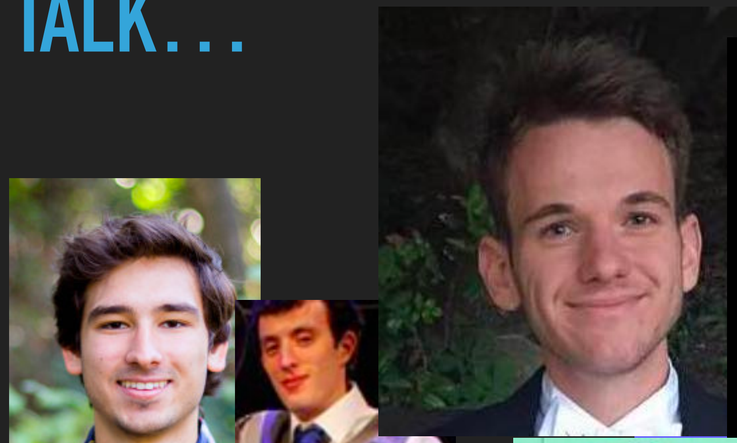
AN INTRODUCTION TO

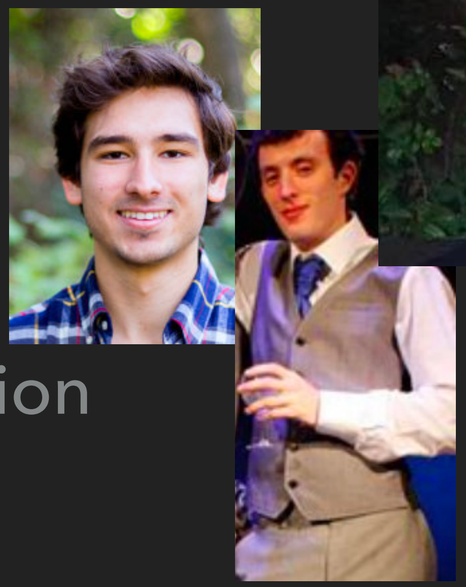
CONTINUAL LEARNING

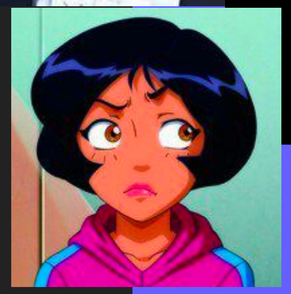
By Kate Highnam

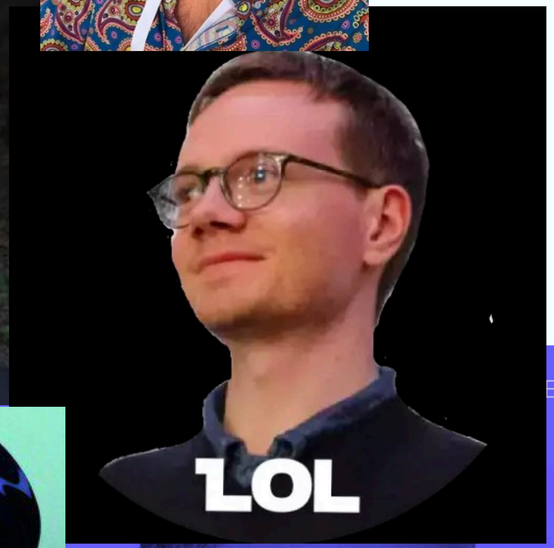
INCLUDED IN TODAY'S TALK…

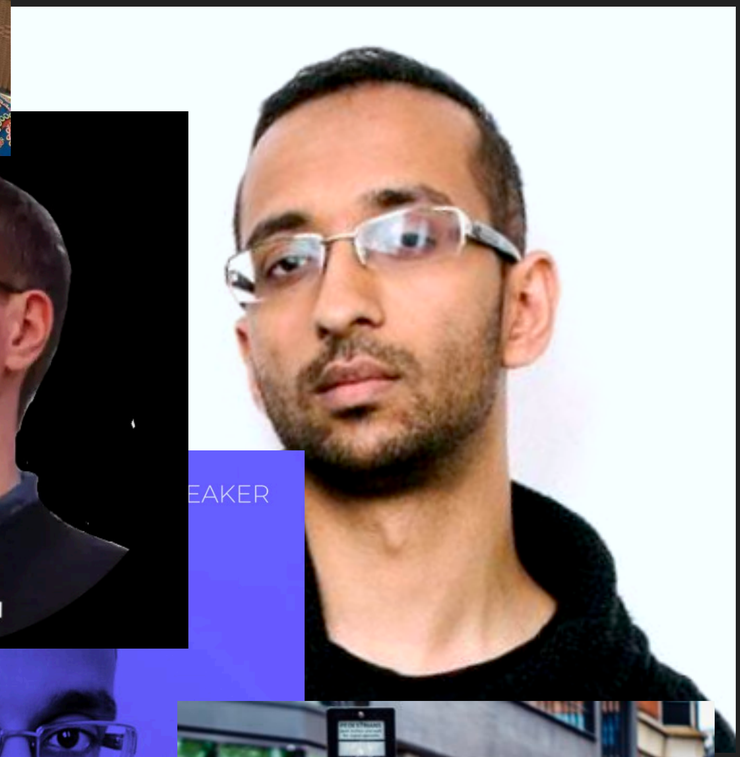
- Entertainment
- Mockery
- Memes
- Technical information











PLEASE INTERRUPT MY RAMBLING

WHAT IS THIS NEW MACHINE LEARNING BUZZ TERM?

