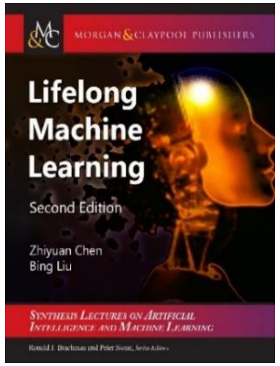
Unifying Continual Learning, OOD Detection & Open World Learning



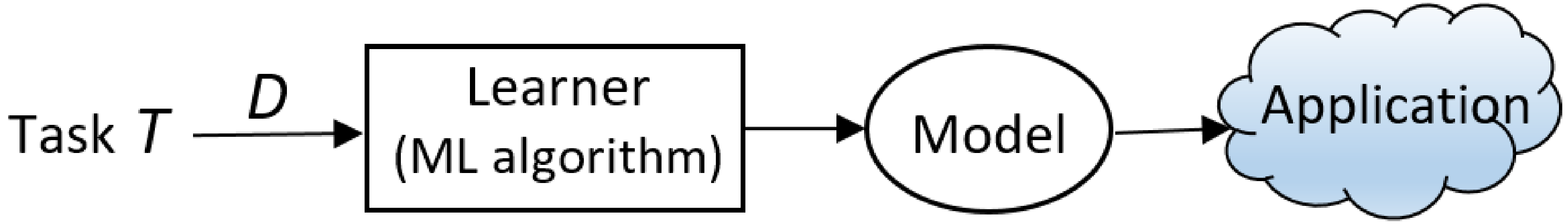
Bing Liu
Department of Computer Science
University of Illinois Chicago

Introduction

- Our human brains are hardwired to acquire knowledge. We are curious creatures who try to make sense of the world that confronts us. This natural inclination has helped our species evolve from early times (Brown & Dryden, 2004; Holt, 1983).
- To build an AI agent, we must endow it the ability to
  - learn continually in the open and dynamic world that is full of unknowns on its own initiative (by itself) with self-motivation and
  - with interaction with humans, other agents and the environment.

Introduction

- Classic machine learning: Isolated single-task learning



- Key weaknesses
  - **Closed-world assumption**: nothing new/novel occurs in application
  - **No continual learning**: No knowledge accumulation or transfer
  - **No learning after deployment**: model fixed after deployment

Chen and Liu. Lifelong machine learning. Morgan & Claypool. 2016, 2018
Liu. Learning on the Job: Online Lifelong and Continual Learning. AAAI-2020

Open world continual learning

- Closed-world: (test classes) $Y^{test} \subseteq Y^{train}$ (training classes)
  - Classes appeared in testing must have been seen in training, **nothing new.**
- Open world: with unknowns or novelties, i.e., $Y^{test} - Y^{train} \neq \phi$
  - **A system unable to detect anything new cannot learn by itself.**
- Open-World (continual) Learning (OWL)
  - **Out-of-distribution (OOD) detection**: novelty detection, anomaly/outlier detection
  - **Continual learning**: learning detected/given new objects/tasks incrementally.
- **Autonomy:** OWL is still insufficient. AI agents must learn by itself
  - ***S***elf-initiated ***O***pen-world continual ***L***earning and ***A***daptation (**SOLA**)

Liu, Mazumder, Robertson and Grigsby. AI Autonomy: Self-Initiated Open-World Continual Learning and Adaptation. AI Magazine, May 21, 2023

Chen and Liu. Lifelong machine learning. Morgan & Claypool. 2015, 2018.

Outline

- Continual or lifelong learning
- Class incremental learning (CIL)
  - Theoretical result and algorithms
- Open world continual learning
  - A continuous learning chatbot
- Continual pre-training of language models
- Summary

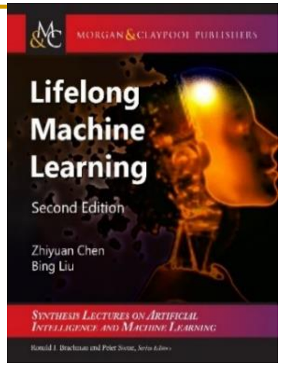
Continual or lifelong learning

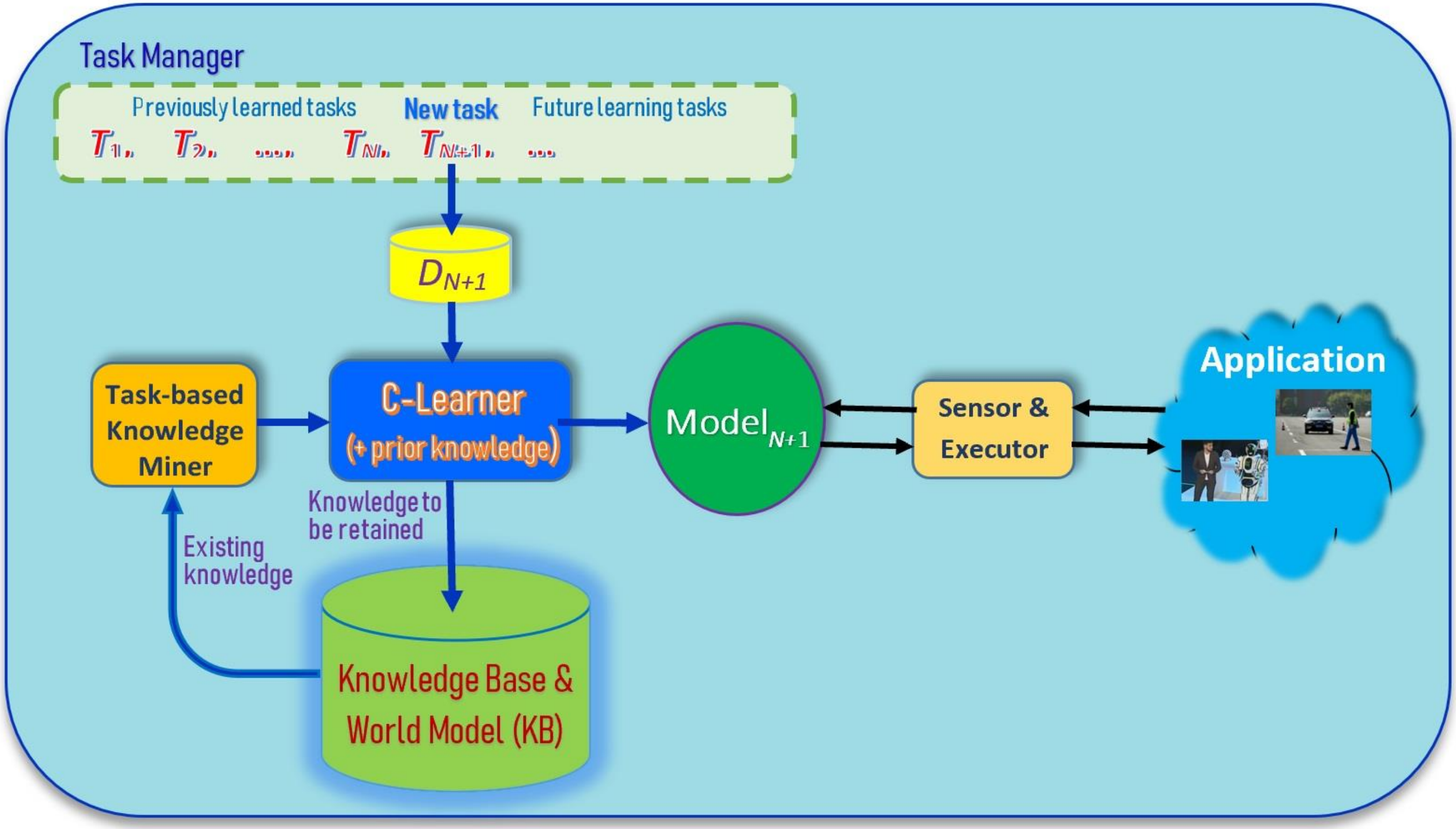
(Thrun 1996, Silver et al 2013; Ruvolo and Eaton, 2013; Chen and Liu, 2018)

