Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

b-it Bonn-Aachen International Center for Information Technology

Lifelong Action Learning for Socially Assistive Robots

November 28th, 2022

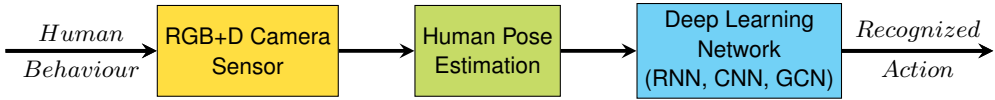
Hasnainali Walli

Advisors

Prof. Dr. Paul G. Plöger, Prof. Dr. Sebastian Houben, Alex Mitrevski

Motivation

- Action recognition is a key function for socially assistive robots
- **Challenge**: Conventional models' inability to learn new actions
- **How can robotic systems learn new actions without forgetting?**



Lifelong Action Learning

- Robotic systems fine-tune their knowledge with experience
- New actions are learnt while retaining the knowledge of the previous actions
- Concept of lifelong action learning was explored in the context of CRI

- **Objectives**:
  - Develop an action learning model using incremental learning
  - Integrate model on QTRobot for the MigrAVE project

Lifelong Action Learning

- Efthymiou et al. work was first to look at IL with action recognition for CRI
- Edutainment system for scenarios such as classroom settings

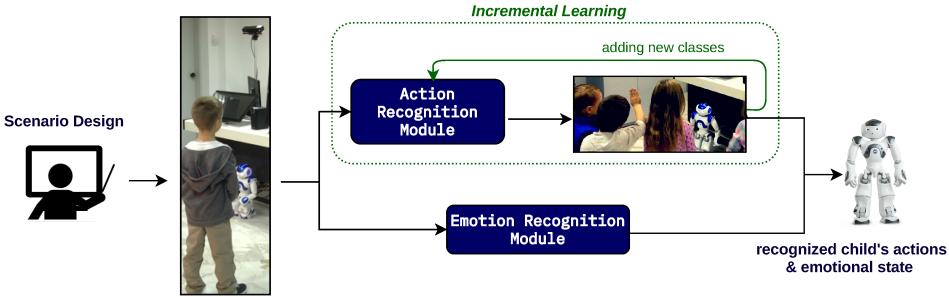


Figure 1: Incremental learning pipeline for action and emotion recognition[1]

[1] N. Efthymiou, P. P. Filntisis, G. Potamianos, and P. Maragos, "Visual Robotic Perception System with Incremental Learning for Child–Robot Interaction Scenarios," Technologies, vol. 9, no. 86, November 2021.

Lifelong Action Learning

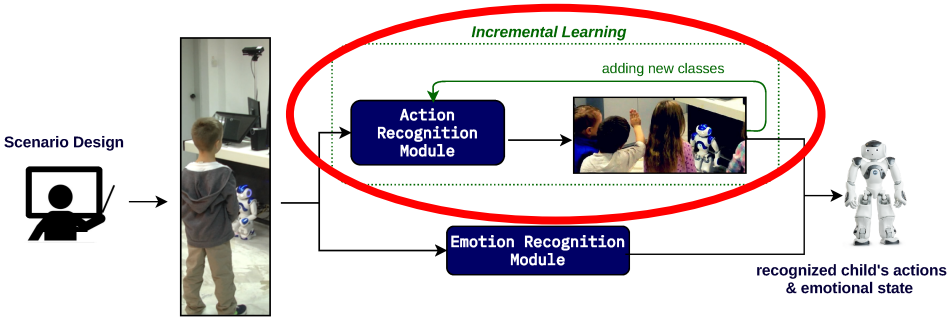


Figure 2: Incremental learning pipeline for action and emotion recognition[1]

[1] N. Efthymiou, P. P. Filntisis, G. Potamianos, and P. Maragos, "Visual Robotic Perception System with Incremental Learning for Child–Robot Interaction Scenarios," Technologies, vol. 9, no. 86, November 2021.

Lifelong Action Learning

Their Approach

- TSN Network
- iCaRL Algorithm
- RGB+D and Optical Flow data
- BabyRobot Dataset

Our Approach

- CTR-GCN Network
- BiC Algorithm
- 3D Skeleton data
- NTU RGB+D Dataset

Our Approach

Methodology

1. Performed a comparative analysis on skeleton-based action recognition networks
2. Performed a comparative analysis on class-incremental learning algorithms
3. Integrated the final model on QTRobot

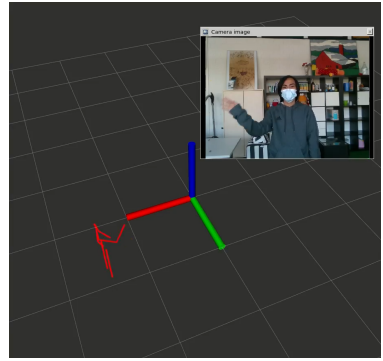


Figure 3: Hand waving action visualized in RVIZ

NTU RGB+D Dataset

- Features 120 everyday actions
- 40 subjects; 3 cameras; 2 demos
- 25 skeletal joints tracked
- Evaluation:
  - Cross-Subject Accuracy: train on 20 subjects; test on 20 subjects
  - Cross-View Accuracy: train using 2 views; test on 1 view

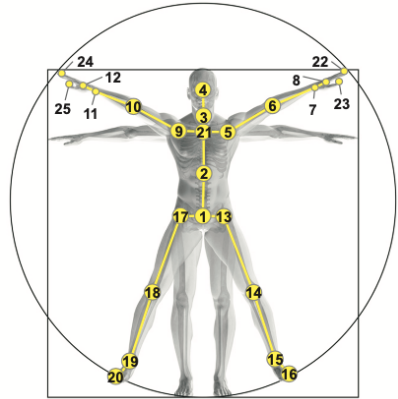


Figure 4: Joint configurations for NTU RGB-D dataset[2]

[2] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1010–1019.

