INCREMENTAL LEARNING IN IMAGE CLASSIFICATION

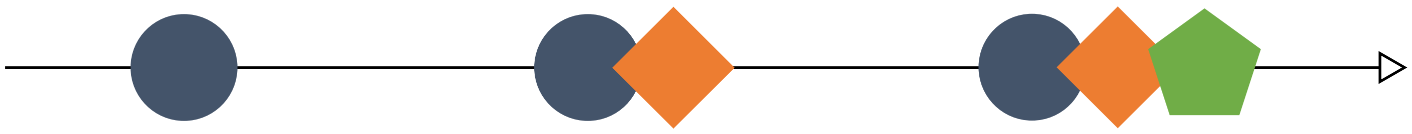
Manuele Macchia Francesco Montagna Giacomo Zema

Machine Learning and Deep Learning

A.Y. 2019/2020

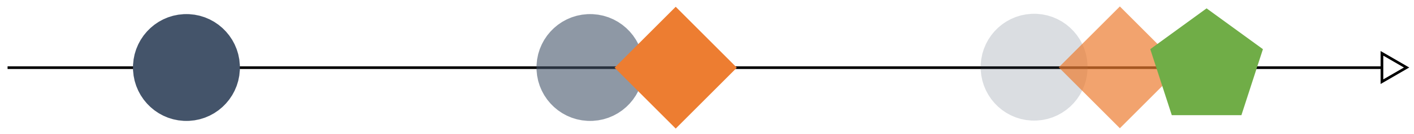
INCREMENTAL LEARNING

Incremental learning is a paradigm that allows **extending the knowledge** of an existing model, gradually incorporating new information



CATASTROPHIC FORGETTING

Training a model with new data interferes with previously acquired knowledge



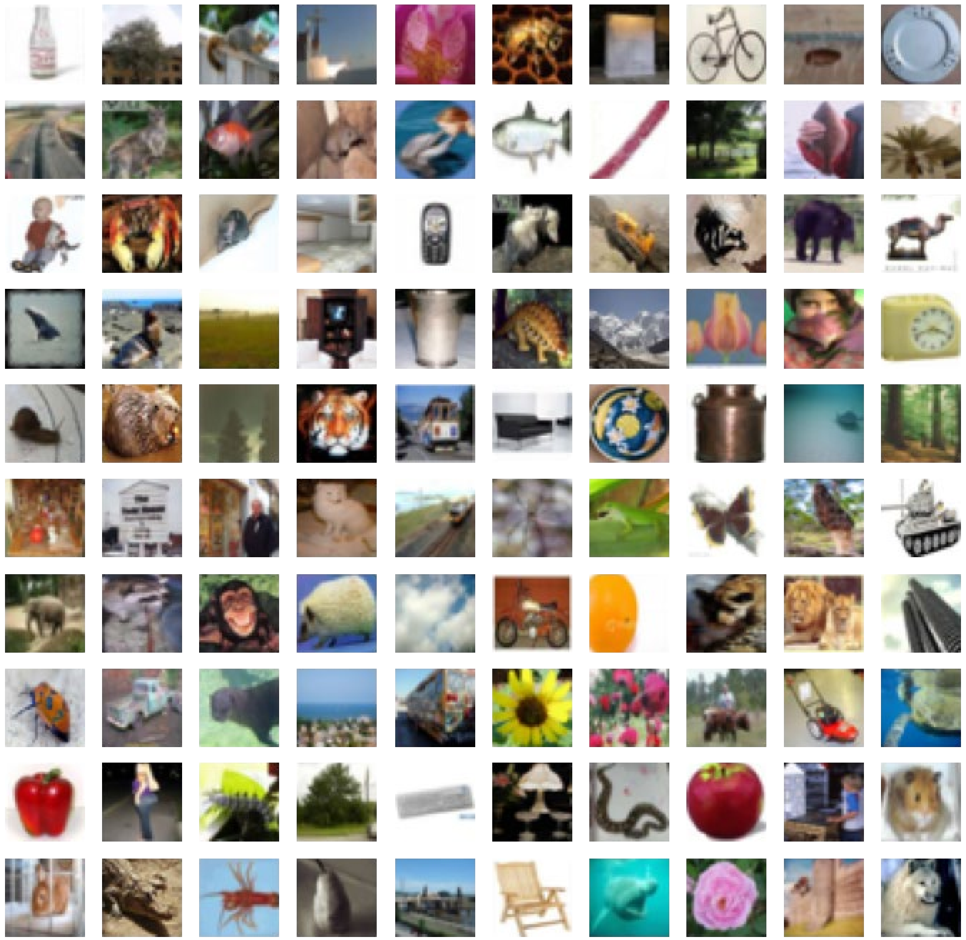
DATASET

CIFAR-100

- 100 classes
- 60’000 images
- 32 by 32 pixels

Incremental protocol

- 10 batches of 10 classes
- Model learns one batch at a time

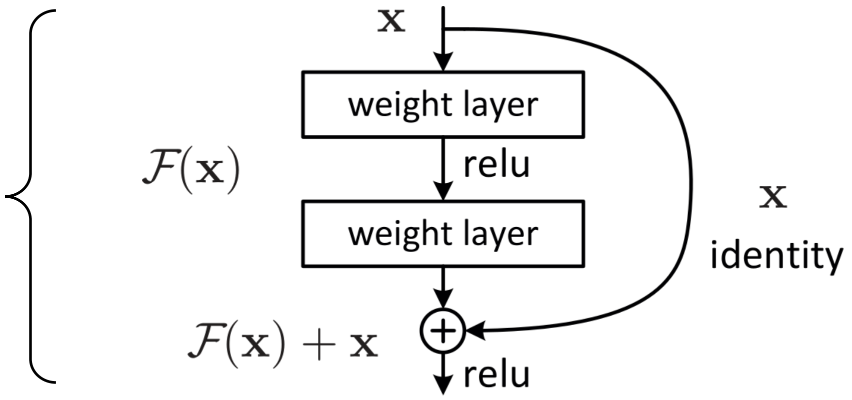


Krizhevsky “Learning Multiple Layers of Features from Tiny Images.” 2009.

MODEL

32-layers ResNet

Residual learning block



He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

BASELINES

BASELINES

Fine-tuning

LwF

iCaRL

Useful to understand
catastrophic forgetting effects

BASELINES

Fine-tuning

LwF

iCaRL

Distillation loss

$$\mathcal{L}_{BCE} = -\sum_{i=1}^{s-1} y_i \log g_i(x) + (1 - y_i) \log (1 - g_i(x))$$

Zhizhong and Hoiem. “Learning without forgetting.” *IEEE transactions on pattern analysis and machine intelligence*. 2017.

BASELINES

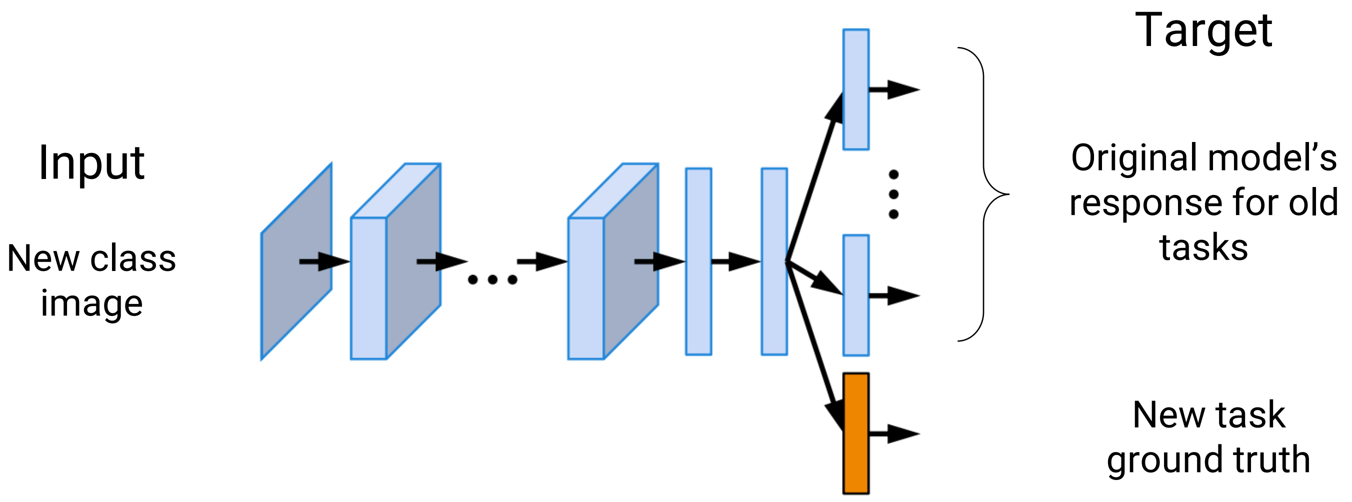
Fine-tuning

LwF

iCaRL

Distillation loss

$$\mathcal{L}_{BCE} = -\sum_{i=1}^{s-1} y_i \log g_i(x) + (1 - y_i) \log (1 - g_i(x))$$



Zhizhong and Hoiem. "Learning without forgetting." *IEEE transactions on pattern analysis and machine intelligence*. 2017.

BASELINES

Fine-tuning

LwF

iCaRL

Exemplars

Fixed-size memory containing samples of previous classes

$$K = 2000$$

Rebuffi, Sylvestre-Alvise, et al. "iCaRL: Incremental classifier and representation learning." *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2017.

BASELINES

Fine-tuning

LwF

iCaRL

Exemplars

Fixed-size memory containing samples of previous classes

$$K = 2000$$

Nearest-mean-of-exemplars classifier

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

Rebuffi, Sylvestre-Alvise, et al. "iCaRL: Incremental classifier and representation learning." *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2017.

BASELINES

Fine-tuning

LwF

iCaRL

Exemplars

Fixed-size memory containing samples of previous classes

$$K = 2000$$

Nearest-mean-of-exemplars classifier

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

Rebuffi, Sylvestre-Alvise, et al. "iCaRL: Incremental classifier and representation learning." *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2017.

BASELINES

Fine-tuning

LwF

iCaRL

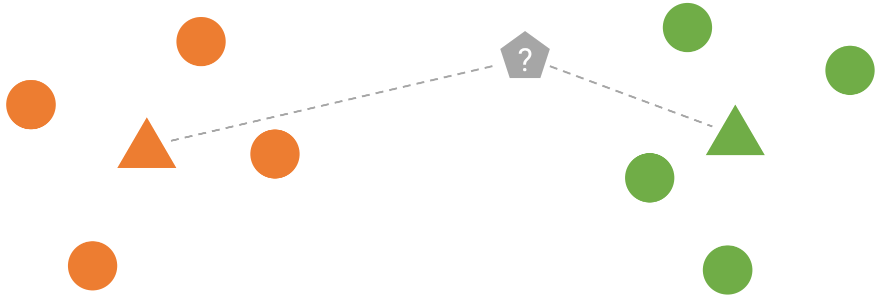
Exemplars

Fixed-size memory containing samples of previous classes

$$K = 2000$$

Nearest-mean-of-exemplars classifier

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



Rebuffi, Sylvestre-Alvise, et al. "iCaRL: Incremental classifier and representation learning." *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2017.