- Learn a sequence of tasks, T_1, T_2, …, T_N, … incrementally. Each task *t* has a training dataset $\mathcal{D}_t = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ in a neural network.
  - In supervised learning, a task is a set of classes to be learned.
- **Goal:** learn each new task T_{N+1} incrementally
  1. **without catastrophic forgetting (CF)**: Learning of the new task T_{N+1} should not result in accuracy degradation for any of the previous *N* tasks.
  2. **with knowledge transfer (KT)**: leveraging the knowledge learned from the previous tasks to learn the new task T_{N+1} better.

Traditional lifelong/continual learning

(Chen and Liu, 2014, 2018)





In supervised ML, a **task** is a set of classes to learn

Assumption:

Both the task T_{N+1} and its training data D_{N+1} are **given**.

Liu. Learning on the Job: Online Lifelong and Continual Learning. AAAI-2020
Chen and Liu. Lifelong machine learning. Morgan & Claypool. 2018

Assumptions

- Once a task is learned its data is no longer accessible, at least most of it.
  - In the replay approach, we can save a small amount of past data
- Both the new task T_{N+1} and its training data D_{N+1} **are given by the user**.

Two popular CL settings: TIL

- Task incremental learning (TIL): train a "separate" model for each task and task-id is provided during testing
  - **Example**: Task 1: learn to recognize different breeds of dogs. Task 2: learn to recognize different animals. Task 3: learn to recognize different types of fish.
  - Testing needs task information (e.g., task id).

Task incremental learning (TIL). TIL learns a sequence of tasks, $1, 2, ..., T$. Each task k has a training dataset $\mathcal{D}_k = \{((x_k^i, k), y_k^i)_{i=1}^{n_k}\}$, where n_k is the number of data samples in task $k \in \mathbf{T} = \{1, 2, ..., T\}$, and $x_k^i \in \mathbf{X}$ is an input sample and $y_k^i \in \mathbf{Y}_k \subset \mathbf{Y}$ is its class label. The goal of TIL is to construct a predictor $\boxed{f : \mathbf{X} \times \mathbf{T} \rightarrow \mathbf{Y}}$ to identify the class label $y \in \mathbf{Y}_k$ for (x, k) (the given test instance x from task k).

Two popular CL settings: CIL

- Class incremental learning (CIL): produce a single model from all tasks and classify all classes during testing
  - **Example**: Task 1: learn to recognize pig and cat. Task 2: sheep. Task 3: chicken and dog. Task 4: horse and cow
  - Testing:

Class incremental learning (CIL). CIL learns a sequence of tasks, $1, 2, ..., T$. Each task k has a training dataset $\mathcal{D}_k = \{(x_k^i, y_k^i)_{i=1}^{n_k}\}$, where n_k is the number of data samples in task k, and $x_k^i \in \mathbf{X}$ is an input sample and $y_k^i \in \mathbf{Y}_k$ (the set of all classes of task k) is its class label. All $\mathbf{Y}_k$'s are disjoint ($\mathbf{Y}_k \cap \mathbf{Y}_{k'} = \emptyset, \forall k \neq k'$) and $\bigcup_{k=1}^{T} \mathbf{Y}_k = \mathbf{Y}$. The goal of CIL is to construct a single predictive function or classifier $\boxed{f : \mathbf{X} \rightarrow \mathbf{Y}}$ that can identify the class label y of each given test instance x.

Kim, Xiao, Konishi, Ke and Liu. A Theoretical Study on Solving Continual Learning. NeurIPS-2022, Nov. 28 - Dec. 9, 2022

van de Ven and Tolias Three scenarios for continual learning, *arXiv preprint arXiv*:1904.07734, 2019.

Additional challenge of CIL

- In addition to the challenges of catastrophic forgetting and knowledge transfer.
- CIL has an additional challenge **inter-task class separation (ICS)**, which is very hard to handle.
  - Since after learning each task, its data is no longer accessible, then how to establish the decision boundaries between the classes of the new task and those of old tasks.
  - Existing research has not dealt with this problem explicitly. Replay methods deal with this implicitly to a limited extent.

- Question: What is the right way to solve CIL regardless what classification algorithm is used?