Action Recognition Analysis

- Networks: CTR-GCN, MS-G3D, EfficientGCN, ViewAdaptive NN
- Modalities: Joint, Bone and Joint Motion
- **Metrics**: Cross-Subject & Cross-View Accuracy & Training Time

Drink Water	Eat Meal	Brush Teeth	Drop
Pick Up	Throw	Sit Down	Stand Up
Clapping	Hand Waving	Kick Something	Hopping
Jump Up	Play with Phone	Point to Something	Rub Hands
Nod Head/Bow	Shake Head	Wipe Face	Cross Hands

Table 1: Subset of action classes from the NTU RGB-D dataset

Action Recognition Analysis Results

Network	Cross Subject	Cross View
CTR-GCN (Joint)	92.63%	96.37%
CTR-GCN (Bone)	92.78%	96.02%
CTR-GCN (Motion)	92.51%	96.40%
MS-G3D (Joint)	91.27%	96.85%
MS-G3D (Bone)	90.90%	95.44%
EfficientGCN-B4 (SG Layer)	94.05%	97.47%
EfficientGCN-B4 (EpSep Layer)	94.43%	97.56%
VA-NN (CNN)	92.97%	92.20%
VA-NN (RNN)	-	-

Table 2: Action recognition networks accuracy

Network	Training Time
CTR-GCN	4 hrs
MS-G3D	8 hrs
EfficientGCN-B4	5 hrs
VA-NN	0.5 hrs

Table 3: Networks training time

Action Recognition Analysis Results

Action	CTR-GCN	MS-G3D	Action	CTR-GCN	MS-G3D
Drink Water	82.48%	83.94%	Kick Something	97.83%	94.93%
Eat Meal	78.91%	73.82%	Hopping	98.91%	95.27%
Brush Teeth	90.84%	91.21%	Jump Up	98.91%	98.55%
Drop	90.18%	91.64%	Play with Phone	86.91%	90.91%
Pick Up	98.91%	94.55%	Point to Something	92.39%	92.03%
Throw	96.36%	90.91%	Rub Hands	90.58%	89.49%
Sit Down	98.90%	97.80%	Nod Head/Bow	96.01%	95.65%
Stand Up	98.17%	98.90%	Shake Head	96.00%	95.64%
Clapping	82.42%	72.89%	Wipe Face	92.39%	94.20%
Hand Waving	94.16%	94.89%	Cross Hands	93.84%	94.57%

Table 4: Cross-Subject accuracy results per class for CTR-GCN and MS-G3D models

Class-Incremental Learning

Class-Incremental Learning Problem

An algorithm attempts to learn a given sequence of tasks, T:

$$T = [(C^1, D^1), (C^2, D^2), \ldots, (C^n, D^n)] \quad (1)$$

Tasks

- Set of actions to be learnt:

$$D^t = \{(x_1, y_1), \ldots, (x_{m^t}, y_{m^t})\} \quad (2)$$

- Action set is distinct per task

$$C^i \cap C^j = \oslash, \; if \; i \neq j \quad (3)$$

Exemplars

- Memory of training data from previous tasks
- Augments training data if $t > 1$
- Memory scenarios: fixed or growing
- Selection methods: random, herding, distance, entropy

Class-Incremental Learning Metrics

Task-Aware Accuracy

Calculated with the knowledge of the action classes learnt within each task.

Task-Agnostic Accuracy

Calculated with the overall set of the action classes learnt.

Forgetting Percentage

Estimated percentage of data forgotten

$$F_{i,t} = max(A[i, 0 : t-1]) - A[i, t],\ i \leq t \quad (4)$$

where i = evaluated task & t = current task

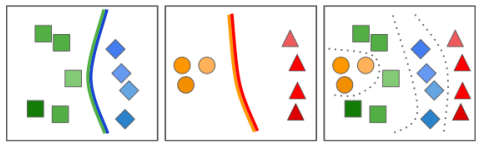


Figure 5: Incremental learning process of 4 classes split into 2 tasks[3]

[3] M. Masana et al., "Class-Incremental Learning: Survey and Performance Evaluation," CoRR, vol. abs/2010.15277, October 2020.

Class-Incremental Learning Analysis

- IL Algorithms: LwF[4], iCaRL[5], LUCIR[6], BiC[7]
- Memory Size: 400 (Fixed) & 20 per class (Growing)
- Selection Method: Random & Herding
- **Metrics**: Task-Aware & Task-Agnostic Accuracy

Task #	Action	Task #	Action
Task 1	Wipe Face Eat Meal	Task 6	Throw Point to Something
Task 2	Cross Hands Clapping	Task 7	Hand Waving Stand Up
Task 3	Kick Something Shake Head	Task 8	Nod Head/Bow Hopping
Task 4	Sit Down Play with Phone	Task 9	Drop Drink Water
Task 5	Pick Up Brush Teeth	Task 10	Rub Hands Jump Up

Table 5: Task sequence for class-IL comparative analysis

[4] Z. Li and D. Hoiem, "Learning without Forgetting," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 40, no. 12, pp. 2935–2947, Dec 2018.

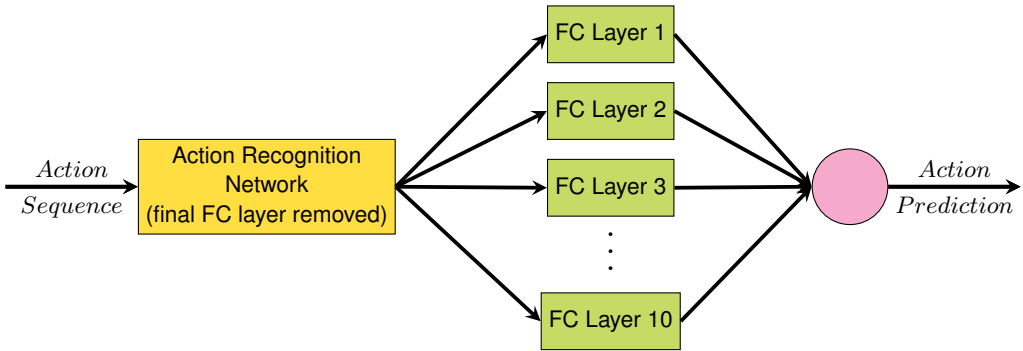
[5] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental Classifier and Representation Learning," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2017, pp. 5533–5542.

[6] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, "Learning a Unified Classifier Incrementally via Rebalancing," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2019, pp. 831–839.

[7] Y. Wu et al., "Large Scale Incremental Learning," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2019, pp. 374–382.

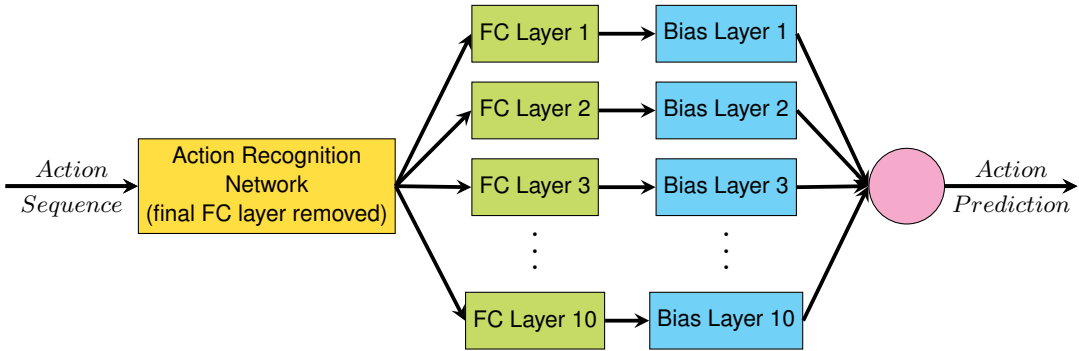
Class-Incremental Learning Analysis

LwF, iCaRL, LUCIR Model Architecture



Class-Incremental Learning Analysis

BiC Model Architecture



Class-Incremental Learning Analysis Results

Training Time

Algorithm	Memory Config	CTR-GCN			MS-G3D		
		Total Time (hrs)	Time per Task (min)	Time Incr per Task	Total Time (hrs)	Time per Task (min)	Time Incr per Task
iCaRL	Fixed	6.3	18.9 & 39.2	-	11.2	30.2 & 68.7	-
	Growing	5.6	-	2.8%	11.0	-	2.6%
LWF	Fixed	6.6	19.9 & 41.8	-	11.3	30.6 & 69.6	-
	Growing	5.8	-	2.8%	10.1	-	2.9%
BIC	Fixed	7.0	19.6 & 44.1	-	11.5	30.2 & 68.2	-
	Growing	6.2	-	3.3%	10.4	-	2.6%
LUCIR	Fixed	6.4	19.6 & 39.6	-	11.4	33.0 & 71.4	-
	Growing	5.8	-	2.9%	10.1	-	2.9%