BASELINES

Fine-tuning

LwF

iCaRL

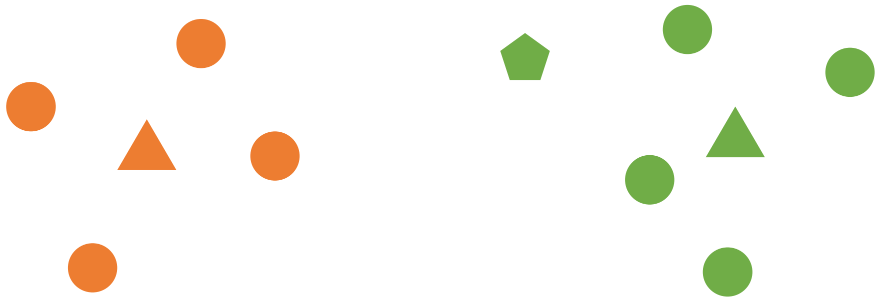
Exemplars

Fixed-size memory containing samples of previous classes

$$K = 2000$$

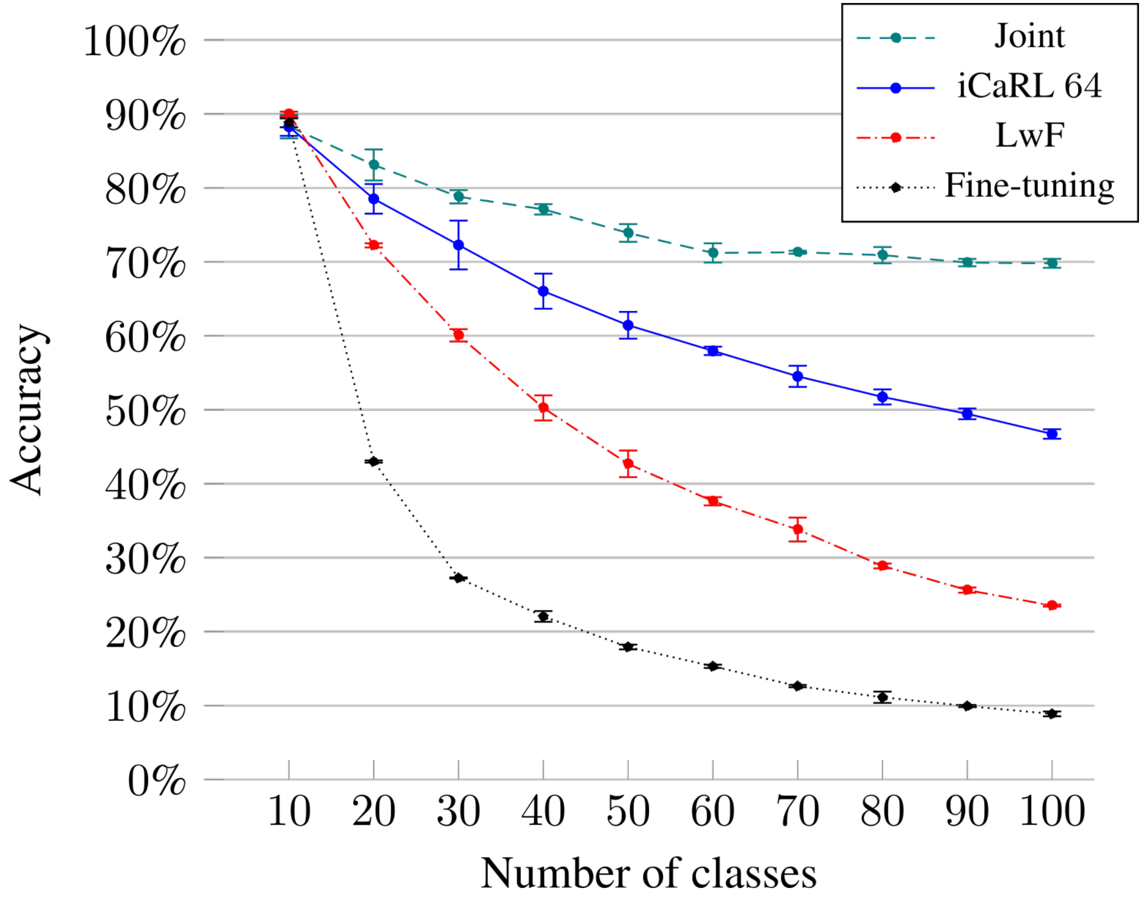
Nearest-mean-of-exemplars classifier

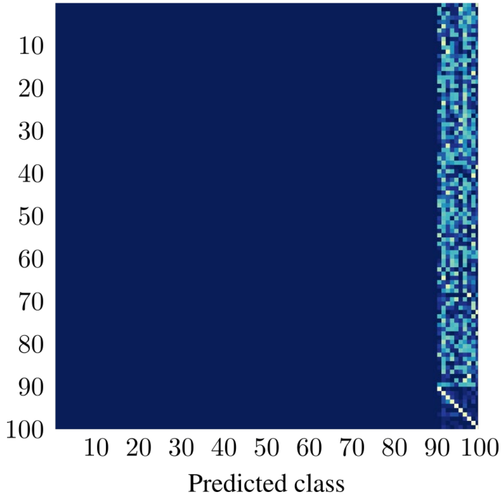
$$y^* \leftarrow \underset{y=1,\ldots,t}{\operatorname{argmin}} \|\varphi(x) - \mu_y\|$$



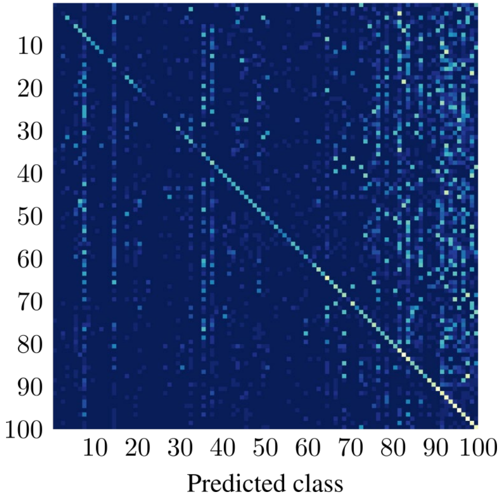
Rebuffi, Sylvestre-Alvise, et al. "iCaRL: Incremental classifier and representation learning." *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2017.

RESULTS

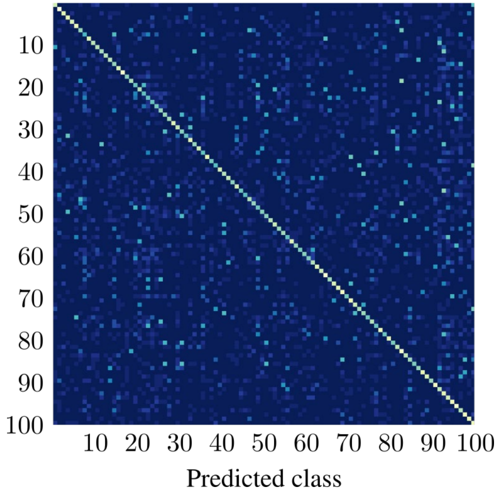




Fine-tuning



LwF



iCaRL

LOSS

- Observe model behaviour with different loss combinations
- Understand limitations of existing frameworks

LOSS

LWF

Asymmetric + BCE

ICARL

CE + BCE

CE + KD

Asymmetric + BCE

Asymmetric + L2

Asymmetric classification loss

$$\mathcal{L}_{asym} = \sum_{i=s}^{t} -y_i \log g_i(x) + (1 - y_i)\ (g_i(x))^2$$

LOSS

LWF

Asymmetric + BCE

ICARL

CE + BCE

CE + KD

Asymmetric + BCE

Asymmetric + L2

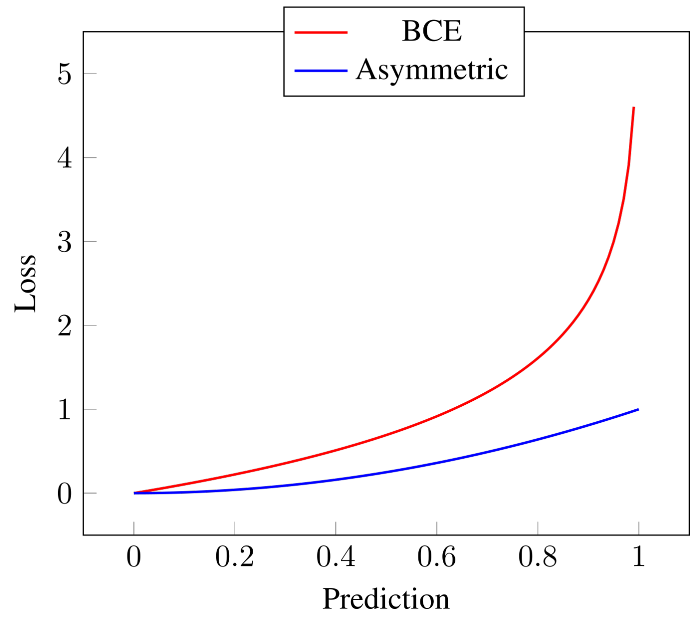
Asymmetric classification loss

$$\mathcal{L}_{asym} = \sum_{i=s}^{t} -y_i \log g_i(x) + (1 - y_i)\ (g_i(x))^2$$

Differences

- Less penalty than BCE for possibly informative non-zero outputs

Penalty of $y = 0$ targets



LOSS

LWF

Asymmetric + BCE

ICARL

CE + BCE

CE + KD

Asymmetric + BCE

Asymmetric + L2

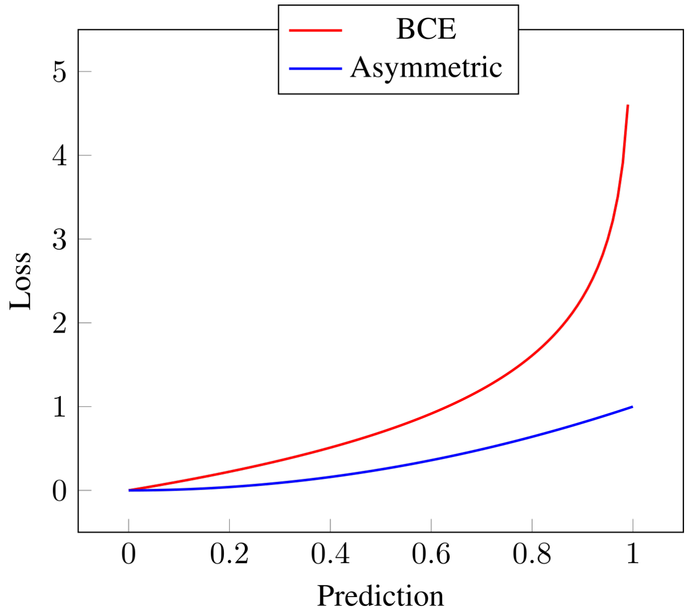
Asymmetric classification loss

$$\mathcal{L}_{asym} = \sum_{i=s}^{t} -y_i \log g_i(x) + (1 - y_i)\ (g_i(x))^2$$

Differences

- Less penalty than BCE for possibly informative non-zero outputs
- Less imbalance than CE between classification and distillation loss contribution

Penalty of $y = 0$ targets



LOSS

LWF

Asymmetric + BCE

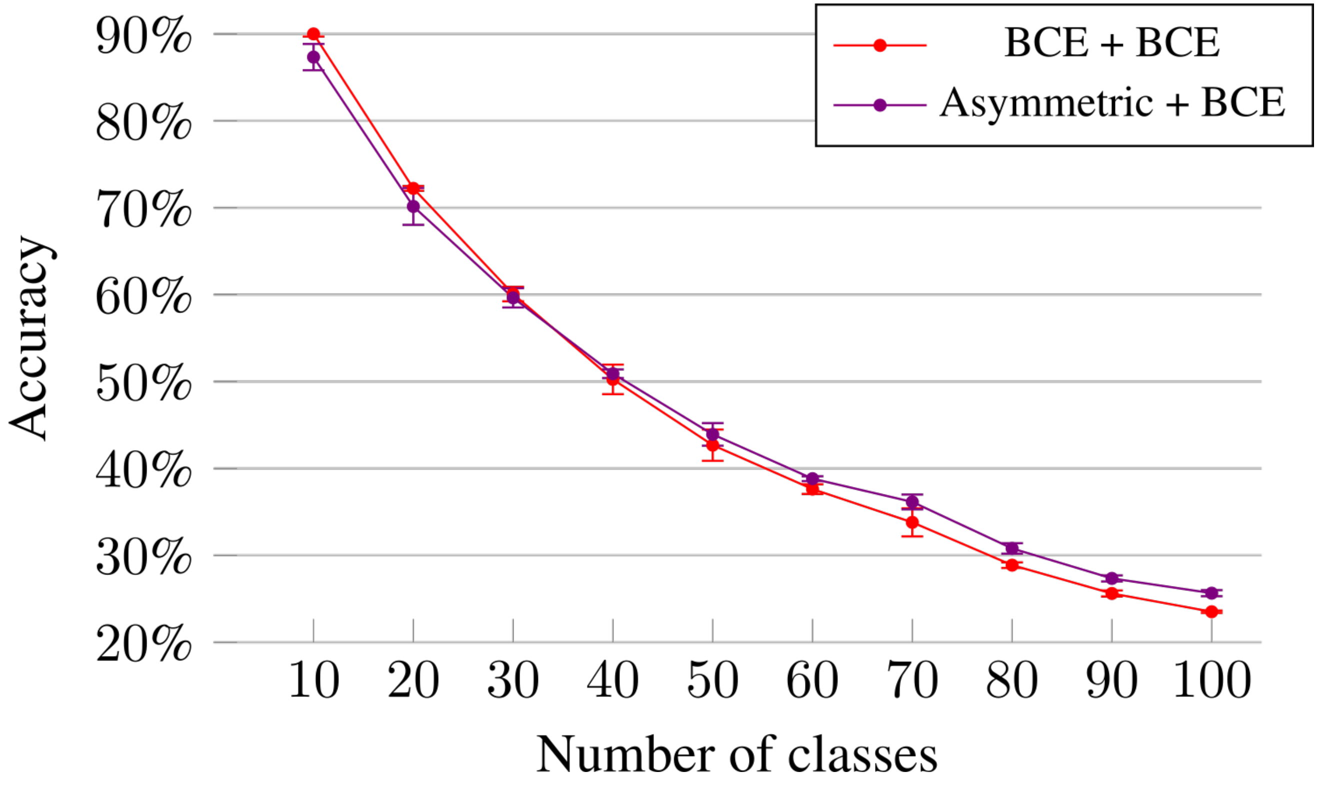
ICARL

CE + BCE

CE + KD

Asymmetric + BCE

Asymmetric + L2



LOSS

LWF

Asymmetric + BCE

ICARL

CE + BCE

CE + KD

Asymmetric + BCE

Asymmetric + L2

Cross entropy classification loss

$$\mathcal{L}_{CE} = -\sum_{i=s}^{t} y_i \log g_i(x)$$

Binary cross entropy distillation loss

$$\mathcal{L}_{BCE} = -\sum_{i=1}^{s-1} y_i \log g_i(x) + (1 - y_i) \log (1 - g_i(x))$$