Kim, Xiao, Konishi, Ke and Liu. A Theoretical Study on Solving Continual Learning. NeurIPS-2022, Nov. 28 - Dec. 9, 2022.
Guo, Liu, Zhao. Cross-task class discrimination in online continual learning. CVPR-2023

CIL decomposition and theoretical result

- CIL problem can be decomposed into two subproblems: within-task prediction (WP) and task-id prediction (TP)

$$\mathbf{P}(x \in \mathbf{X}_{k_0,j_0}|D) = \sum_{k=1,\ldots,n} \mathbf{P}(x \in \mathbf{X}_{k,j_0}|x \in \mathbf{X}_k, D)\mathbf{P}(x \in \mathbf{X}_k|D)$$

$$= \mathbf{P}(x \in \underbrace{\mathbf{X}_{k_0,j_0}|x}_{\text{WP (i.e., TIL)}} \in \mathbf{X}_{k_0}, D)\mathbf{P}(\underbrace{x \in \mathbf{X}_{k_0}}_{\text{TP}}|D)$$

- **Theoretical result:** Good WP and TP (or OOD) are ***necessary*** and ***sufficient*** for good CIL.
  - Connect/unify CIL and OOD

Kim, Xiao, Konishi, Ke and Liu. A Theoretical Study on Solving Continual Learning. NeurIPS-2022, Nov. 28 - Dec. 9, 2022..

Results

- **Loss of CIL is bounded by those of WP and TP**

  Theorem 1. *If $H_{TP}(x) \leq \delta$ and $H_{WP}(x) \leq \epsilon$, we have $H_{CIL}(x) \leq \epsilon + \delta$.*

  - CIL improves with WP or TP

- **TP and OOD detection bound each other**

  Theorem 2. *i) If $H_{TP}(x) \leq \delta$, let $\mathbf{P}'_k(x \in \mathbf{X}_k|D) = \mathbf{P}(x \in \mathbf{X}_k|D)$, then $H_{OOD,k}(x) \leq \delta, \forall\, k = 1, \ldots, T$. ii) If $H_{OOD,k}(x) \leq \delta_k, k = 1, \ldots, T$, let $\mathbf{P}(x \in \mathbf{X}_k|D) = \frac{\mathbf{P}'_k(x \in \mathbf{X}_k|D)}{\sum_k \mathbf{P}'_k(x \in \mathbf{X}_k|D)}$, then $H_{TP}(x) \leq (\sum_k \mathbf{1}_{x \in \mathbf{X}_k} e^{\delta_k})(\sum_k 1 - e^{-\delta_k})$, where $\mathbf{1}_{x \in \mathbf{X}_k}$ is an indicator function.*

- **Loss of CIL is bounded by those of WP and OOD detection**

  Theorem 3. *If $H_{OOD,k}(x) \leq \delta_k$, $k = 1, \ldots, T$ and $H_{WP}(x) \leq \epsilon$, we have*

$$H_{CIL}(x) \leq \epsilon + (\sum_k \mathbf{1}_{x \in \mathbf{X}_k} e^{\delta_k})(\sum_k 1 - e^{-\delta_k}),$$

  where $\mathbf{1}_{x \in \mathbf{X}_k}$ is an indicator function.

Kim, Xiao, Konishi, Ke and Liu. A Theoretical Study on Solving Continual Learning. NeurIPS-2022, Nov. 28 - Dec. 9, 2022..

Necessary condition for CIL

- **Theorems 1, 2, and 3** show that good performances of WP and TP or (OOD) are *sufficient* to guarantee good CIL
- **Theorem 4** shows that good performances of WP and TP (or OOD) are *necessary* for good CIL.

Theorem 4. *If $H_{CIL}(x) \leq \eta$, then there exist i) a WP, s.t. $H_{WP}(x) \leq \eta$, ii) a TP, s.t. $H_{TP}(x) \leq \eta$, and iii) an OOD detector for each task, s.t. $H_{OOD,k} \leq \eta$, $k = 1, \ldots, T$.*

- Most OOD methods can perform both WP and OOD detection.
  - Good CIL requires good OOD performance.
- **Note:** The theory is also applicable to unsupervised CL.

Kim, Xiao, Konishi, Ke and Liu. A Theoretical Study on Solving Continual Learning. NeurIPS-2022, Nov. 28 - Dec. 9, 2022..

Intuition of the theory

- To ensure that good decision boundaries between the classes learned so far and unknown future classes.
  - The model must achieve the OOD detection effect (?).
  - If a CIL model is perfect at detecting OOD samples for each task, the ICS problem is solved, and CIL is reduced to WP.
    - WP is basically the in-distribution (IND) classification.
    - What about CF? That can be solved with a TIL method.
- Note: The theory says what a CIL system should achieve, but it does not say how to achieve it.

An implication

  … 

- CIL needs to consider both the past and the future.
  - The algorithm must learn all the characteristics of the classes in each task;
    - i.e, learn holistic representations (Guo et al. 2022)
  - Otherwise, it will not be able to solve the ICS problem.
- Are traditional principles or theories for machine learning (in the closed world) still "appropriate"?
  - E.g., **Occam's Razor**: the simplest of competing models is preferred
    - But is it correct when we must consider the learning of future unknown classes?
  - Is cross-entropy the right loss function?

Guo, Liu and Zhao. Online Continual Learning through Mutual Information Maximization ICML-2022.