Table 6: Incremental learning model training time

Class-Incremental Learning Analysis Results

Task-Aware Accuracy

		Herding		Random	
Algorithm	Memory Config	CTR-GCN	MS-G3D	CTR-GCN	MS-G3D
iCaRL	Fixed	94.9%	97.2%	95.3%	98.1%
	Growing	91.1%	95.9%	91.2%	96.6%
LWF	Fixed	98.4%	97.8%	98.2%	97.5%
	Growing	97.6%	96.1%	97.1%	96.1%
LUCIR	Fixed	95.5%	97.5%	97.3%	97.5%
	Growing	96.3%	94.8%	95.5%	96.5%
BiC	Fixed	98.2%	98.1%	97.9%	96.9%
	Growing	97.3%	96.9%	97.5%	96.6%

Table 7: Average task-aware accuracy

Class-Incremental Learning Analysis Results

Task-Agnostic Accuracy

		Herding		Random	
Algorithm	Memory Config	CTR-GCN	MS-G3D	CTR-GCN	MS-G3D
iCaRL	Fixed	52.1%	72.5%	50.9%	72.1%
	Growing	54.2%	67.8%	51.3%	66.9%
LWF	Fixed	73.8%	67.8%	75.4%	67.5%
	Growing	69.9%	61.5%	69.4%	63.5%
LUCIR	Fixed	66.2%	64.7%	68.4%	64.4%
	Growing	65.5%	43.6%	62.4%	59.9%
BiC	Fixed	79.1%	70.1%	74.8%	67.1%
	Growing	74.2%	61.1%	72.8%	61.6%

Table 8: Average task-agnostic accuracy

Class-Incremental Learning Analysis

How would the model perform if we grouped similar actions?

- Task-Aware Accuracy:
  - Fixed: 98.2% $\Longrightarrow$ 94.6%
  - Growing: 97.3% $\Longrightarrow$ 93.4%
- Task-Agnostic Accuracy:
  - Fixed: 79.1% $\Longrightarrow$ 77.6%
  - Growing: 74.2% $\Longrightarrow$ 72.7%

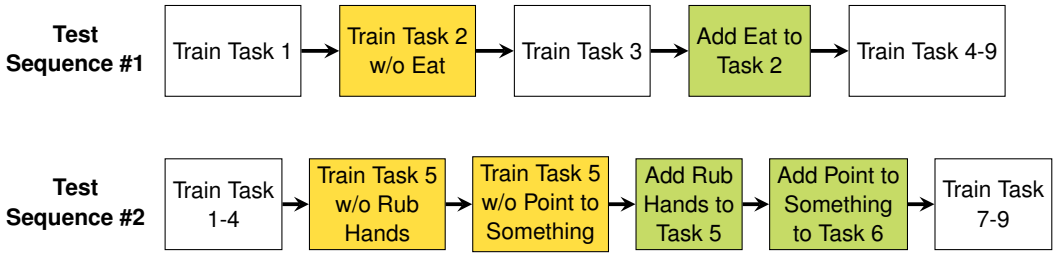
Task #	Action	Task #	Action
Task 1	Brush Teeth Wipe Face	Task 5	Clapping Rub Hands Cross Hands
Task 2	Drink Water Eat Meal	Task 6	Hand Waving Point to Something
Task 3	Drop Pick Up Throw	Task 7	Kick Something Hopping Jump Up
Task 4	Sit Down Stand Up	Task 8	Play with Phone
		Task 9	Nod Head/Bow Shake Head

Table 9: Task sequence with variable task size and sorting similar actions

Class-Incremental Learning Analysis

Model Robustness

- Task-Aware Accuracy:
  - Fixed: 94.6% $\Longrightarrow$ 94.1% & 96.7%
  - Growing: 93.4% $\Longrightarrow$ 95.1% & 95.6%
- Task-Agnostic Accuracy:
  - Fixed: 77.6% $\Longrightarrow$ 76.3% & 78.5%
  - Growing: 72.7% $\Longrightarrow$ 74.5% & 70.6%



Class-Incremental Learning Analysis

Memory Size

Growing						Fixed				
2	5	10	20	50	100	40	100	200	400	1000
78.3%	94.3%	97.2%	97.3%	98.5%	98.7%	94.5%	96.7%	97.6%	98.2%	98.5%

Table 10: Average task-aware accuracy for varying memory size

Growing						Fixed				
2	5	10	20	50	100	40	100	200	400	1000
11.3%	45.7%	63.8%	74.2%	79.8%	84.1%	35.8%	65.2%	70.3%	79.1%	80.0%

Table 11: Average task-agnostic accuracy for varying memory size

- Fixed memory is not feasable in real-life application (18%-20% time increase)
- Recording 50 or 100 demos of an action too onerous for performance benefit
- **Final Model**: CTR-GCN network, BiC algorithm, Growing memory (20/class), Herding selection

QTRobot Platform

- Teaching assistant for educators working with children
- Equipped with a RealSense 3D camera
- Skeletal tracking using Nuitrack SDK (19 joints tracked vs 25 joints in NTU)

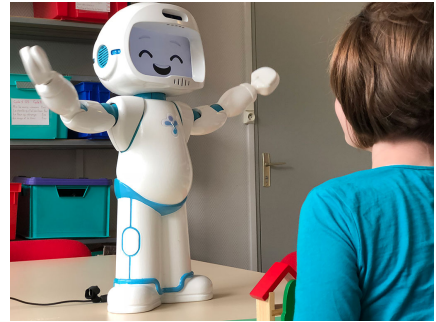


Figure 6: QTRobot interacting with a child[8]

[8] Image taken from: https://robots.ieee.org/robots/qtrobot/

Model Integration

- Task-Aware Accuracy: 97.3% $\Longrightarrow$ 97.4%
- Task-Agnostic Accuracy: 74.2% $\Longrightarrow$ 71.9%

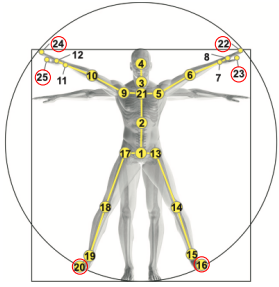


Figure 7: Joint configurations for NTU RGB-D dataset[9]

Figure 8: Joint configuration in the Nuitrack SDK[10]

[9] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1010–1019.