LOSS

LWF

Asymmetric + BCE

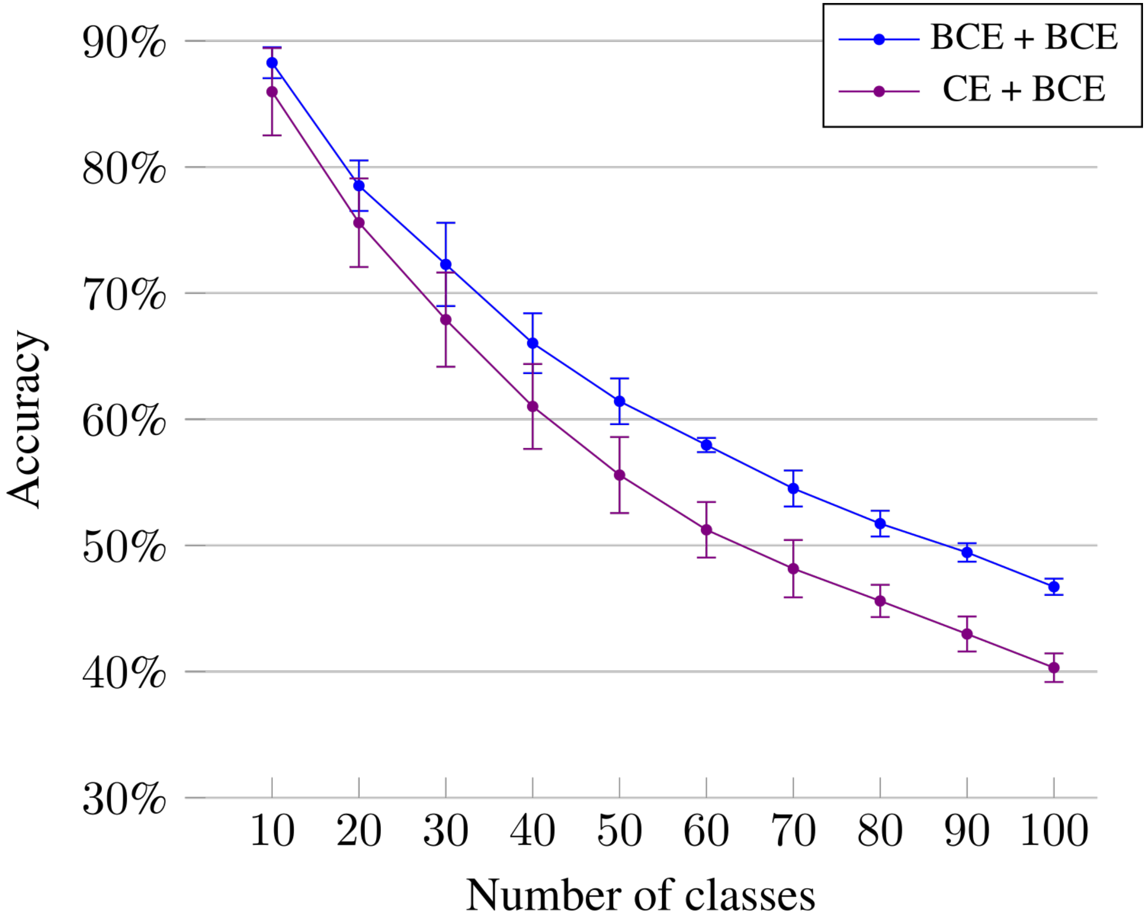
ICARL

CE + BCE

CE + KD

Asymmetric + BCE

Asymmetric + L2



The CE classification contribution loses importance as more learning steps are taken

LOSS

LWF

Asymmetric + BCE

ICARL

CE + BCE

CE + KD

Asymmetric + BCE

Asymmetric + L2

Cross entropy classification loss

$$\mathcal{L}_{CE} = -\sum_{i=s}^{t} y_i \log g_i(x)$$

Knowledge distillation loss

$$\mathcal{L}_{KD} = -\sum_{i=1}^{s-1} y'_i \log g'_i(x)$$

$$y'_i = \frac{y_i^{1/T}}{\sum_j y_j^{1/T}}, \qquad g'_i(x) = \frac{(g_i(x))^{1/T}}{\sum_j (g_j(x))^{1/T}}. \qquad T = 2$$

LOSS

LWF

Asymmetric + BCE

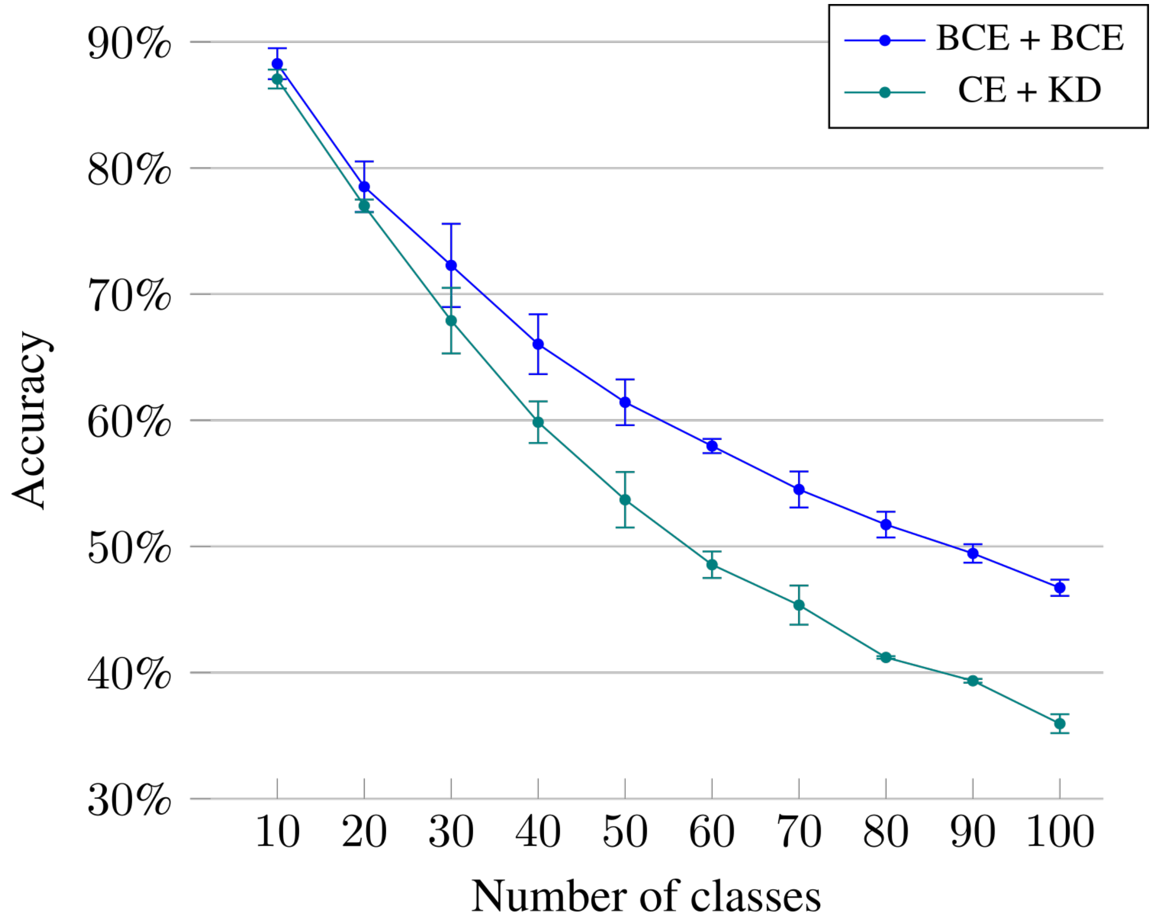
ICARL

CE + BCE

CE + KD

Asymmetric + BCE

Asymmetric + L2



LOSS

LWF

Asymmetric + BCE

ICARL

CE + BCE

CE + KD

Asymmetric + BCE

Asymmetric + L2

Asymmetric classification loss

$$\mathcal{L}_{asym} = \sum_{i=s}^{t} -y_i \log g_i(x) + (1 - y_i)\ (g_i(x))^2$$

Binary cross entropy distillation loss

$$\mathcal{L}_{BCE} = -\sum_{i=1}^{s-1} y_i \log g_i(x) + (1 - y_i) \log (1 - g_i(x))$$