Proposed method 1 (without pre-training)

- Theory-based methods outperform baselines by a large margin

- No replay or pre-training
- Combination of
  - a TIL method to tackle CF
    - E.g., HAT and SupSup
  - a strong OOD detection
    - E.g., CSI.
- **HAT+CSI** and **Sup+CSI**

Method	M-5T	C10-5T	C100-10T	C100-20T	T-5T	T-10T
OWM	95.8±0.13	51.8±0.05	28.9±0.60	24.1±0.26	10.0±0.55	8.6±0.42
MUC	74.9±0.46	52.9±1.03	30.4±1.18	14.2±0.30	33.6±0.19	17.4±0.17
PASS†	76.6±1.67	47.3±0.98	33.0±0.58	25.0±0.69	28.4±0.51	19.1±0.46
LwF	85.5±3.11	54.7±1.18	45.3±0.75	44.3±0.46	32.2±0.50	24.3±0.26
iCaRL*	96.0±0.43	63.4±1.11	51.4±0.99	47.8±0.48	37.0±0.41	28.3±0.18
Mnemonics†*	96.3±0.36	64.1±1.47	51.0±0.34	47.6±0.74	37.1±0.46	28.5±0.72
BiC	94.1±0.65	61.4±1.74	52.9±0.64	48.9±0.54	41.7±0.74	33.8±0.40
DER++	95.3±0.69	66.0±1.20	53.7±1.30	46.6±1.44	35.8±0.77	30.5±0.47
Co^{2}L		65.6				
CCG	97.3	70.1				
HAT	81.9±3.74	62.7±1.45	41.1±0.93	25.6±0.51	38.5±1.85	29.8±0.65
HyperNet	56.6±4.85	53.4±2.19	30.2±1.54	18.7±1.10	7.9±0.69	5.3±0.50
Sup	70.1±1.51	62.4±1.45	44.6±0.44	34.7±0.30	41.8±1.50	36.5±0.36
PR-Ent	74.1	61.9	45.2			
HAT+CSI	94.4±0.26	87.8±0.71	63.3±1.00	54.6±0.92	45.7±0.26	47.1±0.18
Sup+CSI	80.7±2.71	86.0±0.41	65.1±0.39	60.2±0.51	48.9±0.25	45.7±0.76
HAT+CSI+c	96.9±0.30	88.0±0.48	65.2±0.71	58.0±0.45	51.7±0.37	47.6±0.32
Sup+CSI+c	81.0±2.30	87.3±0.37	65.2±0.37	60.5±0.64	49.2±0.28	46.2±0.53

Kim, Xiao, Konishi, Ke and Liu. A Theoretical Study on Solving Continual Learning. NeurIPS-2022, Nov. 28 - Dec. 9, 2022..

Proposed method 2 (with pre-training)

- In method 2,
  - Use a pre-trained model,
    - which is trained without using the class/data used in CIL
  - Leverage the replay data
- The TIL method HAT is still used to deal with forgetting
  - For each model, we build an OOD detection model for each task
  - The replay data is used as the OOD data in training the model for the current task.
  - Partial back-update is also done.

Proposed Method 2 – results

- Use a pre-trained model,
  - which is trained without using the classes/data used in CIL
  - Leverage the replay data

Method	CIFAR10-5T ACA	CIFAR10-5T AIA	CIFAR100-10T ACA	CIFAR100-10T AIA	CIFAR100-20T ACA	CIFAR100-20T AIA	T-ImageNet-5T ACA	T-ImageNet-5T AIA	T-ImageNet-10T ACA	T-ImageNet-10T AIA	Average ACA	Average AIA
OWM	41.69±6.34	59.07±3.31	21.39±3.18	39.71±1.35	16.98±4.44	32.18±1.51	24.55±2.48	45.65±1.15	17.52±3.45	35.57±1.83	24.43	41.99
iCaRL	87.55±0.99	92.75±1.08	68.90±0.47	77.82±1.28	69.15±0.99	77.74±1.82	53.13±1.04	63.35±2.02	51.88±2.36	64.62±0.97	66.12	75.26
A-GEM	56.33±7.77	71.22±1.42	25.21±4.00	43.39±0.88	21.99±4.01	35.56±0.95	30.53±3.99	50.37±2.15	21.90±5.52	39.79±3.28	31.20	47.74
EEIL	82.34±3.13	90.50±0.72	68.08±0.51	81.09±0.37	63.79±0.66	79.54±0.69	53.34±0.54	66.63±0.40	50.38±0.97	66.54±0.61	63.59	76.86
GD	**89.16**±0.53	94.22±0.75	64.36±0.57	80.51±0.57	60.10±0.74	78.43±0.76	53.01±0.97	67.51±0.38	42.48±2.53	63.91±0.40	61.82	76.92
DER++	84.63±2.91	91.81±0.65	69.73±0.99	**81.71**±0.67	70.03±1.46	**82.24**±0.79	55.84±2.21	68.47±0.73	54.20±3.28	68.06±1.04	66.89	78.46
HAL	84.38±2.70	90.41±1.04	67.17±1.50	78.62±0.45	67.37±1.45	78.43±0.61	52.80±2.37	67.52±0.93	55.25±3.60	67.89±2.32	65.39	76.57
PASS	86.21±1.10	91.78±1.12	68.90±0.94	78.27±0.81	66.77±1.18	77.01±1.13	61.03±0.38	70.02±0.56	58.34±0.42	68.45±1.20	68.25	77.11
HAT	83.30±1.54	91.06±0.36	62.34±0.93	73.99±0.86	56.72±0.44	69.12±1.06	57.91±0.72	69.38±1.14	53.12±0.94	65.63±1.64	62.68	73.84
MORE	**89.16**±0.96	**94.23**±0.82	**70.23**±2.27	81.24±1.24	**70.53**±1.09	81.59±0.98	**64.97**±1.28	**74.03**±1.61	**63.06**±1.26	**72.74**±1.04	**71.59**	**80.77**

Kim, Ke, and Liu. A Multi-Head Model for Continual Learning via Out-of-Distribution Replay. CoLLAs-2022

Learnability of CIL

- In (Kim et al. 2023), we show that CIL is learnable
- Main Idea:
  - We still follow the framework/setting of using a TIL method to avoid forgetting or CF, and
  - An OOD detection method to build a model for each task.
  - Then we need OOD detection to be learnable, which fortunately was proven in (Fang et al. 2022).
    - We need a sequence of OOD models to be learnable, which can be recursively defined and proved.