Every ML Researcher

CONTINUAL LEARNING

- “… the problem of learning from an infinite stream of data, with the goal of gradually extending acquired knowledge and using it for future learning [4].” (De Lange, et al. 2019)
  - Predominantly in the deep learning community!
- AKA lifelong learning, never ending learning, sequential learning, incremental learning…

DOES THIS SOUND FAMILIAR? 🤔

SIMILAR FIELDS

MULTI-TASK LEARNING

ONLINE LEARNING

TRANSFER LEARNING

OPEN WORLD LEARNING

DOMAIN ADAPTATION

META-LEARNING

…

De Lange, et al. 2019

SIMILAR FIELDS

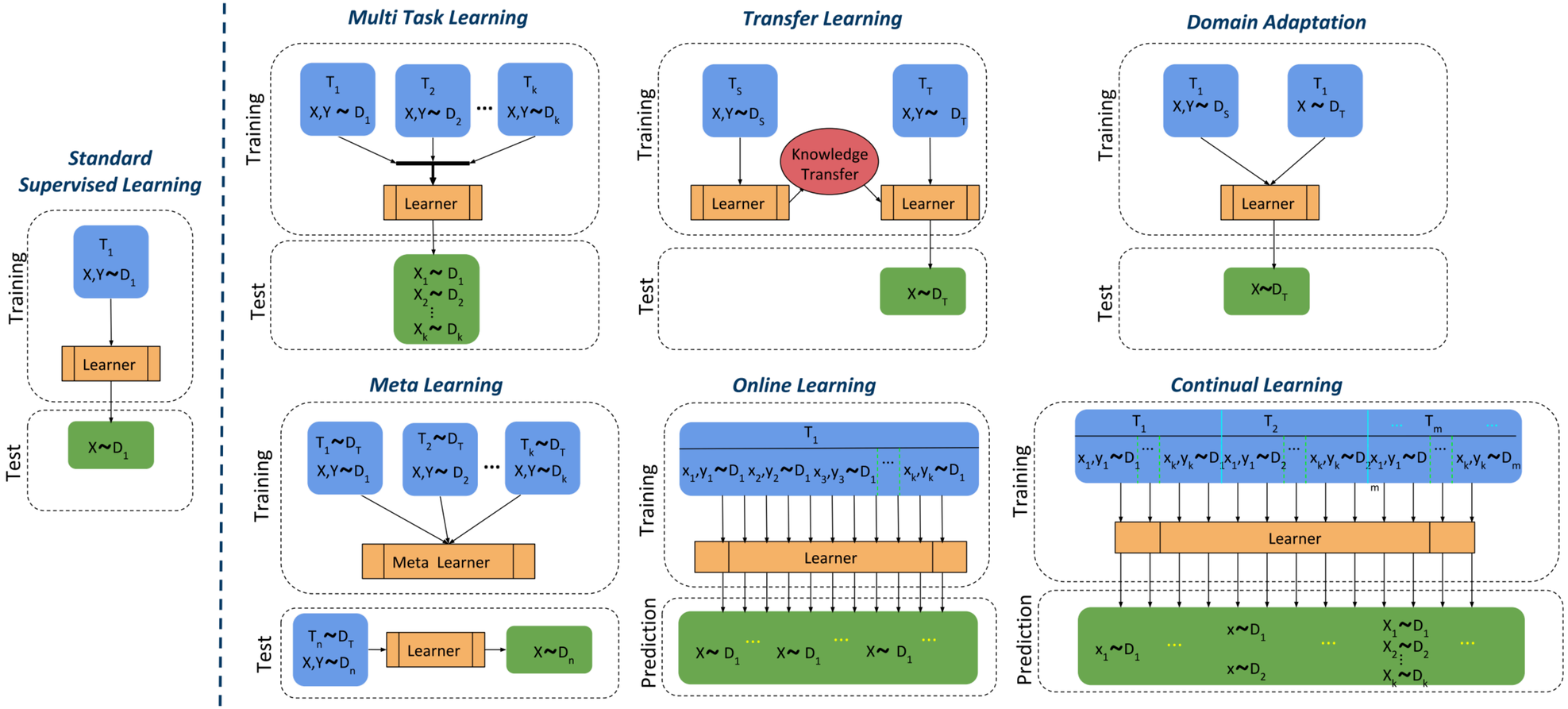
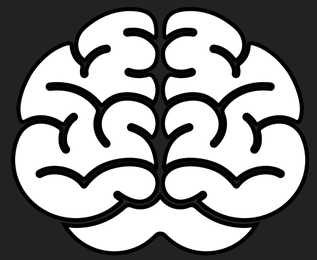
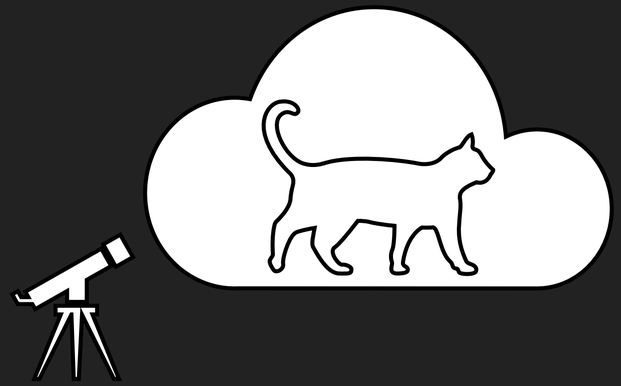


