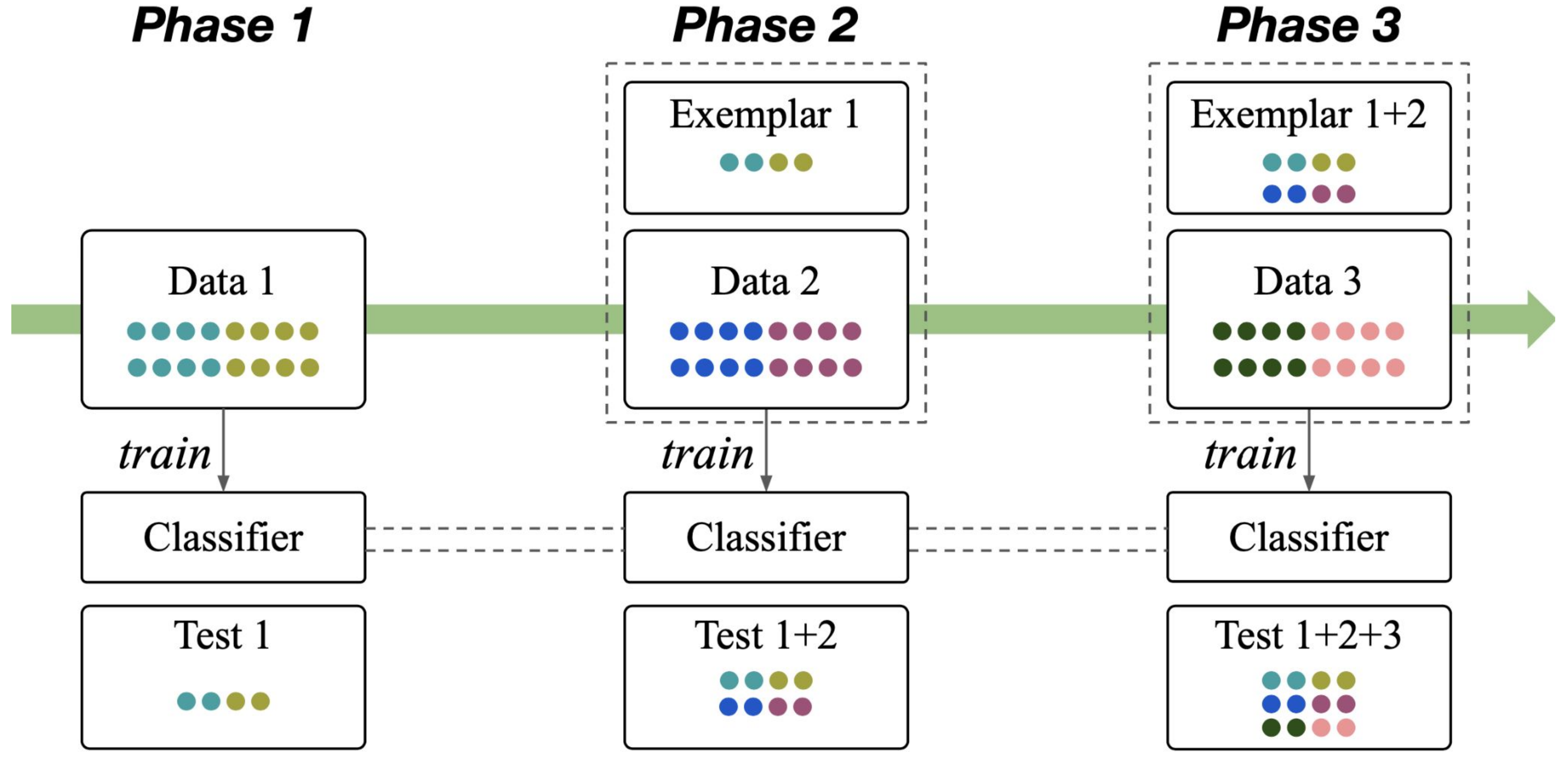
Online Hyperparameter Optimization for Class-Incremental Learning

Yaoyao Liu[1] Yingying Li[2] Bernt Schiele[1] Qianru Sun[3]