LOSS

LWF

Asymmetric + BCE

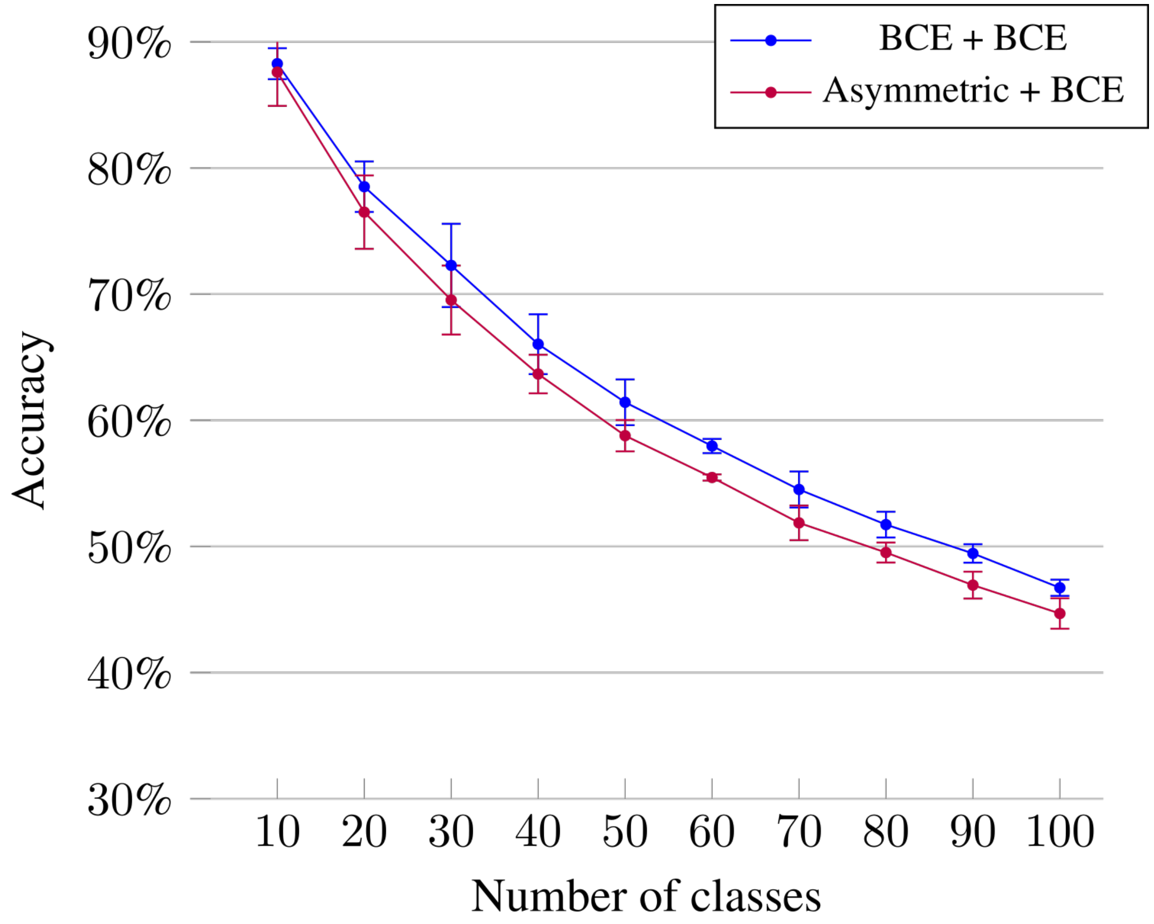
ICARL

CE + BCE

CE + KD

Asymmetric + BCE

Asymmetric + L2



LOSS

LWF

Asymmetric + BCE

ICARL

CE + BCE

CE + KD

Asymmetric + BCE

Asymmetric + L2

Asymmetric classification loss

$$\mathcal{L}_{asym} = \sum_{i=s}^{t} -y_i \log g_i(x) + (1 - y_i)\ (g_i(x))^2$$

L2 distillation loss

$$\mathcal{L}_{L2} = \sum_{i=1}^{s-1} (g_i(x) - y_i)^2$$

LOSS

LWF

Asymmetric + BCE

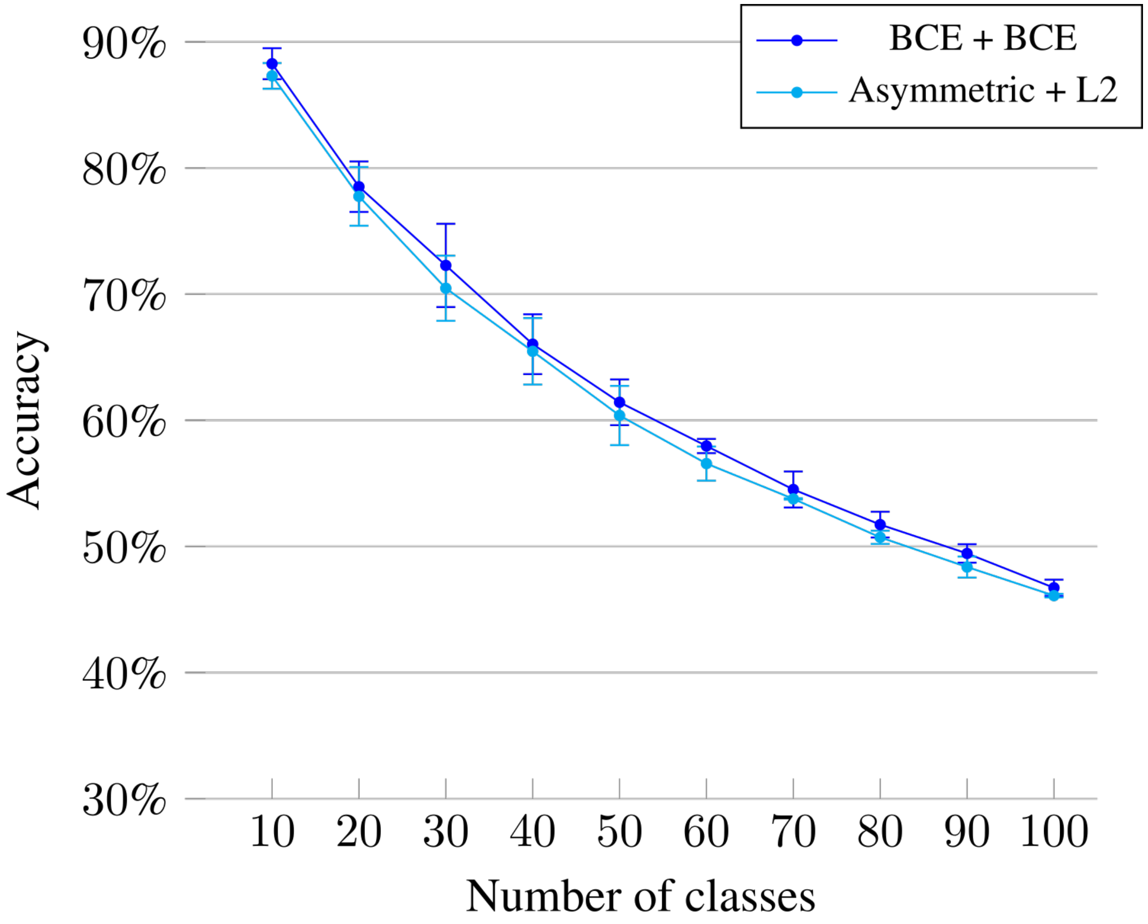
ICARL

CE + BCE

CE + KD

Asymmetric + BCE

Asymmetric + L2



CLASSIFIER

- Evaluate performance of different classifiers
  - Possibly improving performance

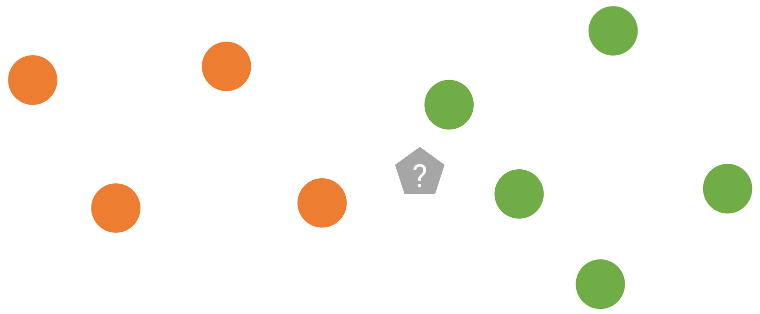
CLASSIFIER

K-nearest neighbors

Cosine similarity

Random forest

Instance-based learning algorithm



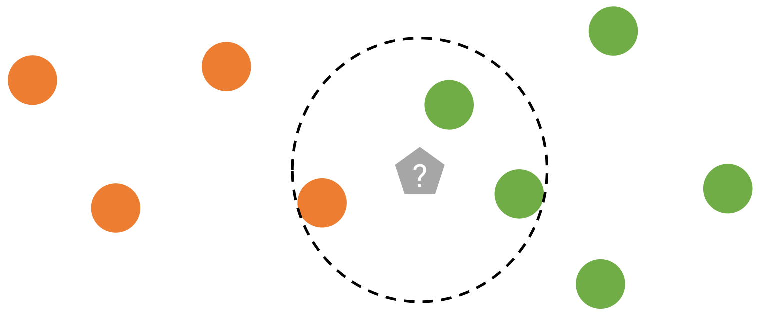
CLASSIFIER

K-nearest neighbors

Cosine similarity

Random forest

Instance-based learning algorithm



$K = 3$

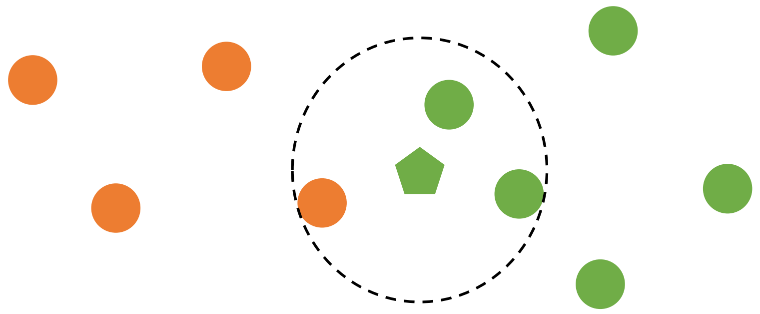
CLASSIFIER

K-nearest neighbors

Cosine similarity

Random forest

Instance-based learning algorithm



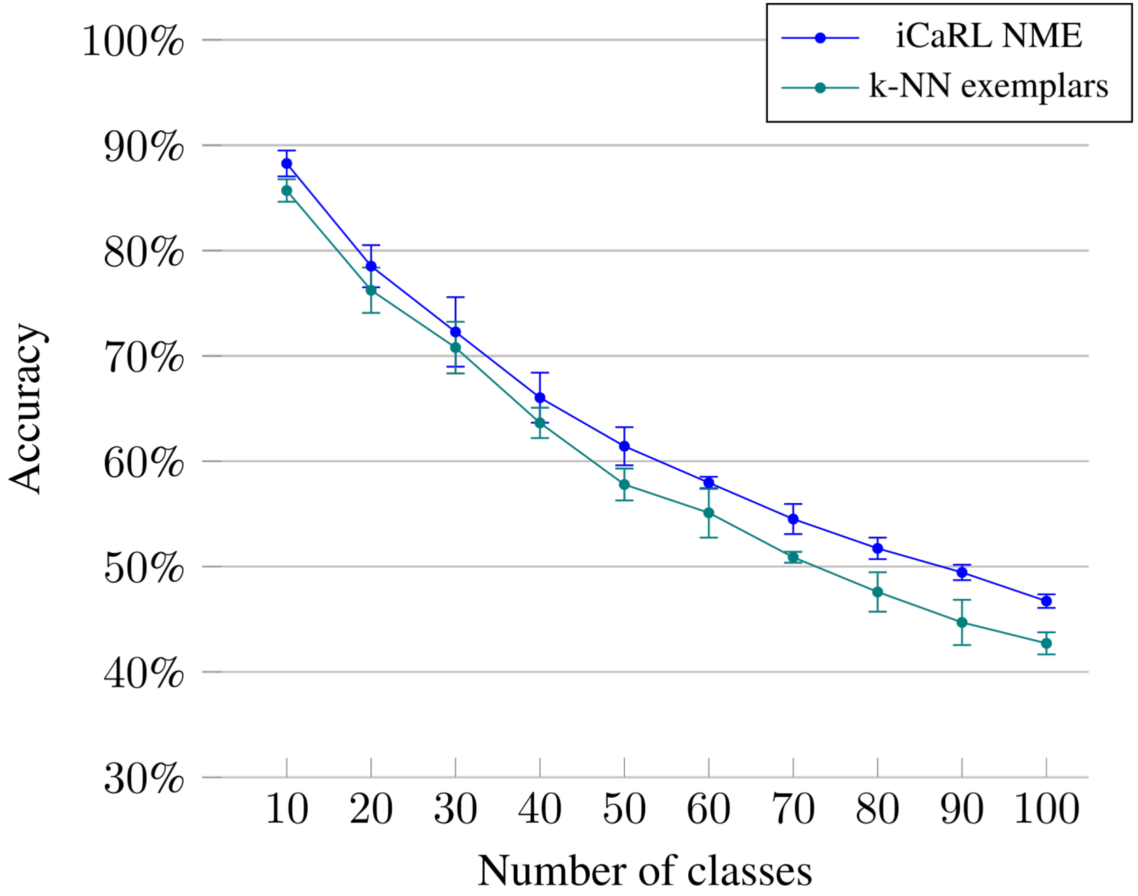
$K = 3$

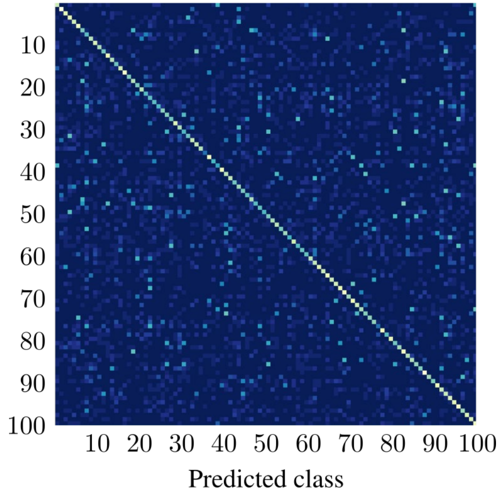
CLASSIFIER

K-nearest neighbors

Cosine similarity

Random forest





k-NN exemplars

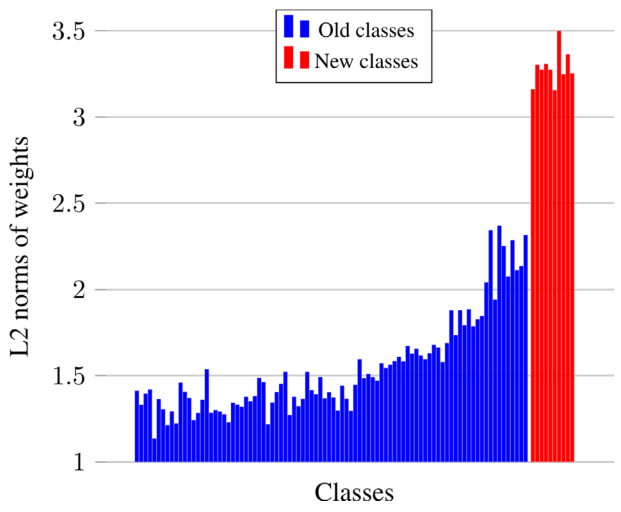
CLASSIFIER

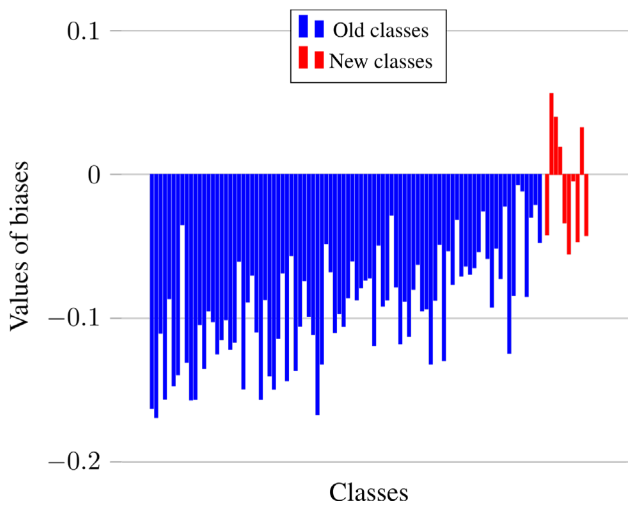
K-nearest neighbors

Cosine similarity

Random forest

Magnitude imbalance





Hou, Saihui, et al. "Learning a unified classifier incrementally via rebalancing." *CVPR*. 2019.