[10] Image taken from: https://github.com/3DiVi/nuitrack-sdk/tree/master/doc

CAL Server

- ROS module w/ FTSM arch
- Two main subcomponents:
  - ActionClassifier
  - ActionLearner
- Skeleton processor active after initialization
- Actions are classified with a 50 frame moving window
- Actions are learnt by capturing 25 demos

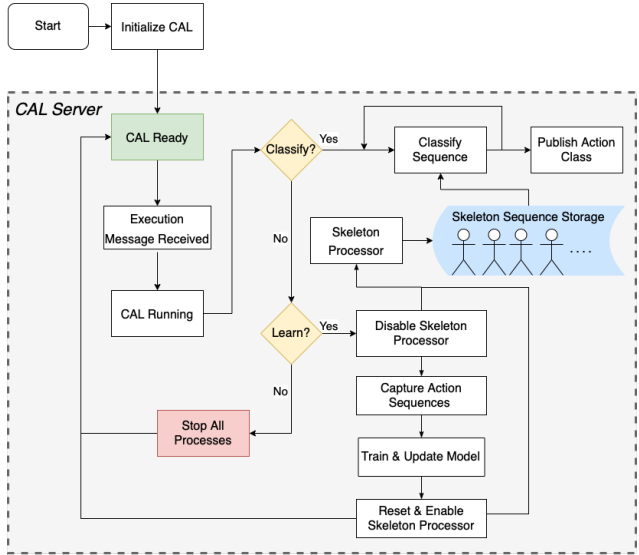


Figure 9: Workflow of the Continual Action Learning Server

Future Work

- Investigating optimal number of frames for effective recognition
- Generation of synthetic trajectories to reduce learning efforts
- Investigating the effectiveness of model in HRI study

HRI Study

Experimental Design

- Three-part experiment:
  1. The robot attempted to recognize users performing actions using the trained model
  2. The robot recorded users perform 4 actions (2 new and 2 old) 25 times
  3. The robot attempted to recognize users performing actions using the 3 experimental models

- **Performed Actions**: Play with Phone, Hopping, Rub Hands, Hand Waving, Shake Head, Drink Water
- **Learnt Actions**: Talking on Phone, Cutting Food, Hand Waving, Hopping

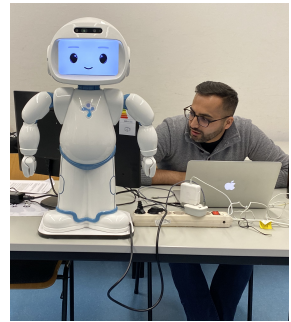
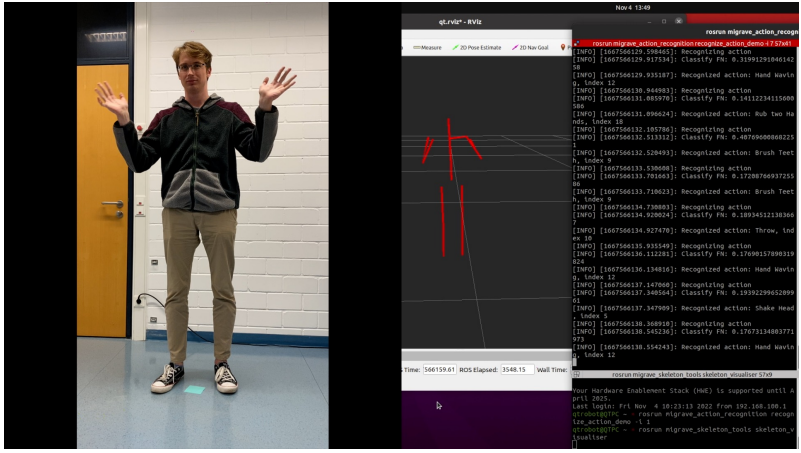


Figure 10: Robot and evaluator setup for HRI study

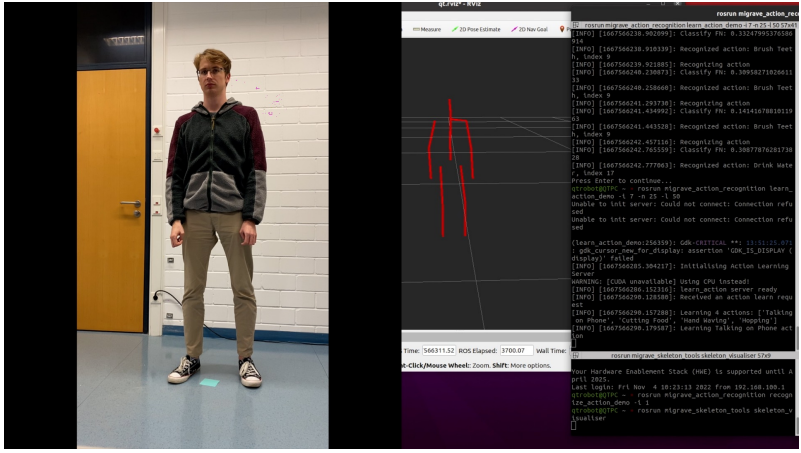
Demo

Action Recognition



Demo

Action Learning



HRI Study

Results

- 15 users participated
- Four models evaluated
  1. Original trained model
  2. Inclusion of 2 new actions as a new task
  3. Improvement of old actions with new data
  4. Modifying old task with new action