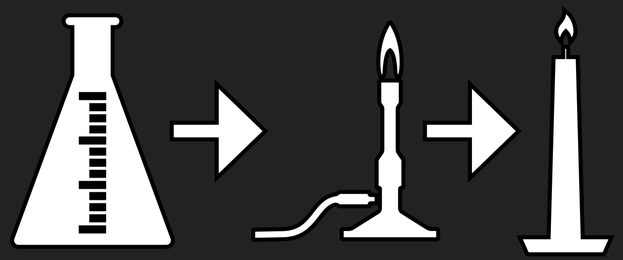
Fig. 5: The main setup of each related machine learning field, illustrating the differences with general continual learning settings.

De Lange, et al. 2019

CONTINUAL LEARNING TARGETS





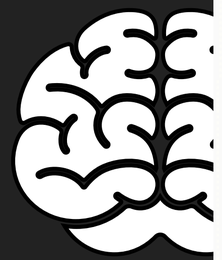


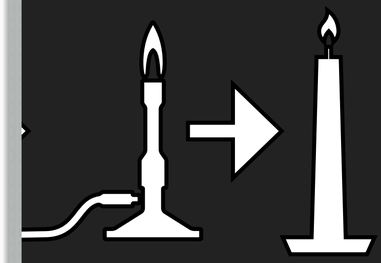
HANDLING MEMORIES

CATASTROPHIC FORGETTING

DATA DISTRIBUTION SHIFTS

CONTINUAL LEA



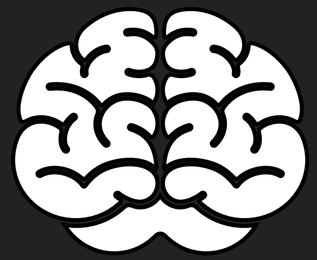


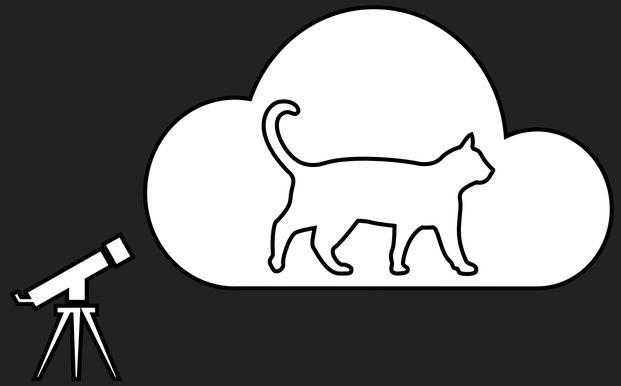
HANDLING ME

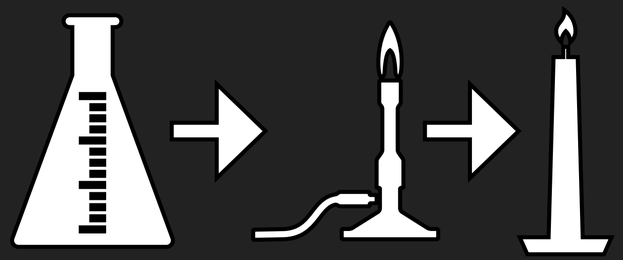
RIBUTION SHIFTS

It's all about the balance

CONTINUAL LEARNING TARGETS







HANDLING MEMORIES

CATASTROPHIC FORGETTING

DATA DISTRIBUTION SHIFTS

TASK INCREMENTAL LEARNING SETTING

- Data: $(X^{(t)}, Y^{(t)})$
- Data distribution: $D^{(t)}$
- Loss function: ℓ
- Parameters: θ
- Number of tasks: T
- Network function for task t: f_t
- For some task t, take $X^{(t)}$ samples from $D^{(t)}$ with $Y^{(t)}$ ground truth labels

Goal: Control the statistical risk of all seen task given limited or no access to the data $(X^{(t)}, Y^{(t)})$ from previous tasks $t < T$...

$$\sum_{t=1}^{T} \mathbb{E}_{(X^{(t)}, Y^{(t)})}[\ell(f_t(X^{(t)}; \theta), Y^{(t)})]$$

TASK INCREMENTAL LEARNING SETTING

- Marginal output and input distributions, respectively: $P(Y^{(t)})$, $P(X^{(t)})$ for task t

INCREMENTAL CLASS LEARNING

$$\{Y^{(t)}\} = \{Y^{(t+1)}\}$$

...but...

$$P(Y^{(t)}) \neq P(Y^{(t+1)})$$

...because...

$$P(X^{(t)}) \neq P(X^{(t+1)})$$

INCREMENTAL DOMAIN LEARNING

Guarantees...

$$P(Y^{(t)}) = P(Y^{(t+1)})$$

TASK INCREMENTAL LEARNING

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

...and...

$$P(Y^{(t)}) \neq P(Y^{(t+1)})$$

...because...

$$P(X^{(t)}) \neq P(X^{(t+1)})$$

TASK INCREMENTAL LEARNING SETTING

- Marginal output and input distributions, respectively: $P(Y^{(t)})$, $P(X^{(t)})$ for task t