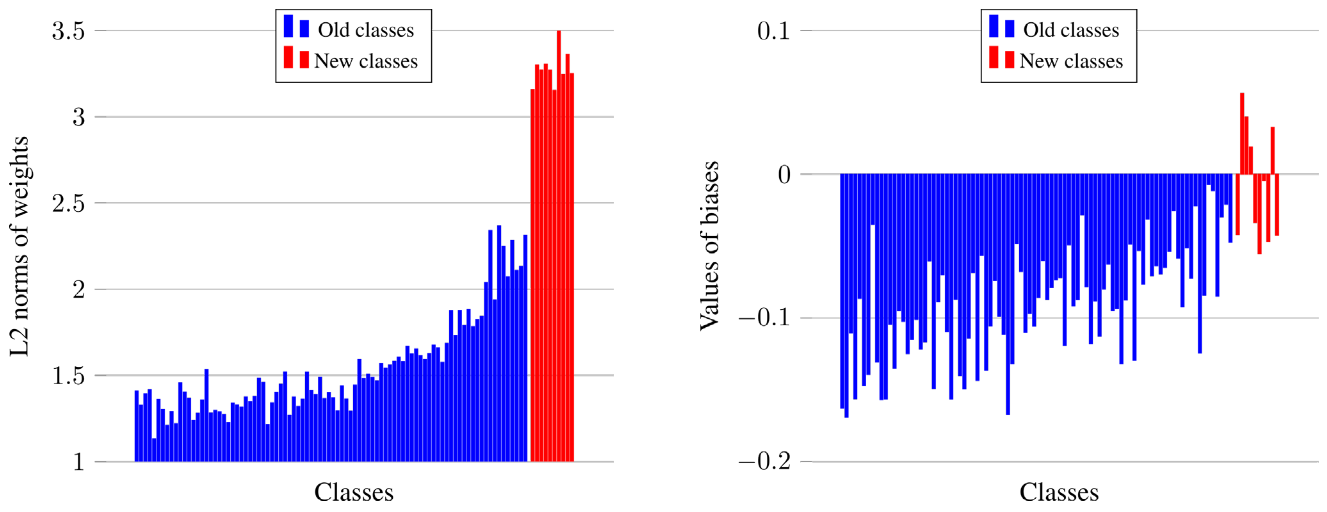
CLASSIFIER

K-nearest neighbors

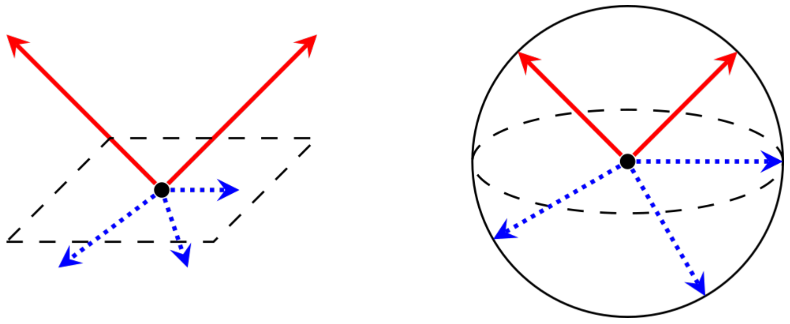
Cosine similarity

Random forest

Magnitude imbalance



Cosine layer



Hou, Saihui, et al. "Learning a unified classifier incrementally via rebalancing." *CVPR*. 2019.

CLASSIFIER

K-nearest neighbors

Cosine similarity

Random forest

Modified nearest-mean-of-exemplars
with cosine similarity

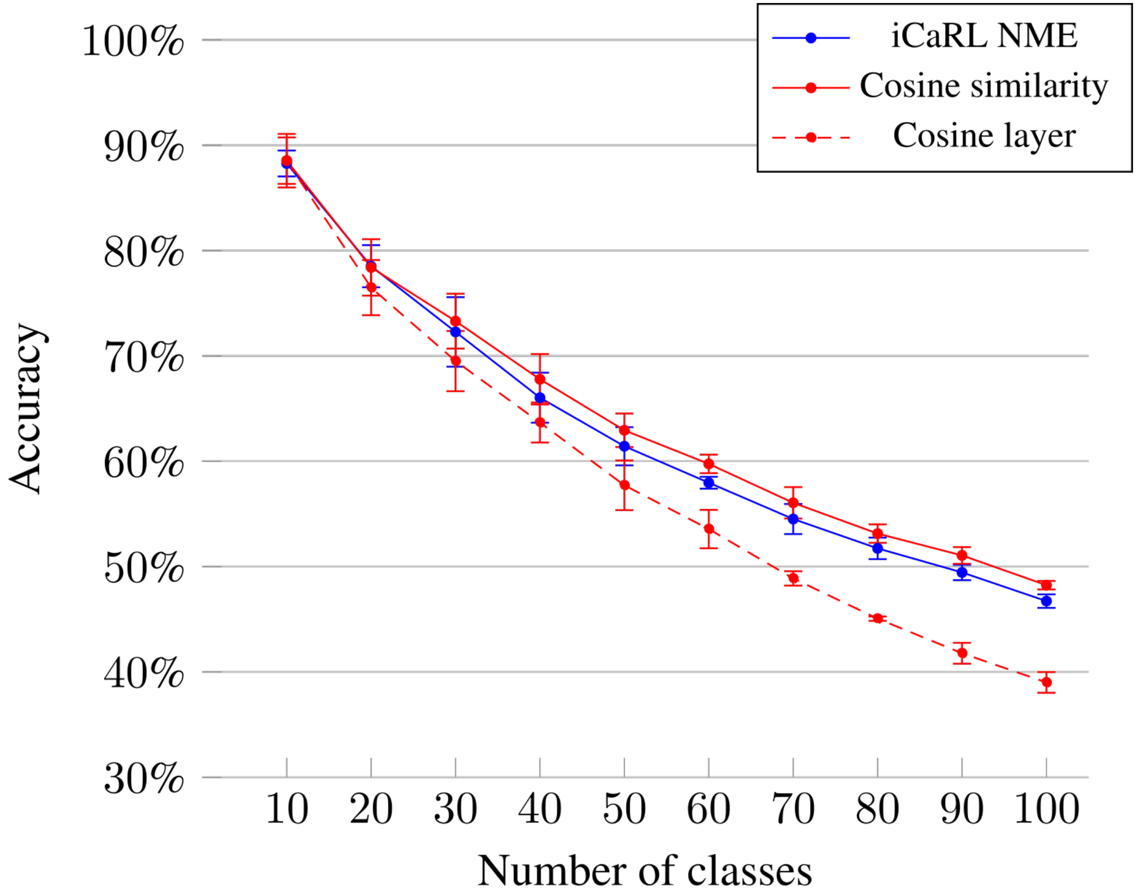
$$y^* \leftarrow \underset{y=1,\dots,t}{\operatorname{argmax}} \ \langle \bar{\varphi}(x), \bar{\mu}_y \rangle$$