Kim, Xiao, Konishi, and Liu. Learnability and Algorithm for Continual Learning. ICML-2023
Fang, Li, Lu, Dong, Han and Liu. Is out-of-distribution detection learnable? NeurIPS-2022

Outline

- Continual or lifelong learning
- Class incremental learning (CIL)
  - Theoretical result and algorithms
- Open world continual learning
  - A continuous learning chatbot
- Continual pre-training of language models
- Summary

Open-world continual learning

- Open-World (continual) Learning (OWL)
  - (1) detect novel/new objects (OOD detection) and
  - (2) incrementally learn the new objects after they are labeled.
- The **theoretical result for CIL** is also applicable here because
  - (1) Theory says WP and OOD detection are necessary and sufficient conditions, and
  - (2) an OOD detection system usually does both OOD detection and in-distribution (IND) classification, which is WP (within-task prediction).
- Autonomous AI agents: SOLA = OWL + Adaptation
  - Including getting training data by the AI agent itself.

Liu, Mazumder, Robertson and Grigsby. AI Autonomy: Self-Initiated Open-World Continual Learning and Adaptation. AI Magazine, May 21, 2023

An experience with self-driving

nature

Explore content ⌄ About the journal ⌄ Publish with us ⌄ Subscribe

nature > outlook > article

OUTLOOK | 20 July 2022 | Correction 01 September 2022

Learning over a lifetime

Artificial-intelligence researchers turn to lifelong learning in the hopes of making machine intelligence more adaptable.

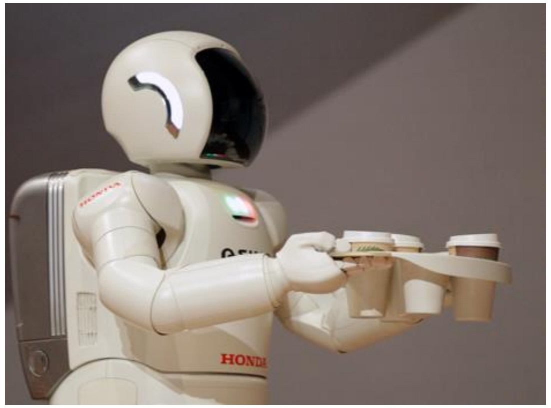
Neil Savage

Bing Liu was road testing a self-driving car, when suddenly something went wrong. The vehicle had been operating smoothly until it reached a T-junction and refused to move. Liu and the car's other occupants were baffled. The road they were on was deserted, with no pedestrians or other cars in sight. "We looked around, we noticed nothing in the front, or in the back. I mean, there was nothing," says Liu, a computer scientist at the University of Illinois Chicago.

- I consulted for a self-driving car company.
- We took a self-driving car for a road test.
  - At a T-junction, it stopped & refused to move.
    - Every direction was clear, nothing on the road.
- Our human driver had to take over.
  - Debugging found that a sensor detected a pebble on the road.
  - If the car had said "*I detected an unknown object here. What should I do?*" we would have replied "*It is safe. Go ahead*."
    - The car can then learn the new object so that it will have no issue next time.
      - **Learning on the fly** (on the job)

Chatbots should learn continually after deployment

- Chatbot: human users may say things a chatbot does not understand.
  - It should learn new knowledge and new language expressions **during chatting**.
    - E.g., asking the current or other users.
  - We humans learn a great deal in our daily conversations
- Chatbots **should not** solely rely on offline training initiated by engineers.



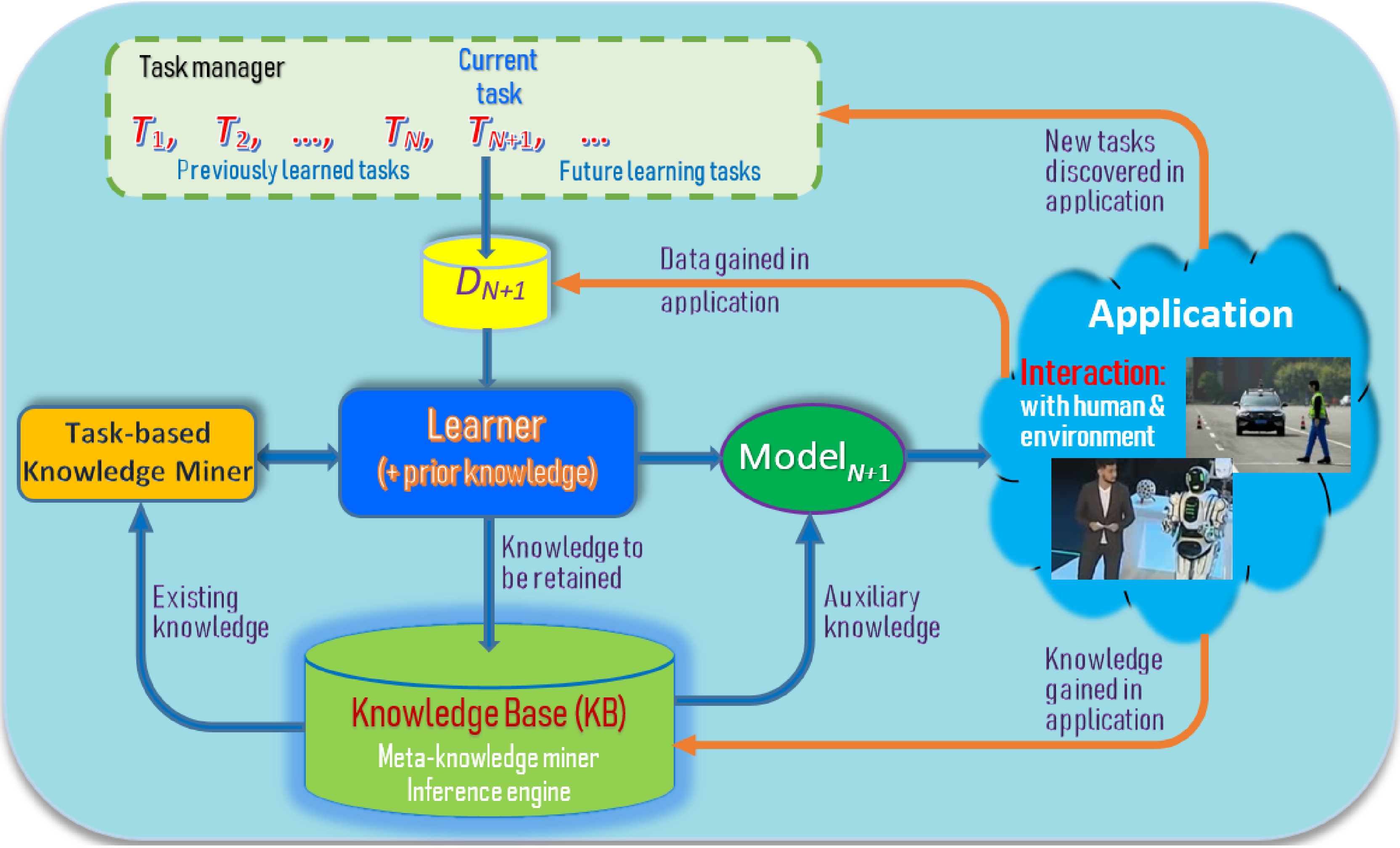
Liu and Mazumder. Lifelong and Continual Learning Dialogue Systems: Learning during Conversation. AAAI-2021