Model	Accuracy
Original	62.4%
Exp. #1	45.9%
Exp. #2	57.0%
Exp. #3	36.0%

Table 12: Accuracy per model

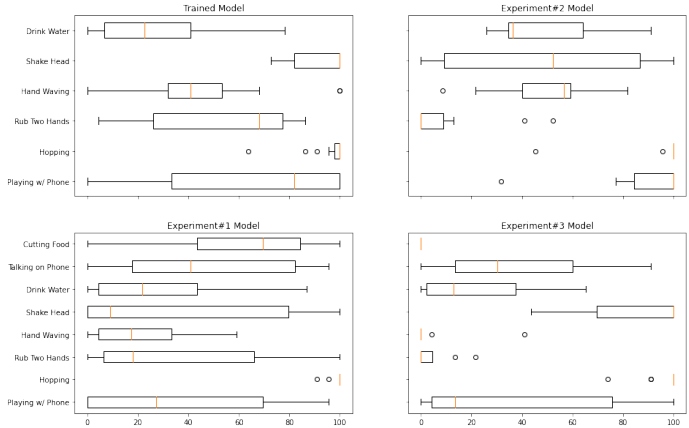


Figure 11: Per action results from HRI Study for each model

HRI Study

Results

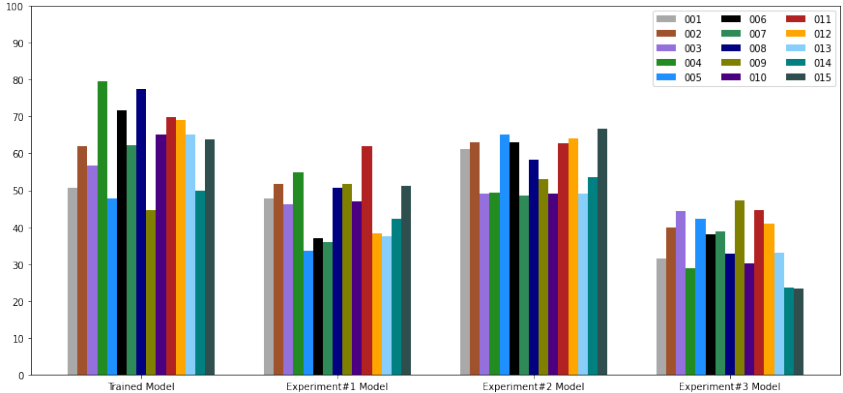


Figure 12: Per person results from HRI Study for each model

Thank You!

Questions?

Appendix

CTR-GCN Confusion Matrix

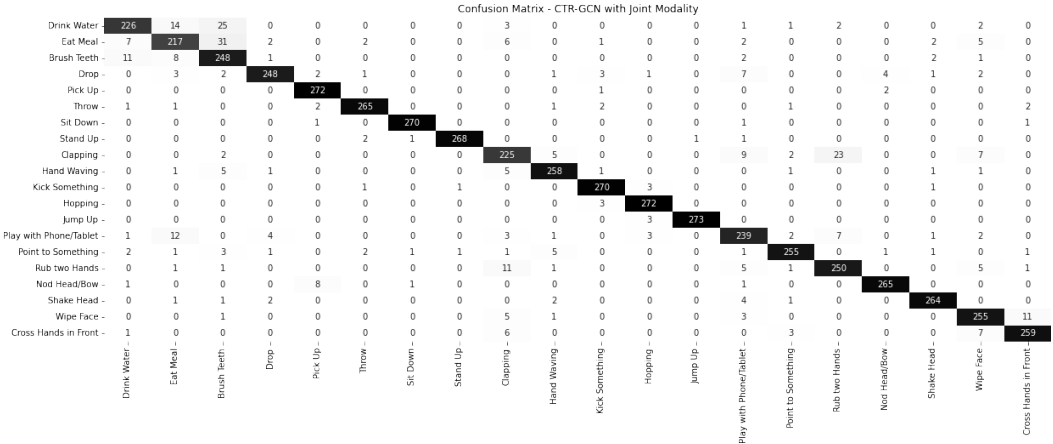


Figure 13: Cross-Subject confusion matrix of CTR-GCN network trained with joint data

CTR-GCN + BiC Confusion Matrix (NTU Joints)

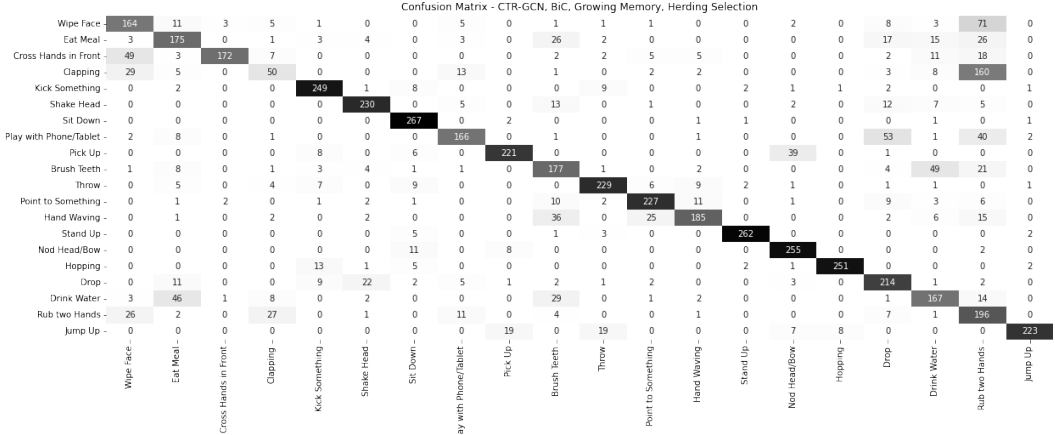


Figure 14: Cross-Subject confusion matrix of CTR-GCN network and BiC IL approach

CTR-GCN + BiC Confusion Matrix (NT Joints)

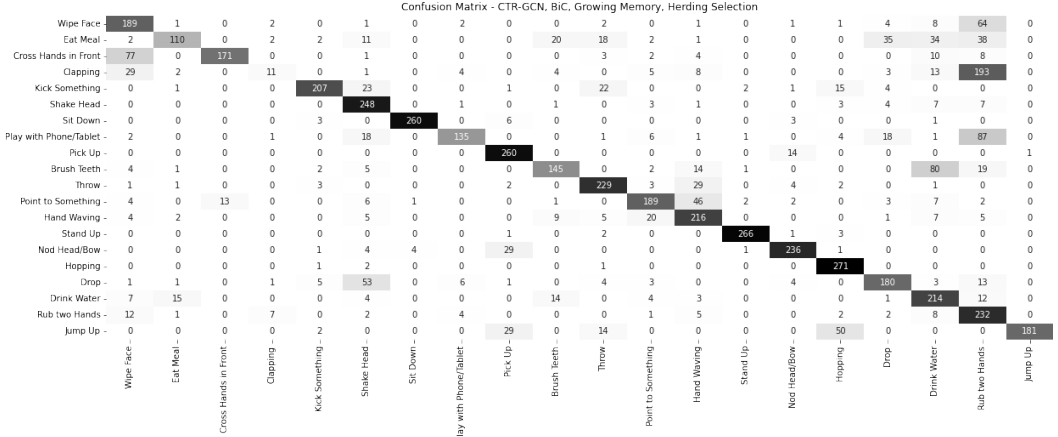


Figure 15: Cross-Subject confusion matrix of CTR-GCN network and BiC IL approach

CTR-GCN, Fixed Results

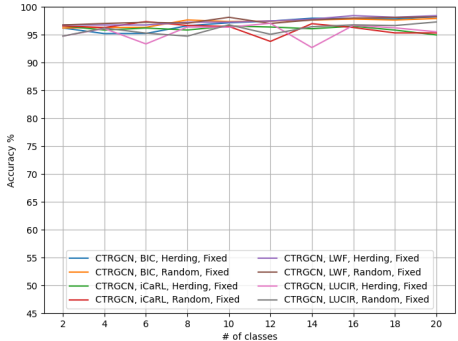


Figure 16: Comparative task-aware accuracy w/ CTR-GCN and fixed memory

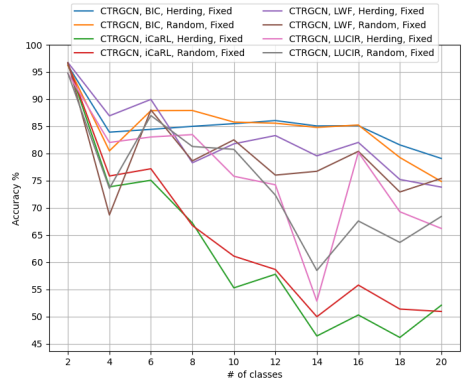


Figure 17: Comparative task-agnostic accuracy w/ CTR-GCN and fixed memory

CTR-GCN, Growing Results

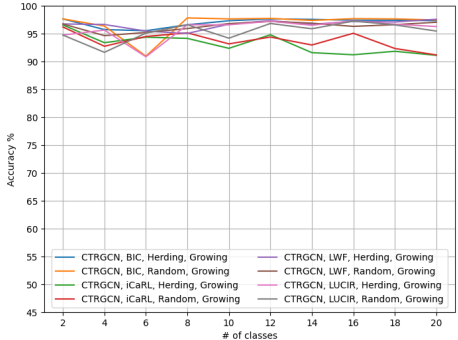


Figure 18: Comparative task-aware accuracy w/ CTR-GCN and growing memory

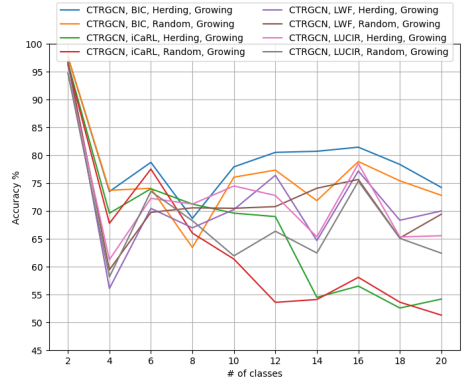


Figure 19: Comparative task-agnostic accuracy w/ CTR-GCN and growing memory

CTR-GCN + BiC (Per Task Accuracy)

```
****************************************************************************************************
TAw Acc
        95.6%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   Avg.: 95.6%
        93.3%  96.5%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   Avg.: 94.9%
        94.6%  96.2%  98.7%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   Avg.: 96.5%
        93.1%  92.9%  98.2%  99.3%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   Avg.: 95.9%
        92.0%  95.3%  98.2%  98.5%  99.6%   0.0%   0.0%   0.0%   0.0%   0.0%   Avg.: 96.7%
        91.7%  95.1%  98.5%  98.4%  99.6%  97.1%   0.0%   0.0%   0.0%   0.0%   Avg.: 96.7%
        93.1%  94.9%  94.0%  98.4%  99.5%  96.9%  99.8%   0.0%   0.0%   0.0%   Avg.: 96.7%
        94.4%  90.5%  92.7%  98.4%  99.3%  96.9%  99.8%  99.5%   0.0%   0.0%   Avg.: 96.4%
        93.5%  89.4%  92.6%  98.7%  99.6%  96.6% 100.0%  99.6%  97.4%   0.0%   Avg.: 96.4%
        94.4%  92.0%  94.9%  98.9%  99.6%  97.6% 100.0%  99.6%  97.4% 100.0%   Avg.: 97.5%
****************************************************************************************************
TAg Acc
        95.6%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   Avg.: 95.6%
        76.8%  68.1%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   Avg.: 72.4%
        54.4%  82.1%  98.5%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   Avg.: 78.4%
        33.6%  69.6%  94.4%  68.2%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   Avg.: 66.4%
        49.7%  58.5%  90.0%  56.2%  97.1%   0.0%   0.0%   0.0%   0.0%   0.0%   Avg.: 70.3%
        57.9%  55.9%  85.7%  70.8%  87.6%  90.9%   0.0%   0.0%   0.0%   0.0%   Avg.: 74.8%
        61.9%  46.6%  83.3%  79.2%  82.7%  80.2%  82.6%   0.0%   0.0%   0.0%   Avg.: 73.8%
        64.4%  52.1%  79.5%  86.1%  87.8%  75.5%  81.5%  93.6%   0.0%   0.0%   Avg.: 77.6%
        53.5%  56.5%  79.3%  81.2%  73.2%  68.8%  89.8%  93.3%  63.2%   0.0%   Avg.: 73.2%
        54.3%  33.2%  82.6%  72.1%  73.9%  75.9%  88.1%  92.0%  71.8%  74.8%   Avg.: 71.9%
****************************************************************************************************
```

Figure 20: Task-aware and task-agnostic accuracy over task sequence

CTR-GCN + BiC (Per Task Forget)

```
**********************************************************************************************************************
TAw Forg
         0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%
         2.4%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   Avg.:  2.4%
         1.1%   0.4%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   Avg.:  0.7%
         2.5%   3.6%   0.5%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   Avg.:  2.2%
         3.6%   1.3%   0.5%   0.7%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   Avg.:  1.5%
         4.0%   1.5%   0.2%   0.9%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   Avg.:  1.3%
         2.5%   1.6%   4.7%   0.9%   0.2%   0.2%   0.0%   0.0%   0.0%   0.0%   Avg.:  1.7%
         1.3%   6.0%   6.0%   0.9%   0.4%   0.2%   0.0%   0.0%   0.0%   0.0%   Avg.:  2.1%
         2.2%   7.1%   6.2%   0.5%   0.0%   0.5%  -0.2%  -0.2%   0.0%   0.0%   Avg.:  2.0%
         1.3%   4.6%   3.8%   0.4%   0.0%  -0.5%   0.0%   0.0%   0.0%   0.0%   Avg.:  1.1%
**********************************************************************************************************************
TAg Forg
         0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%
        18.9%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   Avg.: 18.9%
        41.2% -14.0%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   Avg.: 13.6%
        62.1%  12.6%   4.2%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   Avg.: 26.3%
        45.9%  23.7%   8.5%  12.0%   0.0%   0.0%   0.0%   0.0%   0.0%   0.0%   Avg.: 22.5%
        37.7%  26.2%  12.9%  -2.6%   9.5%   0.0%   0.0%   0.0%   0.0%   0.0%   Avg.: 16.8%
        33.8%  35.5%  15.2%  -8.4%  14.4%  10.7%   0.0%   0.0%   0.0%   0.0%   Avg.: 16.9%
        31.2%  30.1%  19.1%  -6.9%   9.3%  15.4%   1.1%   0.0%   0.0%   0.0%   Avg.: 14.2%
        42.1%  25.7%  19.2%   4.9%  23.9%  22.1%  -7.1%   0.4%   0.0%   0.0%   Avg.: 16.4%
        41.4%  49.0%  16.0%  14.1%  23.2%  15.1%   1.6%   1.6%  -8.6%   0.0%   Avg.: 17.0%
**********************************************************************************************************************
```

Figure 21: Task-aware and task-agnostic forgetting percentage over task sequence

Test Sequence #1 Results

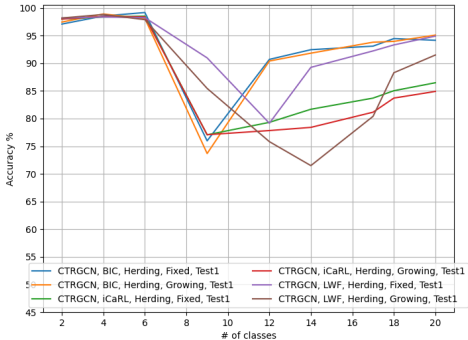


Figure 22: Task-agnostic accuracy for test sequence #1

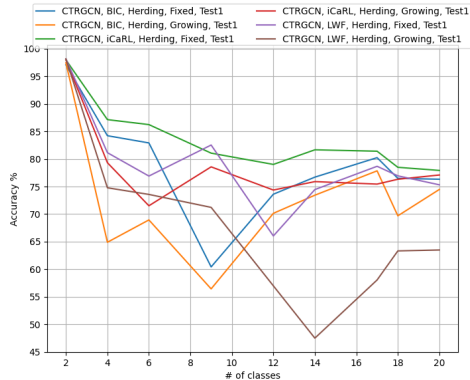


Figure 23: Task-aware accuracy for test sequence #1

Test Sequence #2 Results

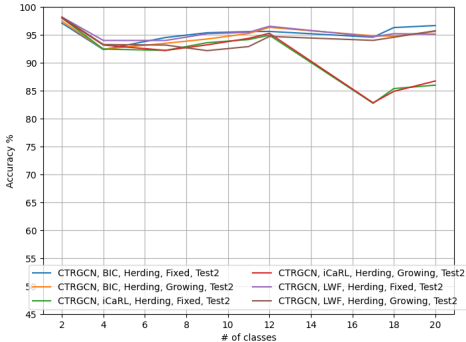


Figure 24: Task-aware accuracy for test sequence #2

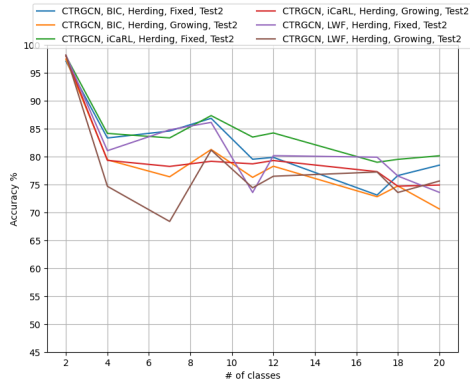


Figure 25: Task-agnostic accuracy for test sequence #2

Variable Memory Size Results

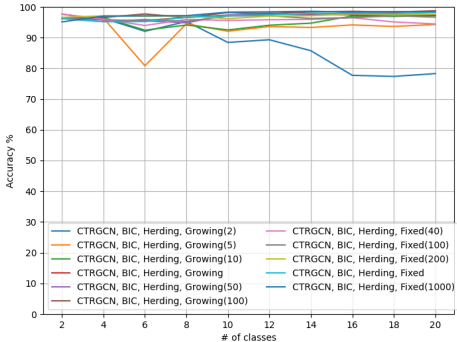


Figure 26: Task-aware accuracy for varying memory size

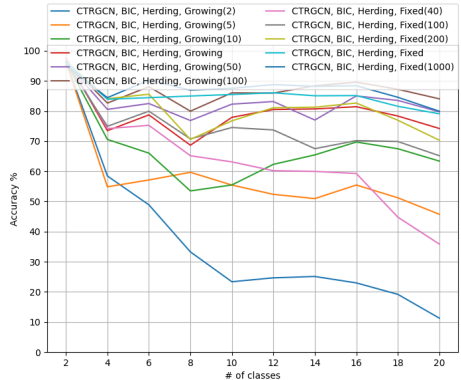


Figure 27: Task-agnostic accuracy for varying memory size

Incremental Learning Training

Algorithm 1 Incremental Learning Training Loop

Input: $X^m, \ldots, X^n$ // Training data
Input: K // memory size
Require: Θ // current model
Require: $P = (P_1, \ldots, P_{m-1})$ // Exemplar set
$\Theta \leftarrow UpdateModel(X^n, \ldots, X^m, P, \Theta)$ // Train model with new data
$k \leftarrow K/t$ // update number of exemplars per class
for $y = 1, \ldots, m-1$ **do**
    $P_y \leftarrow ReduceExemplars(P_y, k)$ // reduce class exemplars to comply with m
end for
for $y = m, \ldots, n$ **do**
    $P_y \leftarrow SelectExemplars(P_y, k, \Theta)$ // Select exemplars for new classes
end for
$P \leftarrow (P_1, \ldots, P_n)$ // Update exemplar set

LwF Approach

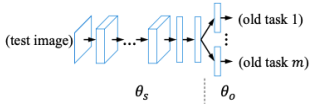


Figure 28: Original CNN model[11]

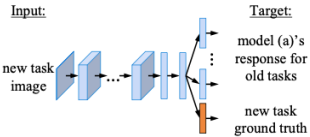


Figure 29: LwF architecture[11]

- Total Loss

$$L = \lambda_o L_{old} + L_{new} \tag{5}$$

- Multinomial Logistic Loss

$$L_{new} = -y_n * log\ \hat{y}_n \tag{6}$$

- Knowledge Distillation Loss

$$L_{old} = -\sum_{i=1}^{l} y_o^{'(l)} * log\ \hat{y}_o^{'(l)} \tag{7}$$

[11] Z. Li and D. Hoiem, "Learning without Forgetting," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 40, no. 12, pp. 2935–2947, Dec 2018.

iCaRL Approach

Algorithm 3 iCaRL UPDATEREPRESENTATION

input $X^s, \ldots, X^t$ // training images of classes $s, \ldots, t$
require $\mathcal{P} = (P_1, \ldots, P_{s-1})$ // exemplar sets
require Θ // current model parameters
// form combined training set:

$$\mathcal{D} \leftarrow \bigcup_{y=s,\ldots,t} \{(x, y) : x \in X^y\} \cup \bigcup_{y=1,\ldots,s-1} \{(x, y) : x \in P^y\}$$

// store network outputs with pre-update parameters:
for $y = 1, \ldots, s-1$ **do**
$q_i^y \leftarrow g_y(x_i)$ for all $(x_i, \cdot) \in \mathcal{D}$
end for
run network training (*e.g.* BackProp) with loss function

$$\ell(\Theta) = -\sum_{(x_i,y_i)\in\mathcal{D}} \Big[\sum_{y=s}^{t} \delta_{y=y_i} \log g_y(x_i) + \delta_{y\neq y_i} \log(1-g_y(x_i)) + \sum_{y=1}^{s-1} q_i^y \log g_y(x_i) + (1-q_i^y) \log(1-g_y(x_i))\Big]$$

that consists of *classification* and *distillation* terms.

Figure 30: iCaRL model training algorithm[12]

Algorithm 1 iCaRL CLASSIFY

input x // image to be classified
require $\mathcal{P} = (P_1, \ldots, P_t)$ // class exemplar sets
require $\varphi : \mathcal{X} \rightarrow \mathbb{R}^d$ // feature map
for $y = 1, \ldots, t$ **do**

$$\mu_y \leftarrow \frac{1}{|P_y|} \sum_{p \in P_y} \varphi(p)$$ // mean-of-exemplars

end for

$$y^* \leftarrow \underset{y=1,\ldots,t}{\operatorname{argmin}} \|\varphi(x) - \mu_y\|$$ // nearest prototype

output class label y^*

Figure 31: iCaRL classifier algorithm[12]

[12] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental Classifier and Representation Learning," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2017, pp. 5533–5542.

LUCIR Approach

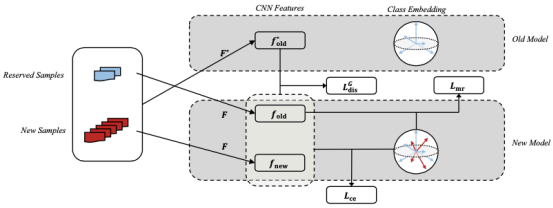


Figure 32: LUCIR approach[13]

- Classification Loss

$$L_{ce}(x) = -\sum_{i=1}^{|C|} y_i log(p_i) \tag{8}$$

- Distillation Loss

$$L^G_{dis}(x) = 1 - \langle \bar{f}^*(x), \bar{f}(x) \rangle \tag{9}$$

- Margin Ranking Loss

$$L_{mr}(x) = \sum_{k=1}^{K} max(m - \langle \bar{\theta}(x), \bar{f}(x) \rangle + \langle \bar{\theta}^k, \bar{f}(x) \rangle, 0) \tag{10}$$

- Total Loss

$$L = \frac{1}{|N|} \sum_{x \in N} (L_{ce}(x) + \lambda L^G_{dis}(x)) + \frac{1}{|N_o|} \sum_{x \in N_o} L_{mr}(x) \tag{11}$$

[13] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, "Learning a Unified Classifier Incrementally via Rebalancing," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2019, pp. 831–839.

BiC Approach

- Total Loss

$$L = \lambda L_d + (1 - \lambda) L_c \qquad (12)$$

- Classification Loss

$$L_{ce}(x) = -\sum_{i=1}^{|C|} y_i log(p_i) \qquad (13)$$

- Distillation Loss

$$L_d = \sum_{x \in \hat{X}^n \cup X^m} \sum_{k=1}^{n} -\hat{\pi}_k(x) log[\pi_k(x)],$$

$$\hat{\pi}_k(x) = \frac{e^{\hat{o}_k(x)/T}}{\sum_{j=1}^{n} e^{\hat{o}_j(x)/T}}, \ \pi_k(x) = \frac{e^{o_k(x)/T}}{\sum_{j=1}^{n} e^{o_j(x)/T}} \qquad (14)$$

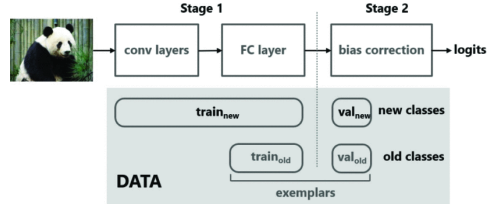


Figure 33: BiC approach[14]

- Bias Correction Loss

$$L_b = -\sum_{k=1}^{n+m} \delta_{y=k} log[softmax(q_k)],$$

$$q_k = \begin{cases} o_k & 1 \leq k \leq n \\ \alpha o_k + \beta & n+1 \leq k \leq n+m \end{cases} \qquad (15)$$

[14] Y. Wu et al., "Large Scale Incremental Learning," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2019, pp. 374–382.

CTR-GCN Network

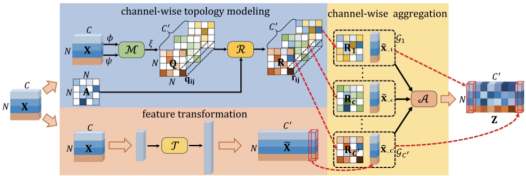


Figure 34: CTR-GC block architecture[15]

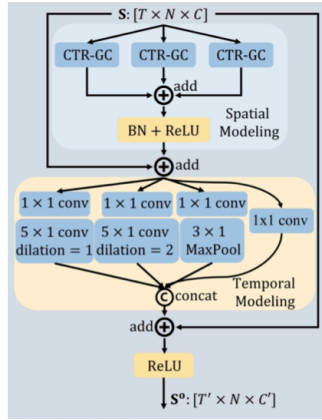


Figure 35: CTR-GCN network architecture[15]

[15] Y. Chen et al., "Channel-Wise Topology Refinement Graph Convolution for Skeleton-Based Action Recognition," in Proc. IEEE/CVF Int. Conf. Computer Vision (ICCV), 2021, pp. 13 359–13 368.

MS-G3D Network

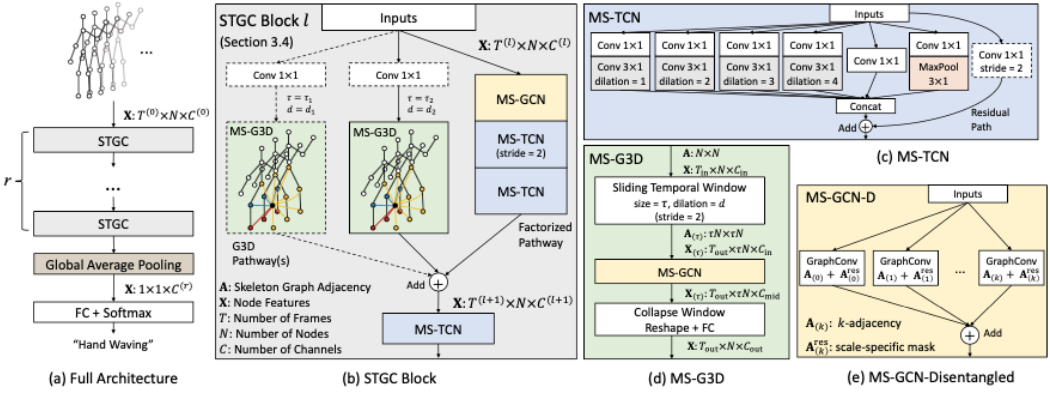


Figure 36: MS-G3D Architecture[16]

[16] Z. Liu, H. Zhang, Z. Chen, Z. Wang, and W. Ouyang, "Disentangling and Unifying Graph Convolutions for Skeleton-Based Action Recognition," in IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2020, pp. 140–149.