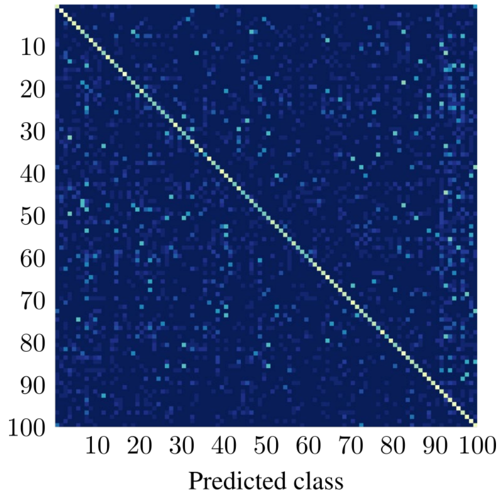
CLASSIFIER

K-nearest neighbors

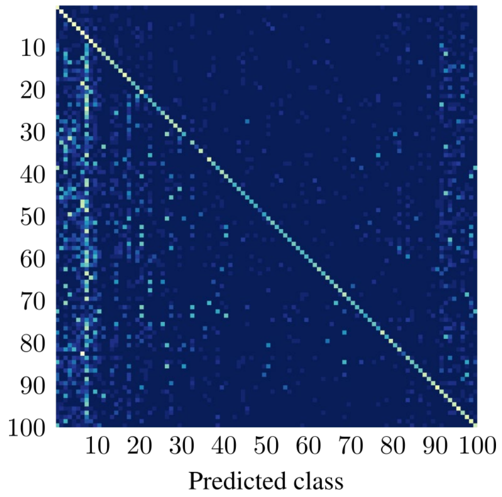
Cosine similarity

Random forest





Cosine similarity



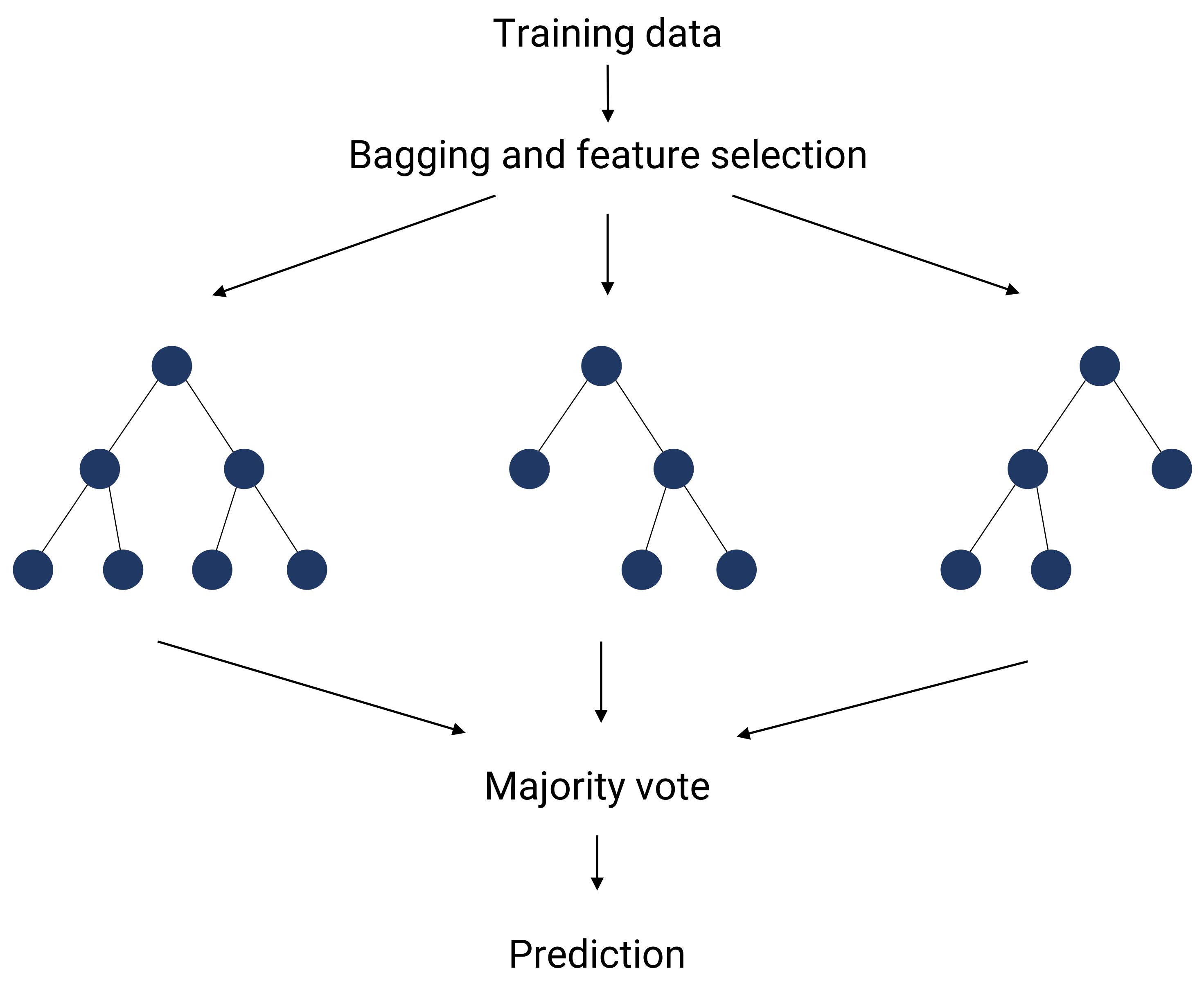
Cosine layer

CLASSIFIER

K-nearest neighbors

Cosine similarity

Random forest

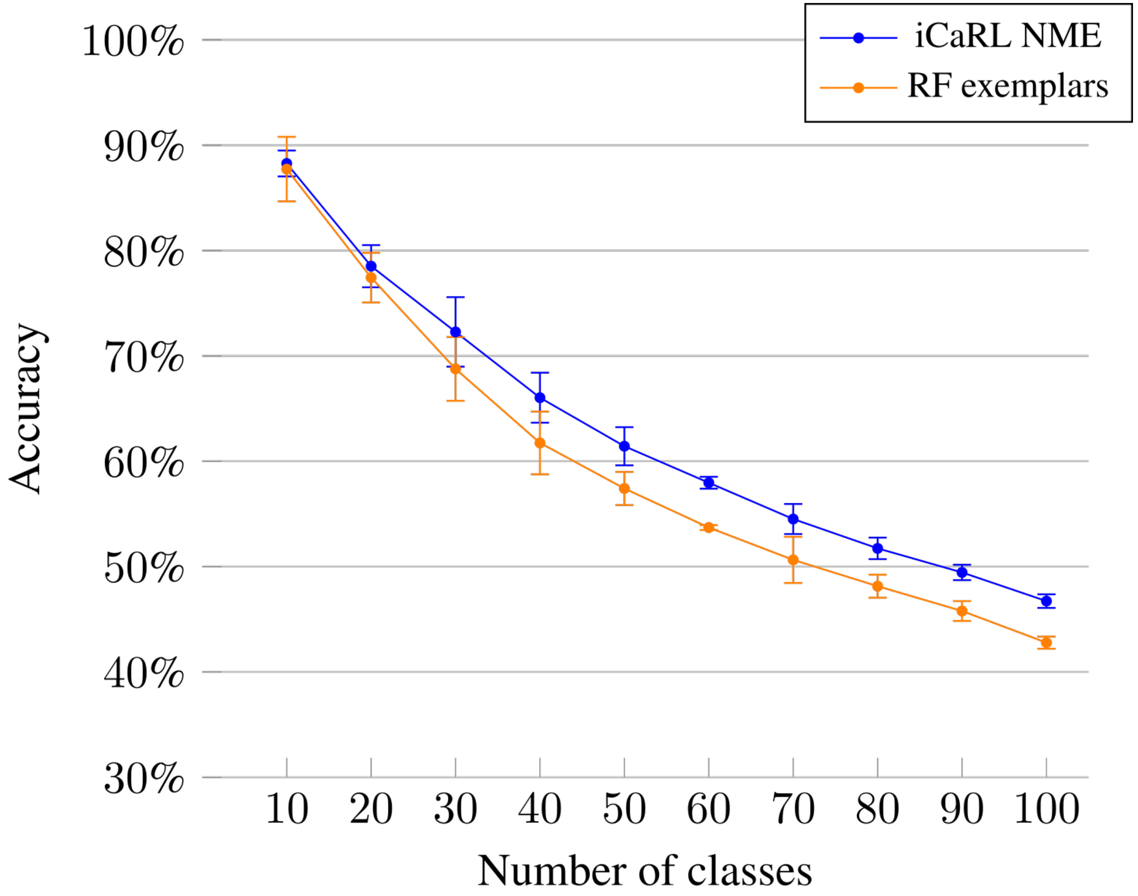


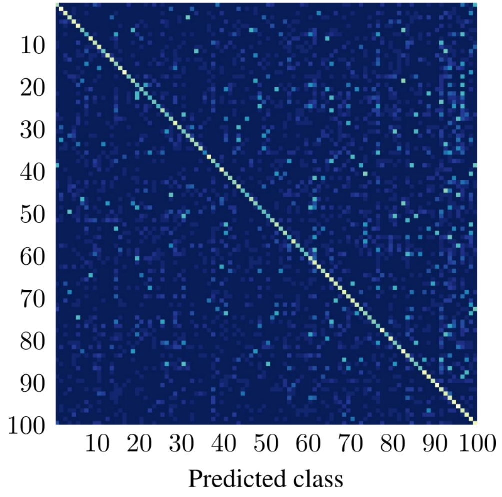
CLASSIFIER

K-nearest neighbors

Cosine similarity

Random forest





RF exemplars

BEYOND THE BASELINES

- Explore more deeply existing limitations
  - Propose variations to mitigate them

1

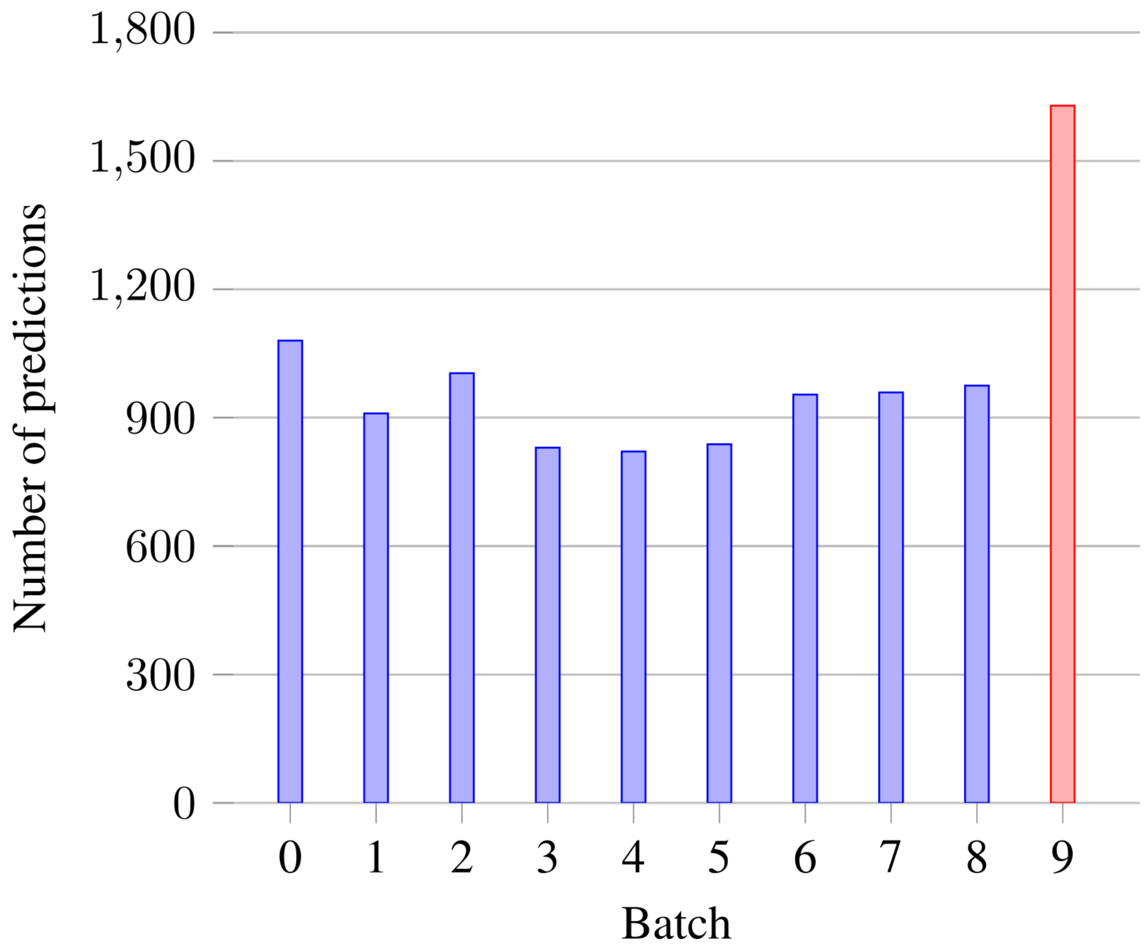
FEATURE REPRESENTATION DRIFT ANALYSIS

PREDICTION BIAS

Training is done over an unbalanced class distribution

⇓

Probability scores are biased towards new classes



OUR HYPOTHESIS

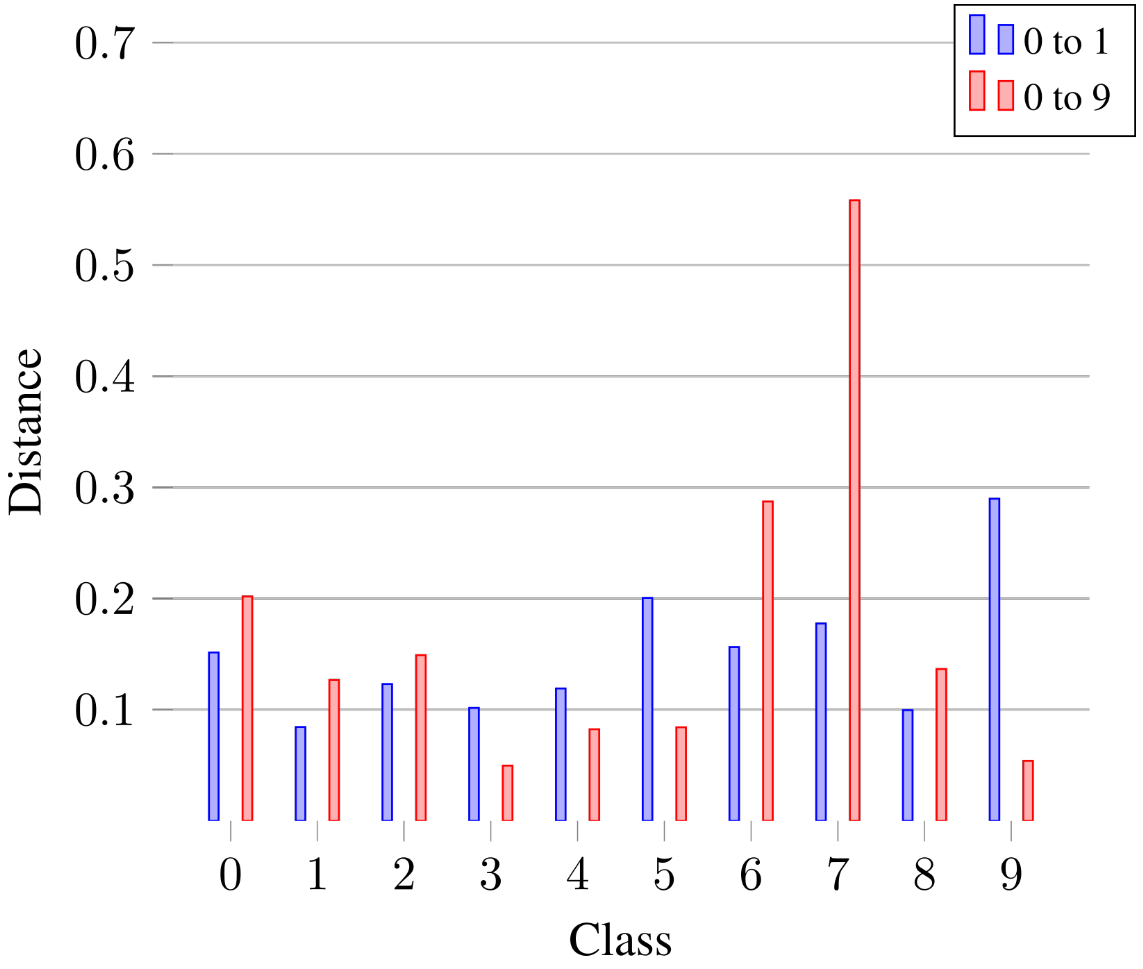
- Model learns a feature representation that best represents *new* classes

- Distillation contribution does not fully prevent drift of features in consecutive learning steps

FEATURE REPRESENTATION DRIFT

Comparison of class prototypes

- Feature vector is L2 normalized element-wise
- Distance is measured as weighted MSE