SOLA is ***necessary*** for AGI/AI agents

- Eventually, **AI agents needs SOLA** as the real world is an open & dynamic environment, full of unknowns/OOD objects
  - **SOLA:** Perform **OWL** after model deployment **autonomously** and **continually** in a self-motivated and self-supervised manner and **adapt** to the unseen environment.
- Self-motivation: detect novel/unknown/OOD objects to learn.
  - Novelties or unknowns are an intrinsic motivation for (human) learning
- Self-supervision: collect ground-truth training data **by agent itself** via
  - Interact with humans, other AI agents, and the environment
- Adaptation/accommodation: Adapt to the novel/OOD environment
  - Need planning, actions, and risk assessment

Liu, Mazumder, Robertson and Grigsby. AI Autonomy: Self-Initiated Open-World Continual Learning and Adaptation. AI Magazine, May 21, 2023
Chen and Liu. Lifelong machine learning. Morgan & Claypool. 2015, 2018.

Simple version: Open-world continual learning



Orange lines:

Learning after model deployment

- Learning on the job

Liu. Learning on the Job: Online Lifelong and Continual Learning. AAAI-2020
Chen and Liu. Lifelong machine learning. Morgan & Claypool. 2015, 2018

Example - a greeting bot in a hotel

Novelty detection = out-of-distribution (**OOD**) detection

- See an existing guest.
  - Bot: "Hello John, how are you today?"
- See a new guest - recognize he/she is new **(OOD and create a new task)**
  - Bot: "Welcome to our hotel! What is your name, sir?" **(get class label)**
  - Guest: "David" **(got class label: David)**
  - Bot learns to recognize David automatically
    - take pictures of David **(get training data)**
    - learn to recognize David **(continual learning)**
- See David next time.
  - Bot: "Hello David, how are you today?" **(use the new knowledge)**

Liu, Mazumder, Robertson and Grigsby. AI Autonomy: Self-Initiated Open-World Continual Learning and Adaptation. AI Magazine, May 21, 2023
Chen and Liu. Lifelong machine learning. Morgan & Claypool. 2015, 2018.

Example (cont.)

- In a real hotel, the situation is much more complex.
  - How does the bot know that the novel object is a new guest?
    - Is the object a person, an animal, or a piece of furniture?
      - needs to use existing knowledge to characterize the novel object!
  - Different **characterizations** require different responses (**adaptation** or **accommodation** strategies)? E.g.,
    - If it looks like an animal, report to a hotel staff.
    - If it looks like a hotel guest (with luggage), ask for his/her name: "*Welcome to our hotel! What is your name, sir?*" and learn to recognize him/her
- **Thanks to DARPA Sail-On program participants for numerous discussions**

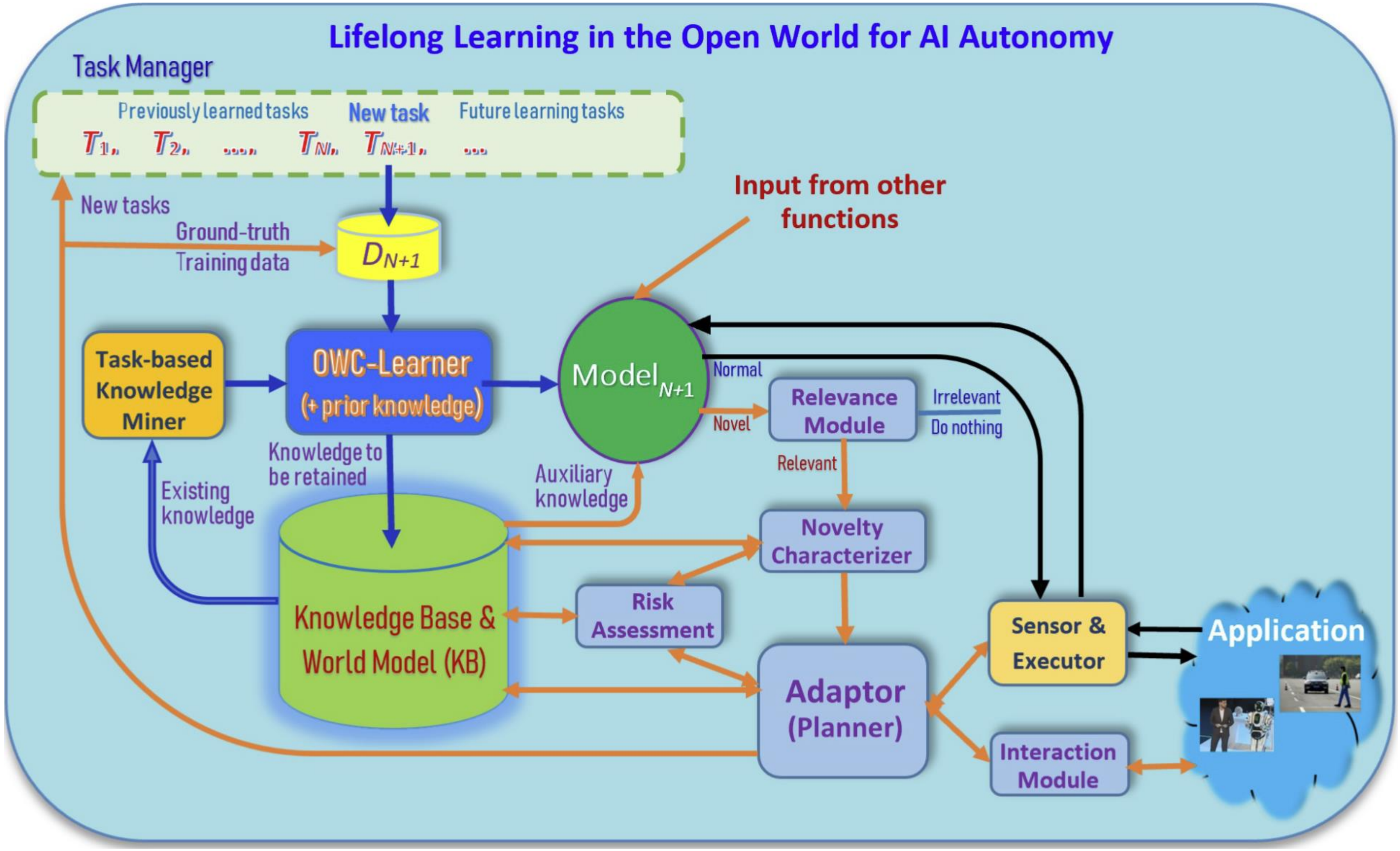
Liu, Mazumder, Robertson and Grigsby. AI Autonomy: Self-Initiated Open-World Continual Learning and Adaptation. AI Magazine, May 21, 2023