INCREMENTAL CLASS LEARNING

$$\{Y^{(t)}\} = \{Y^{(t+1)}\}$$

…but…

$$P(Y^{(t)}) \neq P(Y^{(t+1)})$$

…because…

$$P(X^{(t)}) \neq P(X^{(t+1)})$$

INCREMENTAL DOMAIN LEARNING

Guarantees…

$$P(Y^{(t)}) = P(Y^{(t+1)})$$

TASK INCREMENTAL LEARNING

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

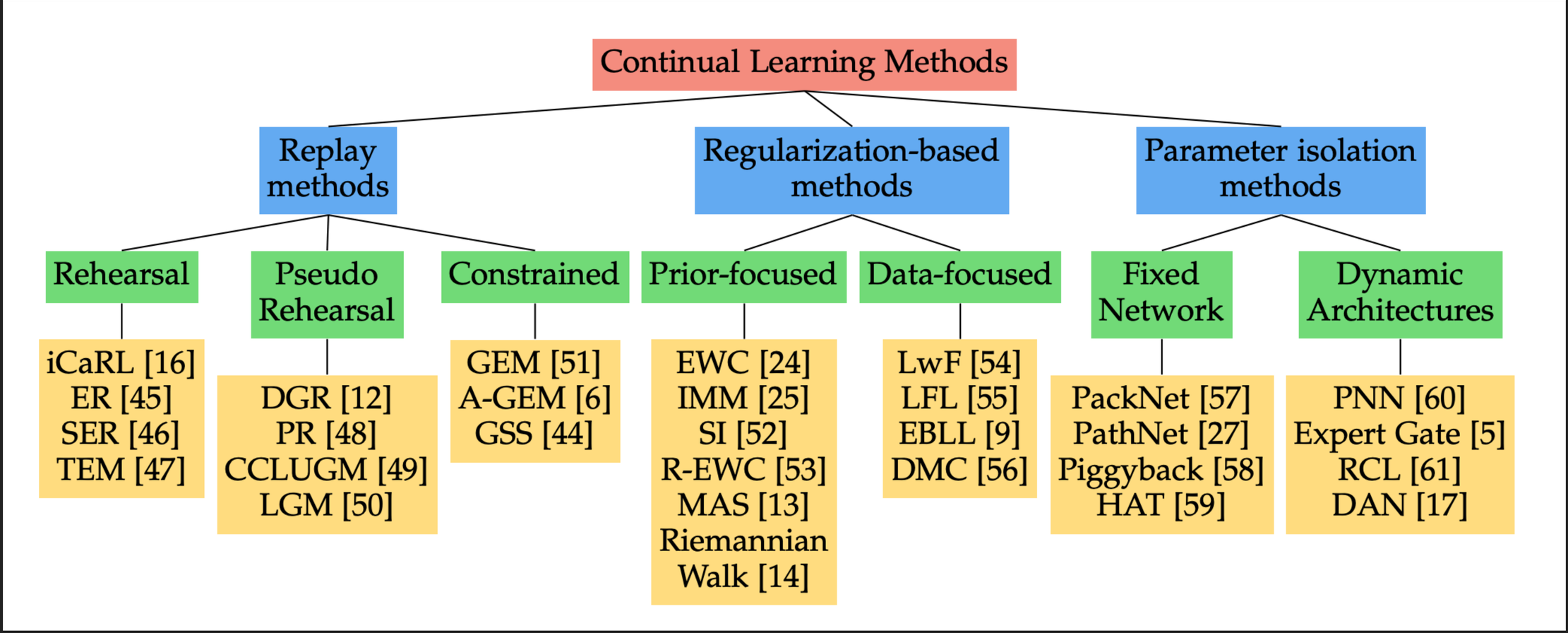
…and…

$$P(Y^{(t)}) \neq P(Y^{(t+1)})$$

…because…

$$P(X^{(t)}) \neq P(X^{(t+1)})$$

CONTINUAL LEARNING APPROACHES



De Lange, et al. 2019

REPLAY METHODS

- Stores samples in raw format or generates pseudo-samples with a generative model
- These previous task samples are replayed while learning a new task to alleviate forgetting

REPLAY METHODS

- Stores samples in raw format or generates pseudo-samples with a generative model
- These previous task samples are replayed while learning a new task to alleviate forgetting

REHEARSAL METHODS

CONSTRAINED OPTIMISATION

PSEUDO REHEARSAL

REPLAY METHODS

- Stores samples in raw format or generates pseudo-samples with a generative model
- These previous task samples are replayed while learning a new task to alleviate forgetting

REHEARSAL METHODS

Retrain on subset of stored samples while training on new tasks...

e.g. iCaRL [16]

Stores subset of exemplars per class, selecting best by means in learned feature space

CONSTRAINED OPTIMISATION

PSEUDO REHEARSAL

REPLAY METHODS

- Stores samples in raw format or generates pseudo-samples with a generative model
- These previous task samples are replayed while learning a new task to alleviate forgetting

REHEARSAL METHODS

Retrain on subset of stored samples while training on new tasks...

e.g. iCaRL [16]

Stores subset of exemplars per class, selecting best by means in learned feature space

CONSTRAINED OPTIMISATION

Constrain the new task updates to not interfere with previous tasks

e.g. GEM [51]

Project estimated gradient direction on the feasible region outlined by previous task gradients through first order Taylor series approximation