OUR PROPOSAL

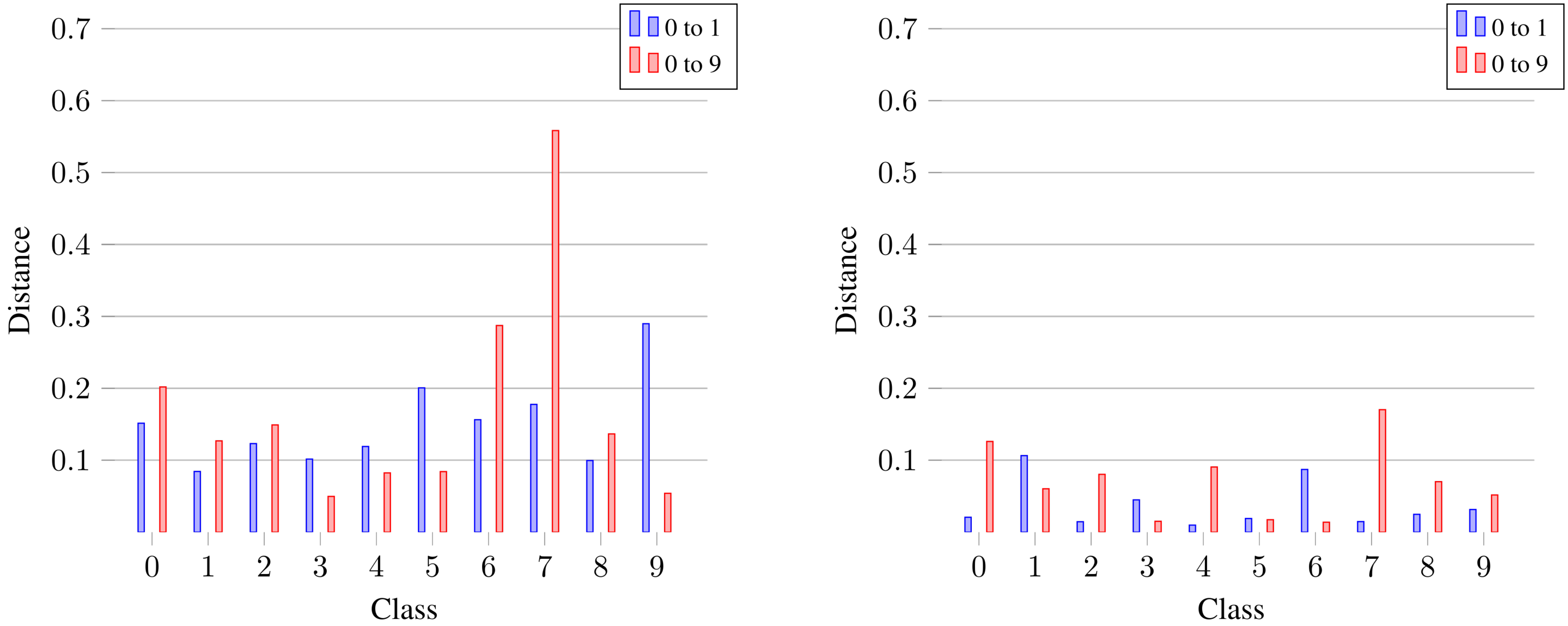
Mitigate drift by means of a loss contribution
to minimize distance between features of the sample
and prototype of the corresponding class

Weighted smooth L1 loss

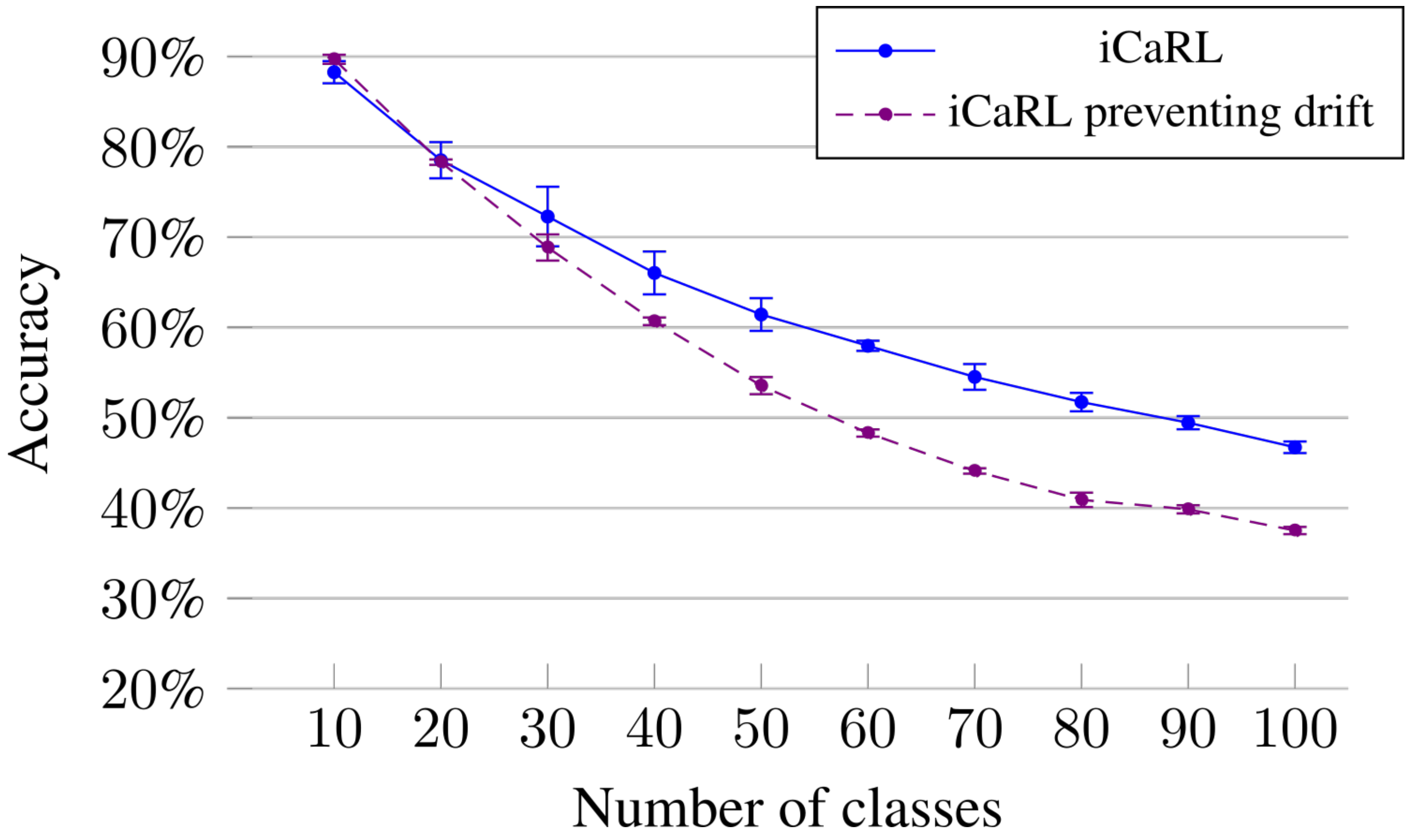
$$\mathcal{L}_{\text{drift}} = \alpha \frac{1}{n} \sum_{i=1}^{n} w_i z_i$$

$$z_i = \begin{cases} 0.5\,(x_i - y_i^L)^2 & \text{if } |x_i - y_i^L| < 1 \\ |x_i - y_i^L| - 0.5 & \text{otherwise} \end{cases}$$

DRIFT COMPARISON



PERFORMANCE COMPARISON

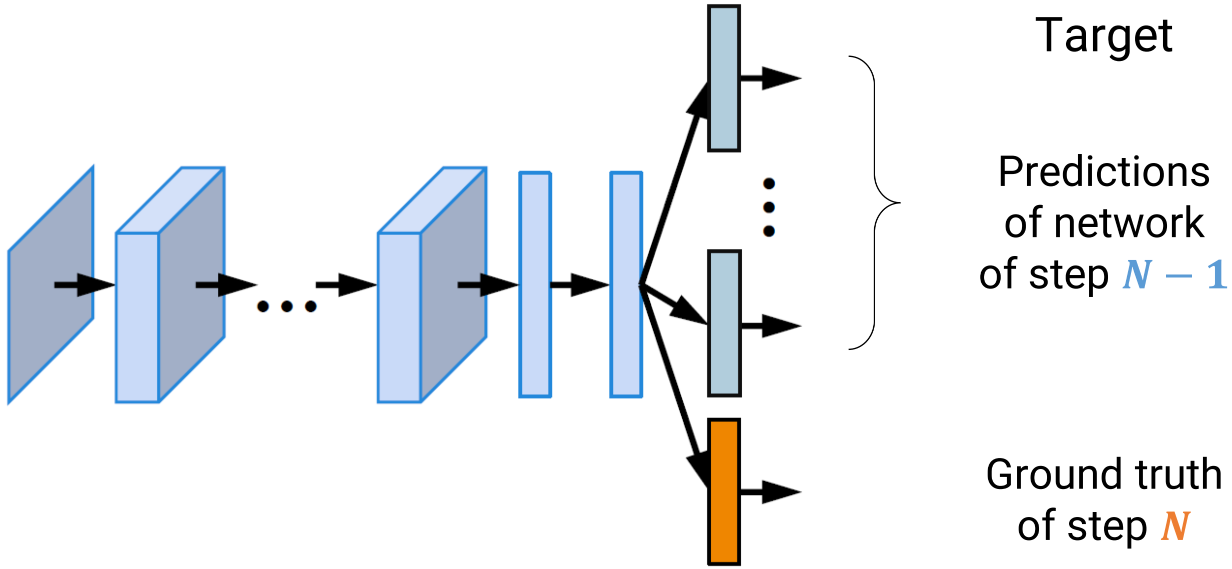


2

DISTILLATION TARGETS ANALYSIS

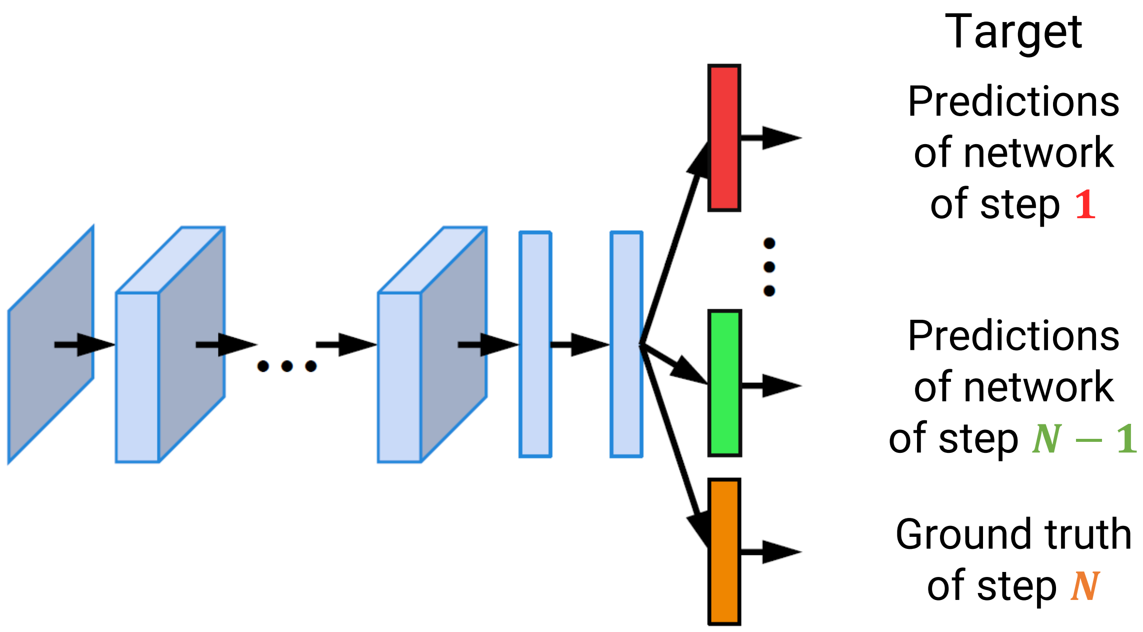
LAST NET POLICY

- Save last trained network
- At learning step N, use predictions from network trained at step $N-1$ as targets for distillation



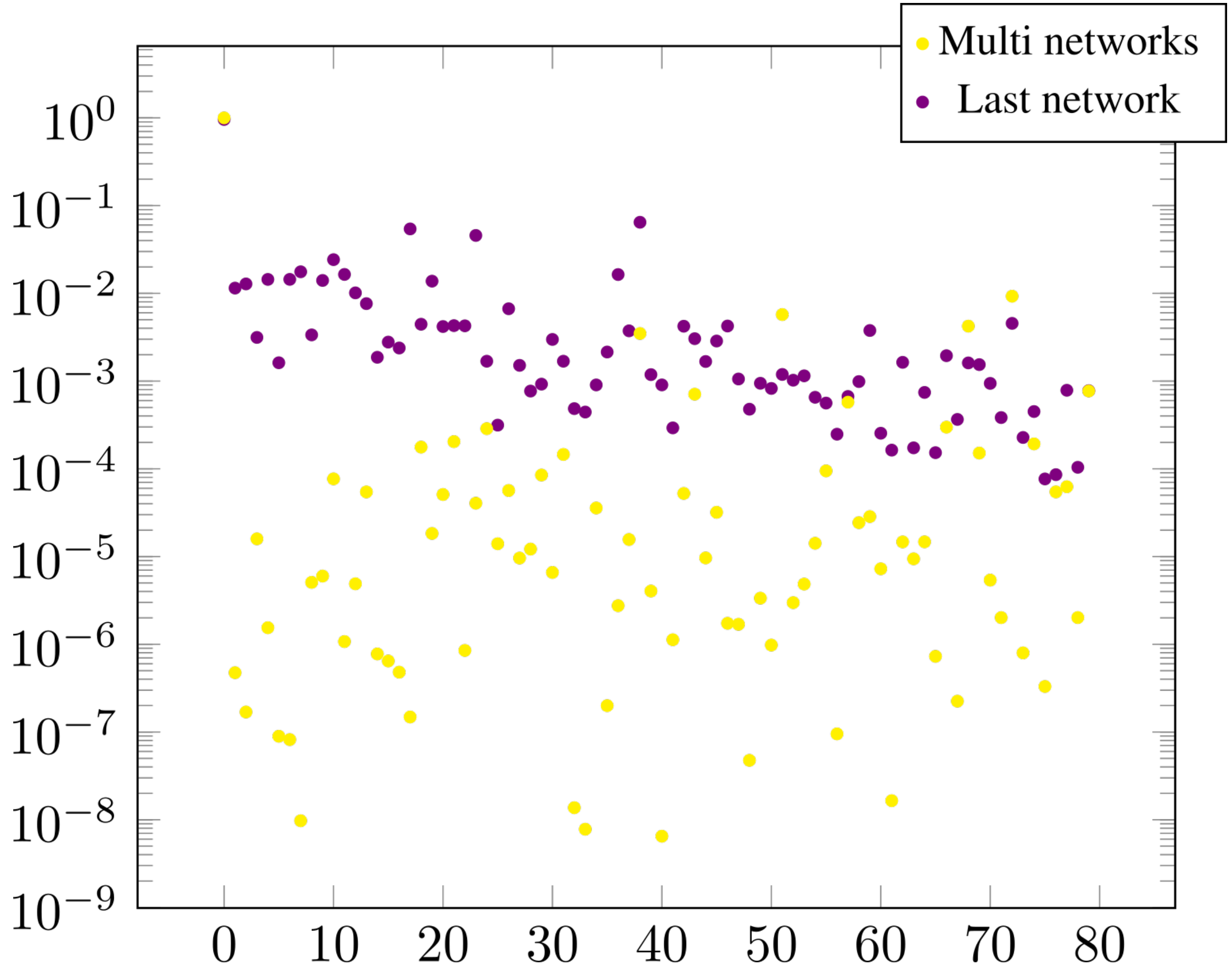
MULTI NET POLICY

- Save networks trained at different learning steps
- Use predictions from network trained at step $M \leq N-1$ as distillation targets for nodes associated with classes of batch M



TARGET COMPARISON: MULTI NET VS LAST NET

- Select 10 images of class 0 from the exemplars
  - $N - 1 = 9$
- Y axis on logarithmic scale

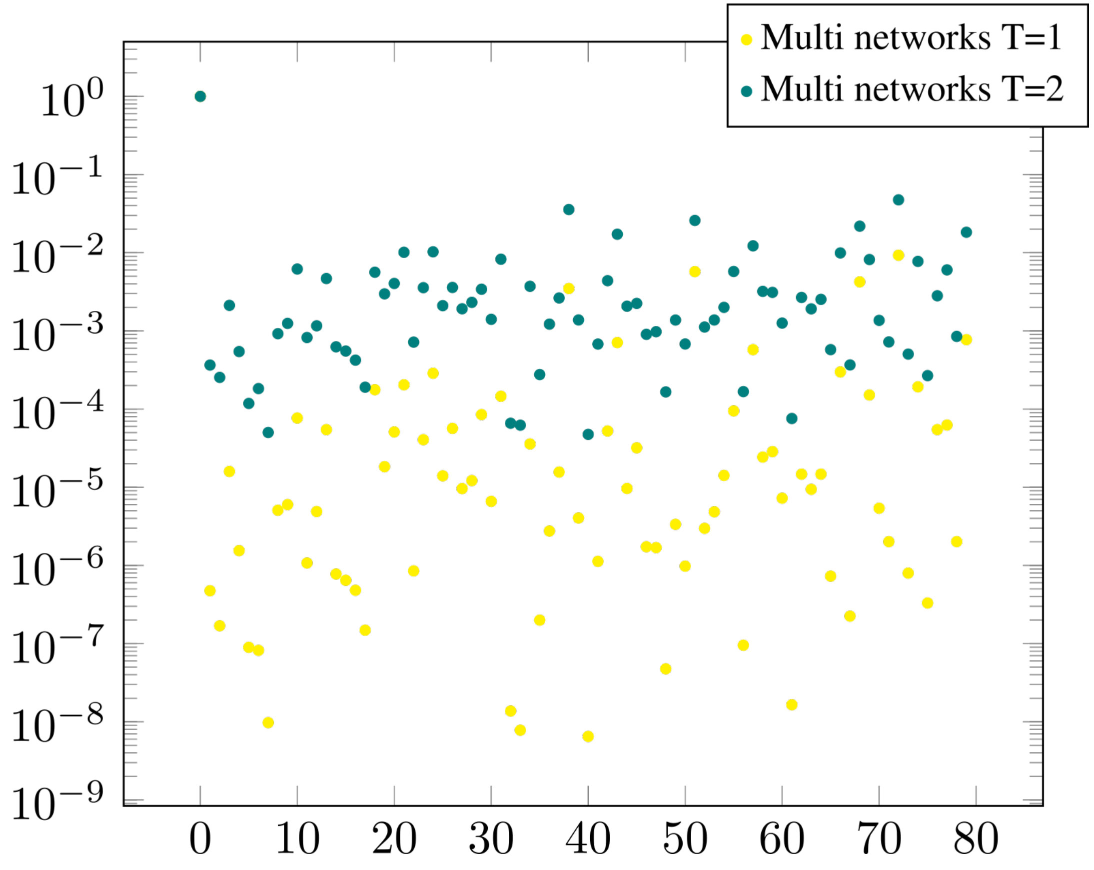


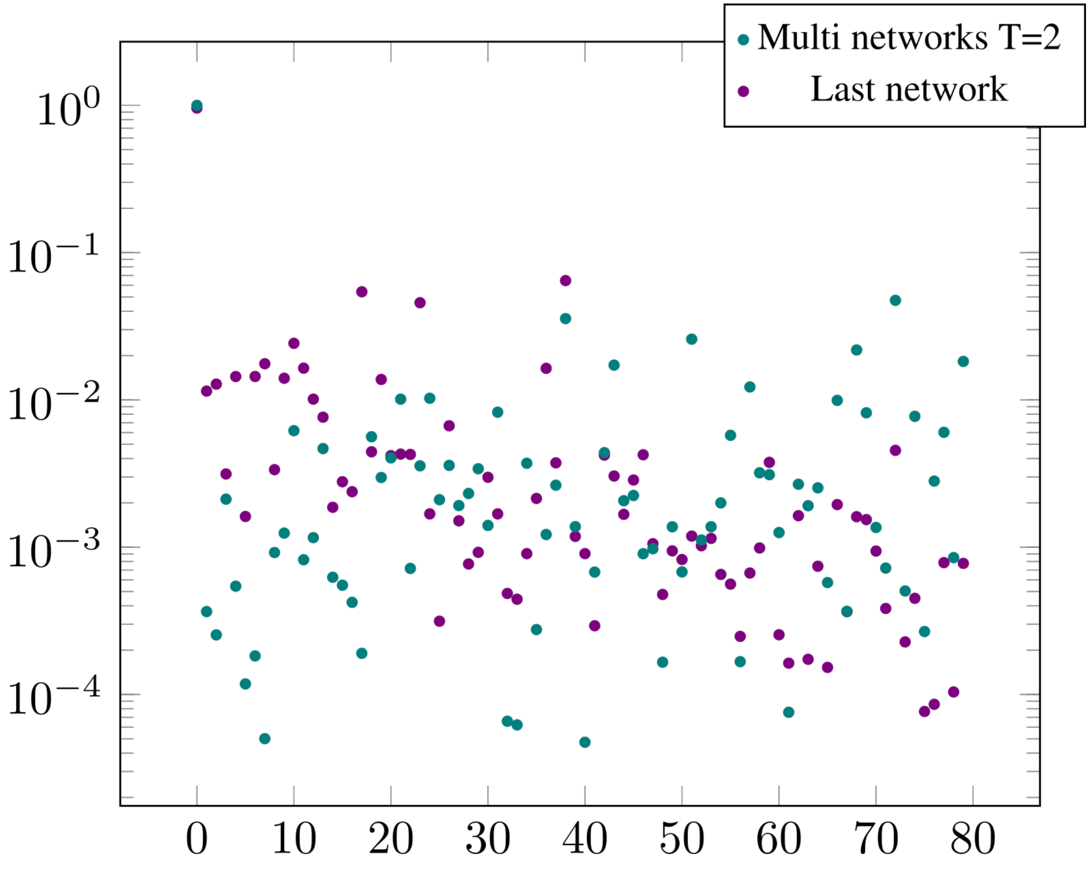
OUR PROPOSAL

- Train a model with **multi net policy**
- At each learning step, distillation targets are computed and stored *una tantum* at first epoch
- Soft targets

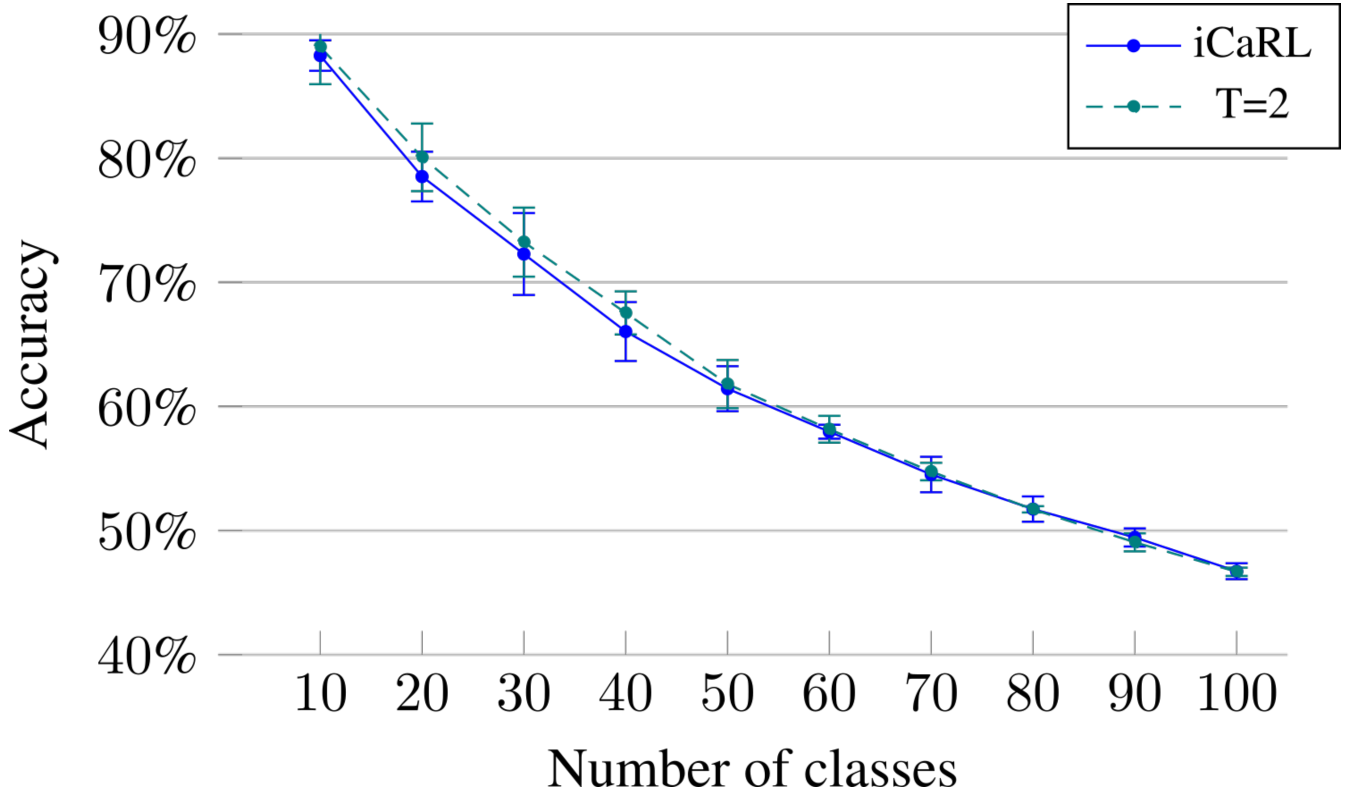
$$\sigma(x) = \frac{1}{1 + \exp(-x/T)}$$

SOFT TARGETS





PERFORMANCE COMPARISON



Method	Avg.
iCaRL	62.7%
Multi net $T = 1$	61.1%
Multi net $T = 2$	63.2%

THANK YOU FOR YOUR ATTENTION!