Novelty characterization and adaptation

- Characterization: a description of the novel object based on the agent's existing knowledge.
    - ***Similarity***: *e.g., it looks like a dog.*
    - **Attributes/properties**: e.g., a moving object, speed and direction of moving.
- Adapting to novelty: a pair (*Characterization*, *Response*)
    - Response: According to characterization, formulate a specific course of actions to respond to the novelty, e.g.,
        - If novel object looks like an animal (characterization), report to hotel staff (response).
        - If ***cannot characterize***, take ***default action*** (e.g., do nothing)
    - Enable continual learning
- Risk assessment: each decision carries risks

Liu, Mazumder, Robertson and Grigsby. AI Autonomy: Self-Initiated Open-World Continual Learning and Adaptation. AI Magazine, May 21, 2023

SOLA: **S**elf-initiated **O**pen-world continual **L**earning & **A**daptation



Blue lines: Existing continual learning

Orange lines:

Learning after model deployment

(Open-world continual learning)

Liu, Mazumder, Robertson and Grigsby. AI Autonomy: Self-Initiated Open-World Continual Learning and Adaptation. AI Magazine, May 21, 2023

Outline

- Continual or lifelong learning
- Class incremental learning (CIL)
  - Theoretical result and algorithms
- Open world continual learning
  - A continuous learning chatbot
- Continual pre-training of language models
- Summary

Example SOLA: natural language interface (NLI)

- Performance task: user asks the system (CML, like **Siri** and **Alexa**) to perform a task in NL, the system does it via API actions
  - Approach: *natural language to natural language matching* (**NL2NL**)
- **CML** builds NLIs for API-driven applications semi-automatically.
- **To build a new NLI** (or add a new skill to an existing NLI),
  - App developer writes a set S_i of seed commands (SCs) to represent each API action *i*.
    - SCs in S_i are just like paraphrased NL commands from users to invoke *i*, but the objects to be acted upon in each SC are replaced with variables, the arguments of action *i*.
  - When the system does not understand a command (**novelty**), it **adapts** and **learns** new (paraphrased) SCs from users interactively and continually.

Mazumder, Liu, Wang, and Esmaeilpour. An Application-Independent Approach to Building Task-Oriented Chatbots with Interactive Continual Learning. *NeurIPS-2020 Workshop on Human in the Loop Dialogue Systems (HLDS-2020)*

An example – Smart home

- SmartHome: API action: *SwitchOnLight*(X1)
  - Switching on a light at a given place X1

API (arg : arg type)	Seed Command (SCs)	Example NL command
SwithOnLight(X1: location)	Switch on the light in X1 Put on light in X1	Switch on the light in the **bedroom** (**X1**)
SwithOffLight(X1: location)	Switch off the light in X1 Put off light in X1	Switch off the light in the **bedroom** (**X1**)
ChangeLightColor (X1 : location, x2: color)	Change the X1 light to X2 I want X1 light to be X2	Change the **bedroom** (**X1**) light to **blue** (**X2**)

- Let an SC be "put on light in X1" for this API,
  - where X1 is a variable representing the argument of the API.
- User command: "power on the light in the bedroom"
  - It can be matched or grounded to this SC, where the grounded API arguments are {X1 = 'bedroom'}.

Novelty detection, characterization, adaptation

- Novelty detection: when CML cannot ground a user command, e.g., it cannot understand "*turn off the light in the kitchen*"
- Novelty characterization: which part of the command it understands and which part it has difficulty based on similarity. E.g., it cannot ground "*turn off*"
- Adaptation (or accommodation):
  - Response: ask the user by providing some options (to collect ground-truth data)

    Bot: Sorry, I didn't get you. Do you mean to:

    option-1. switch off the light in the kitchen,
    option-2. switch on the light in the kitchen, or

  - Continual learning: learn a **new SC.** No issue with related commands in future.

Adaptation enabled continual learning

- **Interaction with humans and learn**
  - E.g., for the greeting bot, ask the human using the interactive module ***I*** (in natural language) to get ground-truth data and incrementally learn.
- **Imitation learning**.
  - E.g., on seeing a novel object by a self-driving car, if the car in front drives through it with no issue, the car may choose the same course of action as well and learn it for future use.
- **Perform limited reinforcement learning**.
  - By interacting with the environment through trial-and-error exploration, the agent learns a good response policy for future use.