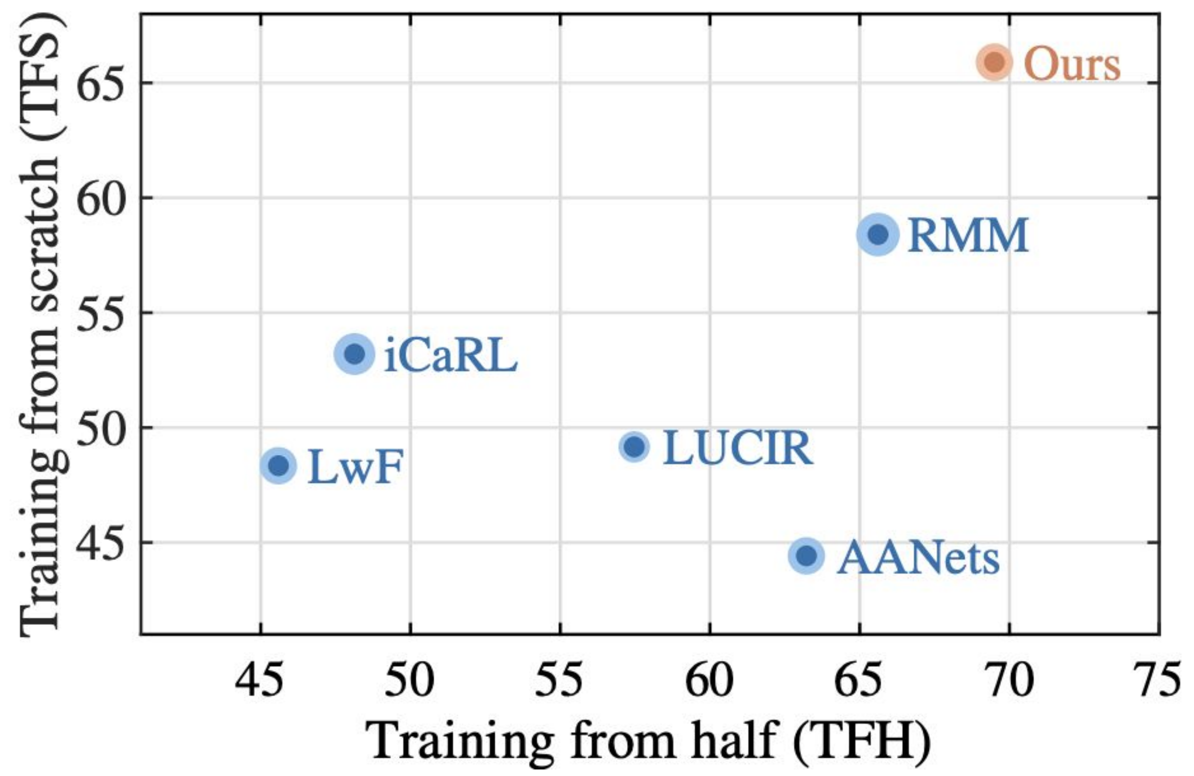
[1]Max Planck Institute for Informatics [2]California Institute of Technology [3]Singapore Management University

Task & Challenge & Contributions

- **Task: Class-Incremental Learning[1]**
  - Different classes arrive in different phases;
  - At any time, it provides a classifier for the classes observed so far;
  - The memory is limited.

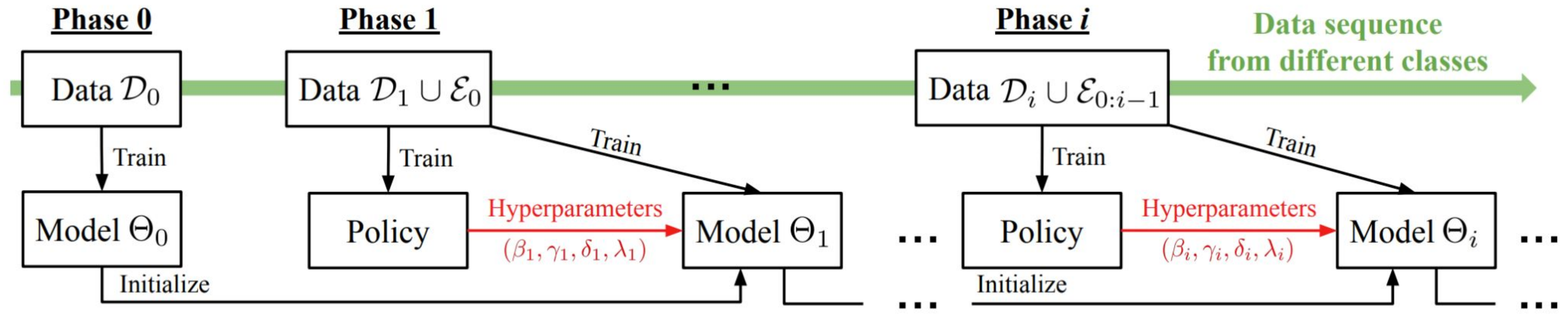


- **Challenge: the stability-plasticity trade-off** for different data-receiving settings
  - LUCIR, AANets, and RMM are more suited for TFH
  - iCaRL and LwF are more suited for TFS

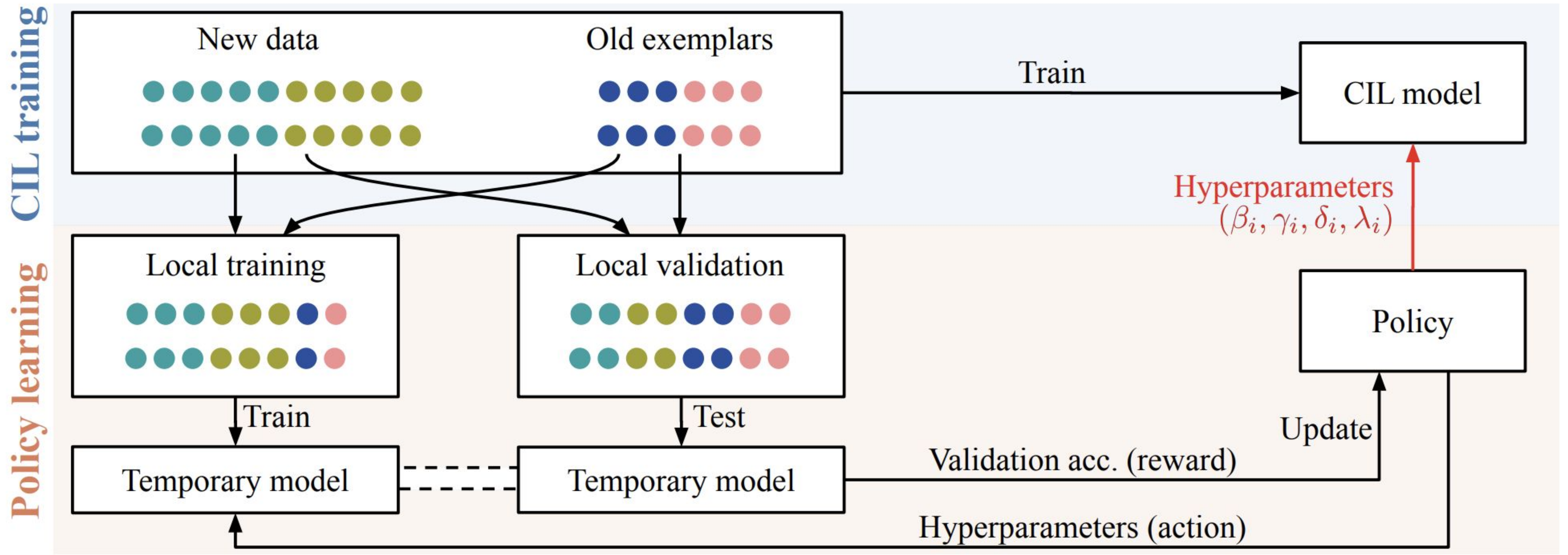


Framework & Optimization Steps

- **The computing flow of our method across all phases**



- **The training process of our method in Phase *i***



- **Algorithms**

Algorithm 1: Our CIL algorithm in Phase i ($i \geq 1$)

Input : Old model Θ_{i-1}, training data $\mathcal{E}_{0:i-1} \cup \mathcal{D}_i$, validation data $\mathcal{Q}_{0:i}$, learnable parameters $\mathbf{w}$, numbers of epochs M_1 and M_2.

Output: New model Θ_i, new exemplars $\mathcal{E}_{0:i}$, learnable parameters $\mathbf{w}$.

```
// Policy learning
if i=1 then
    Initialize w = {1, ..., 1};
for t in i, ..., T do
    Randomly sample a class-balanced subset B_{0:i} from E_{0:i-1} ∪ D_i,;
    Create the local environment h_i = ((E_{0:i-1} ∪ D_i) \ B_{0:i}, B_{0:i});
    Set the policy p = w/||w||;
    Sample an action a_t ~ p;
    for j in i, ..., i+n do
        Train Θ_j for M_1 epochs by Algorithm 2 with inputs Θ_{j-1}, a_t, h_i;
        Collect the reward r_{h_i}(Θ_j, a_t);
    Compute the cumulative reward R̂(a_t, h_i) by Eq. 6;
    Update w by Eq. 8;
// CIL training
Sample an action a_i ~ p;
Train Θ_i for M_2 epochs by Algorithm 2 with inputs Θ_{i-1}, a_i, H_i = (E_{0:i-1} ∪ D_i, Q_{0:i});
Select new exemplars E_{0:i} from E_{0:i-1} ∪ D_i using herding [30].
```

Algorithm 2: Training and evaluation for action $\mathbf{a}$

Input : Old model Θ_{old}, action $\mathbf{a} = \{\beta, \gamma, \delta, \lambda\}$, environment $h = \{\mathcal{T}, \mathcal{Q}\}$.

Output: New model Θ, reward $r_h(\Theta, \mathbf{a})$ (i.e., the validation accuracy).

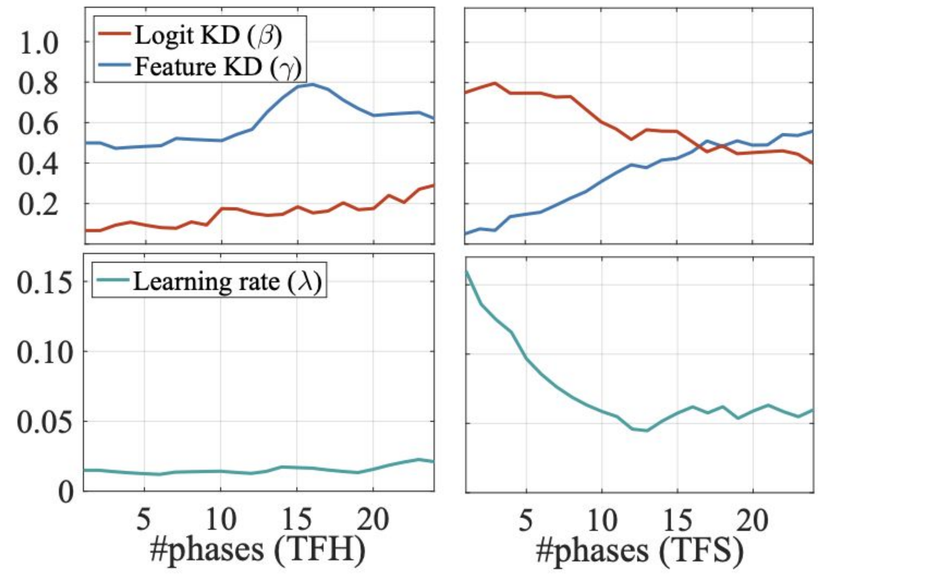
```
Initialize Θ with Θ_old;
for epochs do
    for mini-batch (x, y) ∈ T do
        Compute the loss L_overall(x, y; Θ_old, β, γ) by Eq. 3;
        Update Θ by Eq. 4 with λ;
for mini-batch (x_val, y_val) ∈ Q do
    Compute predictions μ(x_val; Θ) by Eq. 5 with δ;
Compute the reward r_h(Θ, a) using {(μ(x_val), y_val)}_{(x_val, y_val) ∈ Q}.
```

Experiment Results

- **Ablation study**

No.	Optimizing (β, γ)	Optimizing δ	Optimizing λ	N=5 TFH	N=5 TFS	N=25 TFH	N=25 TFS
1	Baseline			63.11	62.96	57.47	49.16
2	✓			63.20	63.60	58.27	50.91
3	✓	✓		63.23	64.08	58.20	51.94
4	✓	✓	✓	**63.88**	**64.92**	**59.27**	**52.44**
5	Cross-val fixed			63.33	64.02	57.50	51.64
6	Offline RL			63.42	63.88	58.12	51.53
7	Bilevel HO			63.20	63.02	57.56	49.42

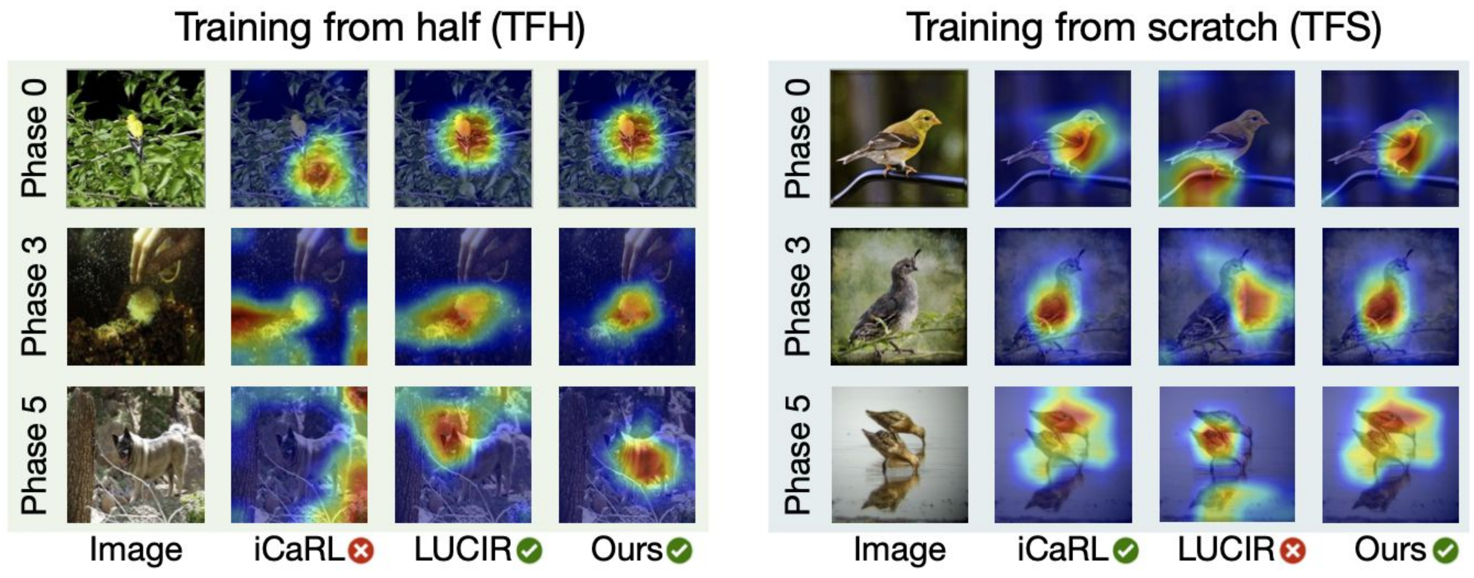
- **The hyperparameter values**



- **Comparing w/ SOTA**

Methods	CIFAR-100, N=5 TFH	CIFAR-100, N=5 TFS	CIFAR-100, N=5 Avg.	CIFAR-100, N=25 TFH	CIFAR-100, N=25 TFS	CIFAR-100, N=25 Avg.	ImageNet-Subset, N=5 TFH	ImageNet-Subset, N=5 TFS	ImageNet-Subset, N=5 Avg.	ImageNet-Subset, N=25 TFH	ImageNet-Subset, N=25 TFS	ImageNet-Subset, N=25 Avg.
iCaRL [30]	58.1	64.0	61.0	48.1	53.2	50.7	65.3	70.4	67.9	53.0	53.5	53.3
PODNet [8]	64.7	63.6	64.2	60.3	45.3	52.8	64.3	58.9	61.6	68.3	39.1	53.7
DER [40]	67.6	72.3	70.0	65.5	67.3	66.4	78.4	76.9	77.7	75.4	71.0	73.2
FOSTER [38]	70.4	72.5	71.5	63.8	**70.7**	67.3	80.2	78.3	79.3	69.3	72.9	71.1
LUCIR [14]	$63.1_{\pm0.7}$	$63.0_{\pm0.6}$	$63.1_{\pm0.7}$	$57.5_{\pm0.4}$	$49.2_{\pm0.5}$	$53.4_{\pm0.5}$	$65.3_{\pm0.6}$	$66.7_{\pm0.5}$	$66.0_{\pm0.6}$	$61.4_{\pm0.7}$	$46.2_{\pm0.8}$	$53.8_{\pm0.8}$
w/ ours	$63.9_{\pm0.6}$ ↑0.8	$64.9_{\pm0.5}$ ↑1.9	$64.4_{\pm0.6}$ ↑1.3	$59.3_{\pm0.5}$ ↑1.8	$52.4_{\pm0.5}$ ↑3.2	$55.9_{\pm0.5}$ ↑2.5	$70.6_{\pm0.7}$ ↑5.3	$68.4_{\pm0.6}$ ↑1.7	$69.5_{\pm0.7}$ ↑3.5	$62.9_{\pm0.6}$ ↑1.5	$54.1_{\pm0.6}$ ↑7.9	$58.5_{\pm0.6}$ ↑4.7
AANets [22]	$65.3_{\pm0.4}$	$63.1_{\pm0.3}$	$64.2_{\pm0.4}$	$63.2_{\pm0.3}$	$44.4_{\pm0.4}$	$53.8_{\pm0.4}$	$77.0_{\pm0.7}$	$68.9_{\pm0.6}$	$73.0_{\pm0.7}$	$72.2_{\pm0.6}$	$60.7_{\pm0.5}$	$66.5_{\pm0.6}$
w/ ours	$67.0_{\pm0.3}$ ↑1.7	$65.1_{\pm0.3}$ ↑2.0	$66.1_{\pm0.3}$ ↑1.9	$64.1_{\pm0.4}$ ↑0.9	$50.3_{\pm0.5}$ ↑5.9	$57.2_{\pm0.5}$ ↑3.4	$77.3_{\pm0.6}$ ↑0.3	$70.6_{\pm0.5}$ ↑1.7	$74.0_{\pm0.6}$ ↑1.0	$72.9_{\pm0.5}$ ↑0.7	$64.8_{\pm0.5}$ ↑4.1	$68.9_{\pm0.5}$ ↑2.4
RMM [23]	$67.6_{\pm0.7}$	$70.4_{\pm0.8}$	$69.0_{\pm0.8}$	$65.6_{\pm0.6}$	$58.4_{\pm0.6}$	$62.0_{\pm0.6}$	$79.5_{\pm0.2}$	$80.5_{\pm0.3}$	$80.0_{\pm0.3}$	$75.0_{\pm0.3}$	$71.6_{\pm0.3}$	$73.3_{\pm0.3}$
w/ ours	$\mathbf{70.8}_{\pm0.7}$ ↑3.2	$\mathbf{72.7}_{\pm0.6}$ ↑2.3	$\mathbf{71.8}_{\pm0.7}$ ↑2.8	$\mathbf{69.5}_{\pm0.8}$ ↑3.9	$65.9_{\pm0.7}$ ↑7.5	$\mathbf{67.7}_{\pm0.8}$ ↑5.7	$\mathbf{81.0}_{\pm0.3}$ ↑1.5	$\mathbf{82.2}_{\pm0.4}$ ↑1.7	$\mathbf{81.6}_{\pm0.4}$ ↑1.6	$\mathbf{76.1}_{\pm0.2}$ ↑1.1	$\mathbf{73.2}_{\pm0.4}$ ↑1.6	$\mathbf{74.7}_{\pm0.3}$ ↑1.4

- **Grad-CAM Visualizations**



References

[1] Rebuffi, Sylvestre-Alvise, et al. "icarl: Incremental classifier and representation learning." CVPR 2017.