PSEUDO REHEARSAL

REPLAY METHODS

- Stores samples in raw format or generates pseudo-samples with a generative model
- These previous task samples are replayed while learning a new task to alleviate forgetting

REHEARSAL METHODS

Retrain on subset of stored samples while training on new tasks...

e.g. iCaRL [16]

Stores subset of exemplars per class, selecting best by means in learned feature space

CONSTRAINED OPTIMISATION

Constrain the new task updates to not interfere with previous tasks

e.g. GEM [51]

Project estimated gradient direction on the feasible region outlined by previous task gradients through first order Taylor series approximation

PSEUDO REHEARSAL

Output of previous model(s) given random inputs are used to approximate previous task samples

Not for deep networks and large input vectors; cannot cover the input space [48]

REGULARISATION-BASED METHODS

- Avoid storing raw inputs, prioritising privacy, and alleviating memory requirements
- Extra regularisation term introduced in the loss function, consolidating previous knowledge when learning on new data

REGULARISATION-BASED METHODS

- Avoid storing raw inputs, prioritising privacy, and alleviating memory requirements
- Extra regularisation term introduced in the loss function, consolidating previous knowledge when learning on new data

DATA-FOCUSED METHODS

PRIOR-FOCUSED METHODS

REGULARISATION-BASED METHODS

- Avoid storing raw inputs, prioritising privacy, and alleviating memory requirements
- Extra regularisation term introduced in the loss function, consolidating previous knowledge when learning on new data

DATA-FOCUSED METHODS

- Use previous task model outputs given new task input images
- e.g. incremental integration of shallow auto encoders to constrain task features in their corresponding learned low dimensional space
- e.g. LwF [54], EBLL [9]

PRIOR-FOCUSED METHODS

REGULARISATION-BASED METHODS

- Avoid storing raw inputs, prioritising privacy, and alleviating memory requirements
- Extra regularisation term introduced in the loss function, consolidating previous knowledge when learning on new data

DATA-FOCUSED METHODS

- Use previous task model outputs given new task input images
- e.g. incremental integration of shallow auto encoders to constrain task features in their corresponding learned low dimensional space
- e.g. LwF [54], EBLL [9]

PRIOR-FOCUSED METHODS

- Estimate a distribution over the model parameters, used as prior when learning from new data
- Parameters assumed independent, changes to important parameters penalised
- e.g. SI [52], MAS [13], EWC [24]

PARAMETER ISOLATION METHODS

PARAMETER ISOLATION METHODS

PARAMETER ISOLATION METHODS

- Dedicate different model parameters to each task, preventing any possible forgetting

PARAMETER ISOLATION METHODS

- Dedicate different model parameters to each task, preventing any possible forgetting
- If no constraints to architecture size, grow new branches for new tasks while freezing previous task parameters or dedicate a model copy to each task

PARAMETER ISOLATION METHODS

- Dedicate different model parameters to each task, preventing any possible forgetting
- If no constraints to architecture size, grow new branches for new tasks while freezing previous task parameters or dedicate a model copy to each task
- If architecture remains static, previous task parts are masked out during new task training, imposed at parameter level or “unit” level
  - Usually requires a “task oracle” activating corresponding mask or task branch during prediction
  - Usually with multi-head setup, this paper restricts to shared head between tasks

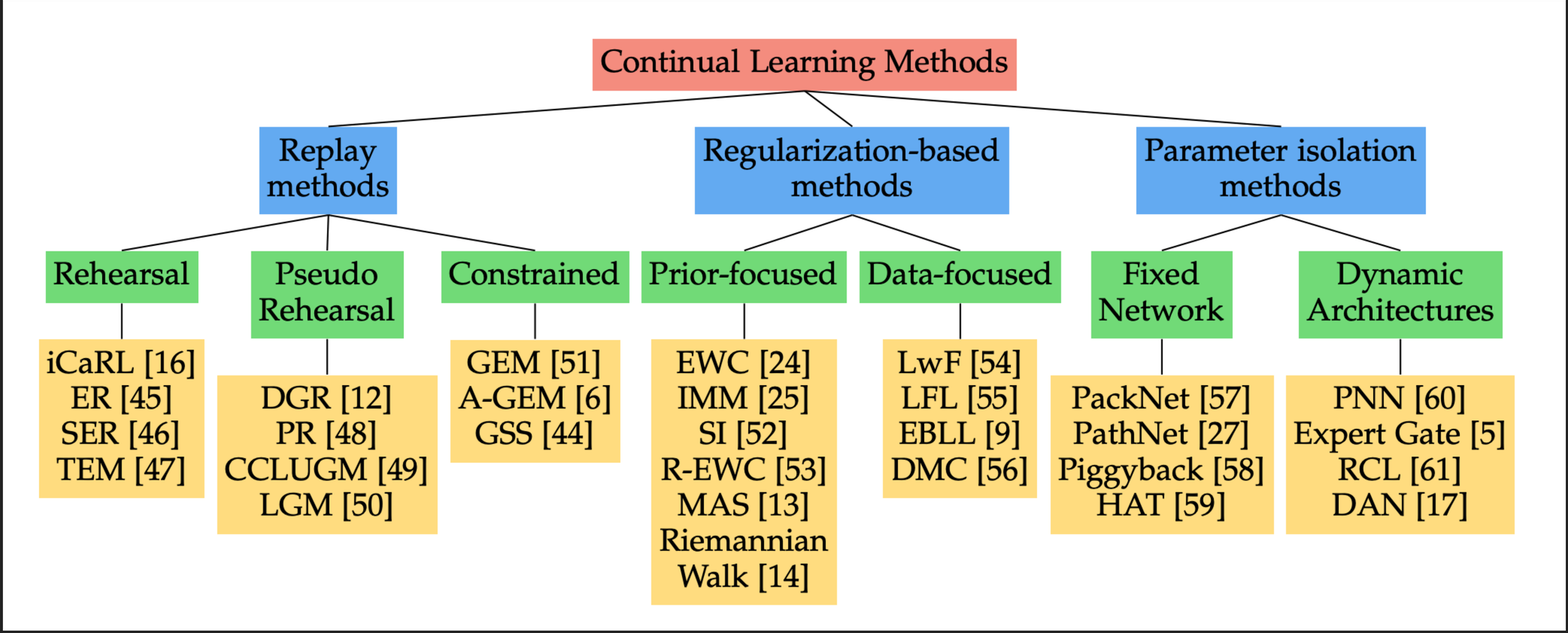
PARAMETER ISOLATION METHODS

- Dedicate different model parameters to each task, preventing any possible forgetting
- If no constraints to architecture size, grow new branches for new tasks while freezing previous task parameters or dedicate a model copy to each task
- If architecture remains static, previous task parts are masked out during new task training, imposed at parameter level or “unit” level
  - Usually requires a “task oracle” activating corresponding mask or task branch during prediction
  - Usually with multi-head setup, this paper restricts to shared head between tasks

PACKNET [57]

HAT [59]

CONTINUAL LEARNING APPROACHES



De Lange, et al. 2019

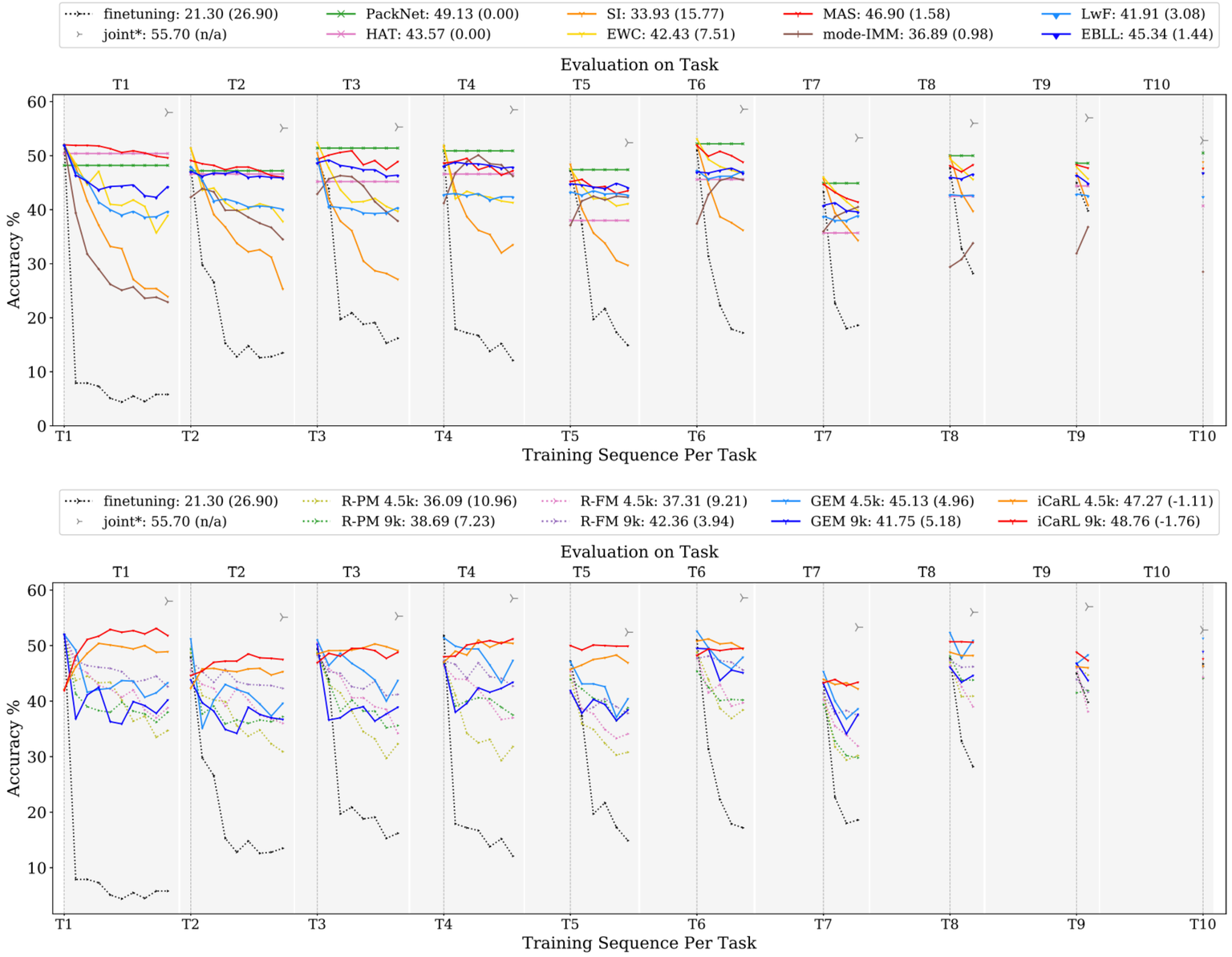
RESULTS FROM THE PAPER

- Models: Small, Base, Wide, Deep
- This is a hard problem! Accuracy doesn’t go over 50% much, if at all…

RESULTS FROM

- Models: Small, B
- This is a hard pro

Fig. 2: Parameter isolation and regularization-based methods (top) and replay methods (bottom) on Tiny Imagenet for the BASE model with random ordering, reporting average accuracy (forgetting) in the legend.



RESULTS FROM THE PAPER

- Replay: Wider not deeper… could this be true in other applications?
- Regularisation: Deeper is better, reg + replay is limiting
- Asymmetric capacity allocation, bad with unbalanced data
- In results, ordering of tasks appears agnostic, counter to initial assumption similar to “curriculum learning [98]”
  - “Implying knowledge is better captured starting with the general easier tasks followed by harder specific tasks. Instead, experimental results on both datasets exhibit ordering agnostic behaviours, corresponding to similar previous findings [99]”

RESULTS FROM THE PAPER

- Qualitative Comparisons: GPU Memory, computation time, task-agnostic & privacy
  - Some store raw images
  - Some eat up memory for size of growing model or storing examples

RESULTS FROM THE

- Qualitative Compar sk-agnostic & privacy
  - Some store raw i
  - Some eat up mer examples

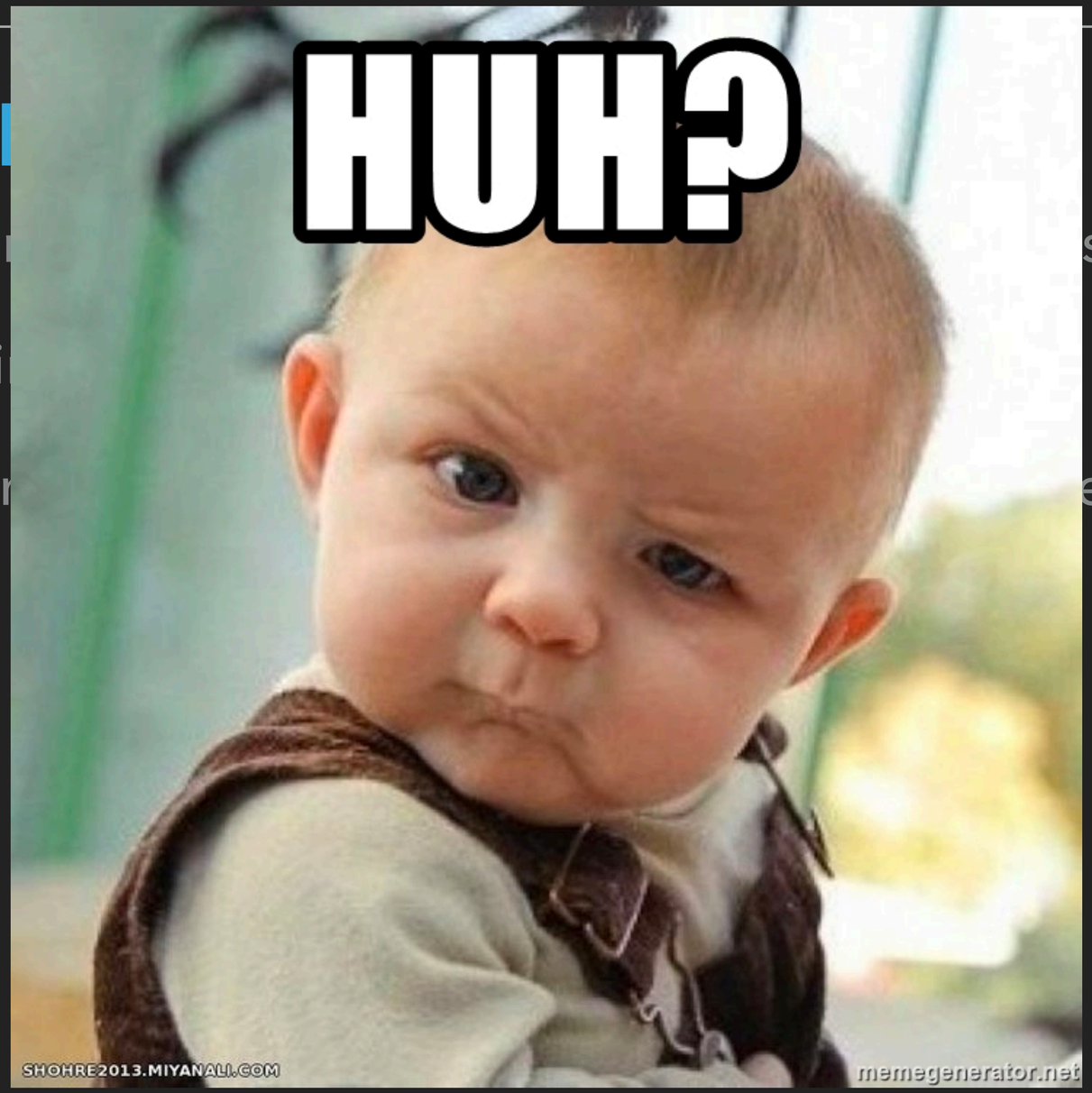


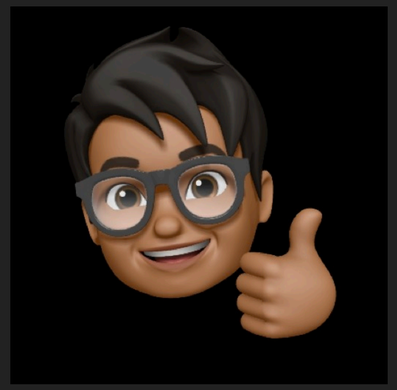
TABLE 10: Summary of our main findings. We report best results over all experiments, i.e. including regularization experiments for Tiny Imagenet. The SMALL, BASE, WIDE models are denoted as S,B,W, and weight decay as L2.

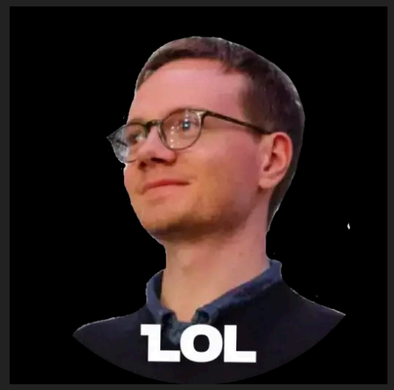
Method	Best Avg. Acc.			Suggested Regularizer	Suggested Model	Comments
	Tiny Imagenet	iNaturalist	RecogSeq			
Replay						- Least sensitive model capacity/regularization - Privacy issues storing raw images - No clear policy for unbalanced tasks
iCaRL 4.5k (9k)	48.55 (49.94)	x	x	dropout	S/B/W	- Lead performance - Designed for class incremental setup - Continual exemplar management - Nearest-neighbor classifier
GEM 4.5k (9k)	45.27 (44.23)	x	x	none/dropout	S/B/W	- Designed for online continual setup - Sensitive to amount of epochs
Regularization-based						
LwF	48.11	48.02	30.59	L2	W	- Invigorated by WIDE model - Requires sample outputs on previous model - Forgetting buildup for older dissimilar tasks
EBLL	48.17	53.30	33.82	L2	W	- Margin over LwF - Autoencoder gridsearch for unbalanced tasks
SI	43.74	51.77	43.40	dropout/L2	B/W	- Efficient training time over EWC/MAS - Requires dropout or L2 (prone to overfitting) - Most affected by task ordering
EWC	45.13	54.02	42.01	none	S	- Invigorated by SMALL capacity model - Deteriorates on WIDE model
MAS	48.98	54.59	45.72	none	B/W	- Lead regularization-based performance - Hyperparameter robustness - Unsupervised importance weight estimation
mean-IMM	32.42	49.82	31.43	none	B/W	- mode-IMM outperforms mean-IMM
mode-IMM	42.41	55.61	34.45	none	B/W	- Both require additional merging step
Parameter isolation						- Efficient memory - Design prevents single-head setup
PackNet	55.96	60.61	64.88	dropout/L2	S/B/W	- Lead performance - No forgetting (after compression) - Requires retraining after compression - Heuristic parameter pruning
HAT	44.19	x	x	L2	B/W	- Nearly no forgetting (nearly binary masks) - End-to-end attention mask learning - Saturating low-level feature capacity

De Lange, et al. 2019

MY RAMBLING IS OVER…

THANK YOU FOR LISTENING





HAPPY BIRTHDAY JANITH!

CITATIONS

- De Lange, Matthias, et al. "Continual learning: A comparative study on how to defy forgetting in classification tasks." *arXiv preprint arXiv:1909.08383* 2.6 (2019).
- Lesort, Timoth'ee. "Continual Learning: Tackling Catastrophic Forgetting in Deep Neural Networks with Replay Processes." *arXiv preprint arXiv:2007.00487* (2020)
- Lomonaco, Vincenzo. “Why Continual Learning is the Key to Machine Intelligence” https://medium.com/continual-ai/why-continuous-learning-is-the-key-towards-machine-intelligence-1851cb57c308

REPLAY METHODS

- Stores samples in raw format or generates pseudo-samples with a generative model
- These previous task samples are replayed while learning a new task to alleviate forgetting
- **Rehearsal Methods**: explicitly retrain on a subset of stored samples while training on new tasks
  - Performance is upper bounded by joint training on previous and current tasks
  - Prone to overfitting the subset of stored samples
- **Constrained Optimisation**: only constrain the new task updates to not interfere with previous tasks
  - Project the estimated gradient direction on the feasible region outlined by previous task gradients through first order Taylor series approximation (GEM [51])
- **Pseudo rehearsal:** output of previous model(s) given random inputs are used to approximate previous task samples
  - Not for deep networks and large input vectors; cannot cover the input space [48]
  - Maybe use a generative model? Then extra complexity in training it continually, need to balance retrieved examples and avoid mode collapse…

REGULARISATION-BASED METHODS

- Avoid storing raw inputs, prioritising privacy, and alleviating memory requirements
- Extra regularisation term introduced in the loss function, consolidating previous knowledge when learning on new data
- **Data-focused methods:** use previous task model outputs given new task input images
  - e.g. facilitate incremental integration of shallow auto encoders to constrain task features in their corresponding learned low dimensional space
- **Prior-focused methods:** estimate a distribution over the model parameters, used as prior when learning from new data
  - Parameters assumed independent, changes to important parameters penalised