Liu, Mazumder, Robertson and Grigsby. AI Autonomy: Self-Initiated Open-World Continual Learning and Adaptation. AI Magazine, May 21, 2023

Risk consideration

- CML manages **risk** in two ways
  - Do not ask user too many questions in order not to annoy the user.
    - Learning can be used to assess each user's tolerance.
  - When characterization is not confident, take the default action, i.e.,
    - Ask the user to **say his/her command in another way**
      - rather than providing a list of random options for user to choose from
        - which can be annoying or make the user lose confidence in the system!

Outline

- Continual or lifelong learning
- Class incremental learning (CIL)
  - Theoretical result and algorithms
- Open world continual learning
  - A continuous learning chatbot
- Continual pre-training of language models
- Summary

Continual pre-training of language models

- Language models (LMs) once trained are hard to be changed.
  - But for a specific domain, fine-grained data may be available to make the language model do better.
- Goal: (1) incrementally pre-train an LM with a sequence of domain corpora, and (2) achieve knowledge transfer across domains
- key challenge: how to preserve the knowledge already in the LM (i.e., deal with forgetting/CF) and encourage KT.
  - We propose a method to do so based on LM robustness (making use of dropout masks) and network pruning to identify important parameters in the LM to be protected using **soft masks**.

Ke, Shao, Lin, Konishi, Kim and Liu. CONTINUALPRE-TRAININGOFLANGUAGEMODELS. ICLR-2023.

End-task evaluation results

Category	Domain / Model	Restaurant MF1	Restaurant Acc	ACL MF1	ACL Acc	AI MF1	AI Acc	Phone MF1	Phone Acc	PubMed MF1	Camera MF1	Camera Acc	Average MF1	Average Acc	Forget R. MF1	Forget R. Acc
Non-CL	Pool	80.96	87.80	69.69	74.11	68.55	75.97	84.96	86.95	73.34	86.03	90.83	77.25	81.50	—	
	RoBERTa	79.81	87.00	66.11	71.26	60.98	71.85	83.75	86.08	72.38	78.82	87.03	73.64	79.27	—	
	DAP-RoBERTa	80.84	**87.68**	68.75	73.44	68.97	75.95	82.59	85.50	72.84	84.39	89.90	76.40	80.89	—	
	DAP-Adapter	80.19	87.14	68.87	72.92	60.55	71.38	82.71	85.35	71.68	83.62	89.23	74.60	79.62	—	
	DAP-Prompt	79.00	86.45	66.66	71.35	61.47	72.36	84.17	86.53	**73.09**	85.52	90.38	74.98	80.03	—	
CL DAP-train	NCL	79.52	86.54	68.39	72.87	67.94	75.71	84.10	86.33	72.49	85.71	90.70	76.36	80.77	1.14	1.05
	NCL-Adapter	80.13	87.05	67.39	72.30	57.71	69.87	83.32	85.86	72.07	83.70	89.71	74.05	79.48	0.15	-0.02
	DEMIX	79.99	87.12	68.46	72.73	63.35	72.86	78.07	82.42	71.73	86.59	91.12	74.70	79.66	0.74	0.36
	BCL	78.97	86.52	**70.71**	**74.58**	66.26	74.55	81.70	84.63	71.99	85.06	90.51	75.78	80.46	-0.06	-0.19
	CLASSIC	79.89	87.05	67.30	72.11	59.84	71.08	84.02	86.22	69.83	86.93	91.25	74.63	79.59	0.44	0.25
	KD	78.05	85.59	69.17	73.73	67.49	75.09	82.12	84.99	72.28	81.91	88.69	75.17	80.06	-0.07	0.01
	EWC	**80.98**	87.64	65.94	71.17	65.04	73.58	82.32	85.13	71.43	83.35	89.14	74.84	79.68	0.02	-0.01
	DER++	79.00	86.46	67.20	72.16	63.96	73.54	83.22	85.61	72.58	87.10	91.47	75.51	80.30	2.36	1.53
	HAT	76.42	85.16	60.70	68.79	47.37	65.69	72.33	79.13	69.97	74.04	85.14	66.80	75.65	-0.13	-0.29
	HAT-All	74.94	83.93	52.08	63.94	34.16	56.07	64.71	74.43	68.14	65.54	81.44	59.93	71.33	3.23	1.83
	HAT-Adapter	79.29	86.70	68.25	72.87	64.84	73.67	81.44	84.56	71.61	82.37	89.27	74.63	79.78	-0.23	-0.18
	DAS	80.34	87.16	69.36	74.01	**70.93**	**77.46**	**85.99**	**87.70**	72.80	**88.16**	**92.30**	**77.93**	**81.91**	**-1.09**	**-0.60**

Ke, Shao, Lin, Konishi, Kim and Liu. CONTINUALPRE-TRAININGOFLANGUAGEMODELS. ICLR-2023.

Outline

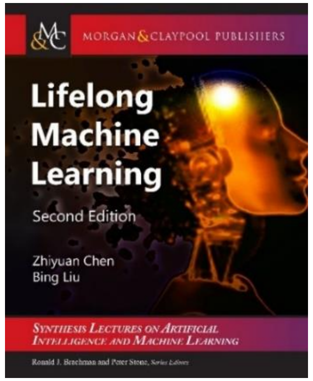
- Continual or lifelong learning
- Class incremental learning (CIL)
  - Theoretical result and algorithms
- Open world continual learning
  - A continuous learning chatbot
- Continual pre-training of language models
- Summary

Summary

- CIL is learnable, and good OOD detection (which also does WP) for each task is *necessary* and *sufficient* for good CIL.
  - The theory unifies CIL, OOD detection and open world learning
- To learn well, CIL needs to consider the past and the unknown future. What are the implications of it?
  - What is the best architecture and what features should be learned?
- Learning in AGI
  - The learning process must be autonomous
    - SOLA: Self-initiated open world continual learning and adaptation

Thank You

Q&A



Students: Zhiyuan Chen (ex), Sepideh Esmaeilpour, Zixuan Ke, Gyuhak Kim, Nianzu Ma, Sahisnu Mazumder (ex), Lei Shu (ex), Hu Xu (ex)

Collaborators: Wenpeng Hu, Scott Grigsby, Yiduo Guo, Tatsuya Konishi, Haowei Lin, Eric Robertson, Yijia Shao, and numerous researchers in the DARPA project.

Funding: