Workshop on Continual Learning
ICML 2020

On Class Orderings for Incremental Learning

Marc Masana*, Bartłomiej Twardowski*, Joost van de Weijer

Universitat Autònoma de Barcelona

Computer Vision Center

*Learning and Machine Perception (**LAMP**)*

Motivation

Does class ordering for task splitting influence overall class-IL performance?

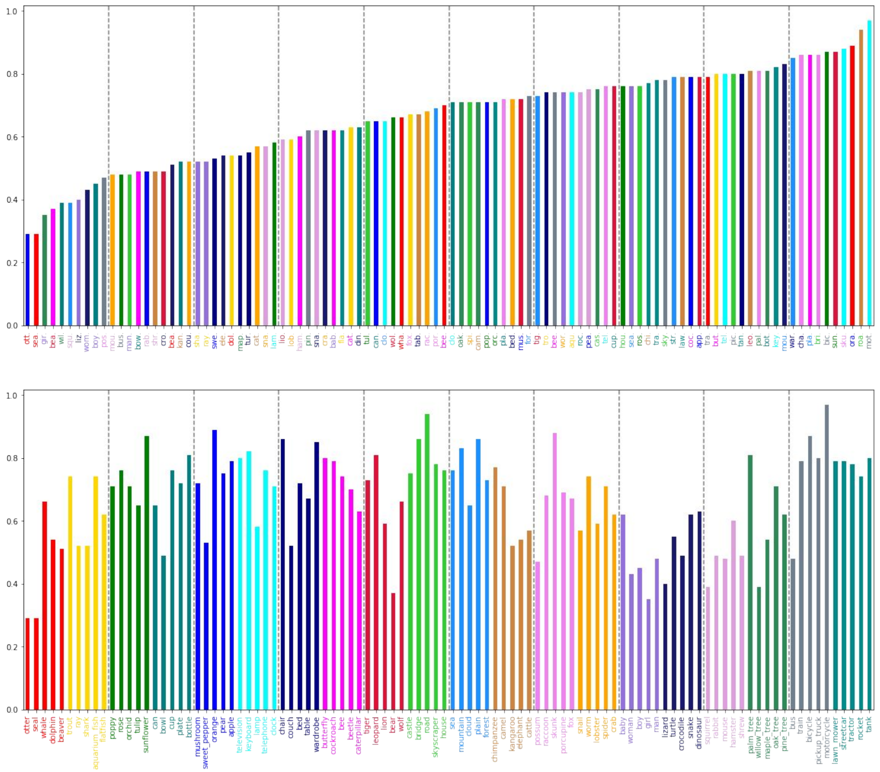
→ Not which is the best order to learn all tasks.

Investigate how relevant is class ordering and how it can affect the results of well-known class-IL methods.

→ Measure robustness.

More insight for class-IL settings.

→ *Tool* for better comparison.



Class ordering can influence task accuracy within a particular split for an already trained model on CIFAR-100.

Class ordering - Baselines

Random can be over-optimistic vs real-life.

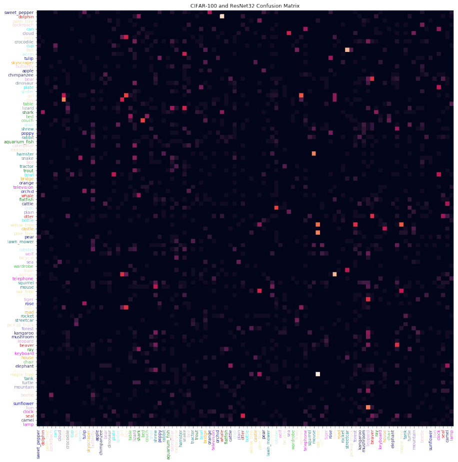
→ we might expect classes inside tasks to be somehow related.

Exploring and preserving semantics between tasks is hard.

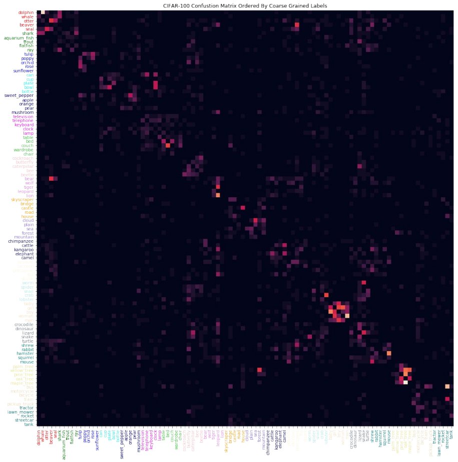
→ intra-task difficulty.

In an IL setting, discriminative features become more difficult to learn or can be modified afterwards, especially when the classes belong to different tasks.

random



coarse grained



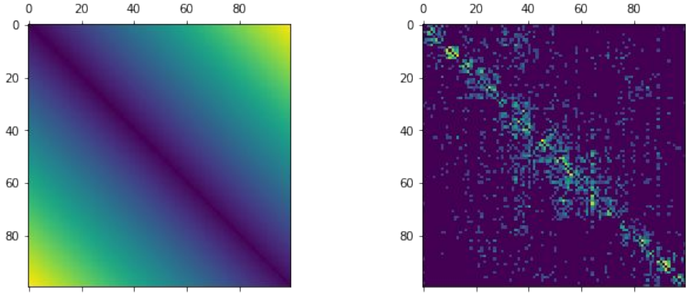
Two baseline class orderings provided with CIFAR-100.

Class ordering - w/o task boundaries

Ordering based on confusion matrix from jointly trained model (oracle) on all data.

<u>max confusion</u> (maxConf)

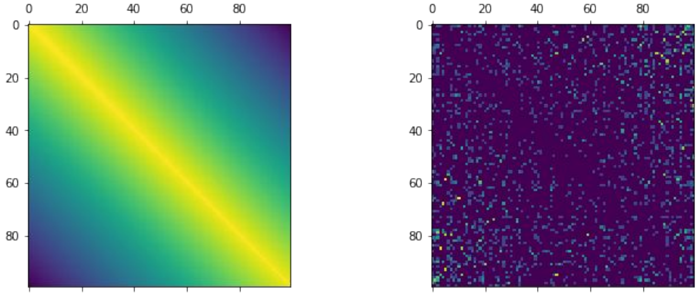
highly miss-classified classes are next to each other. This creates an IL split with more difficult intra-task classification.



max confusion around the diagonal.

<u>min confusion</u> (minConf)

rarely miss-classified classes are next to each other. Intra-task and adjacency tasks classification becomes easier, but also pushes the most miss-classified classes towards the first and last tasks.



max confusion at the corners.

Class ordering - w/ task boundaries

<u>increasing task confusion</u> (incTaskConf)

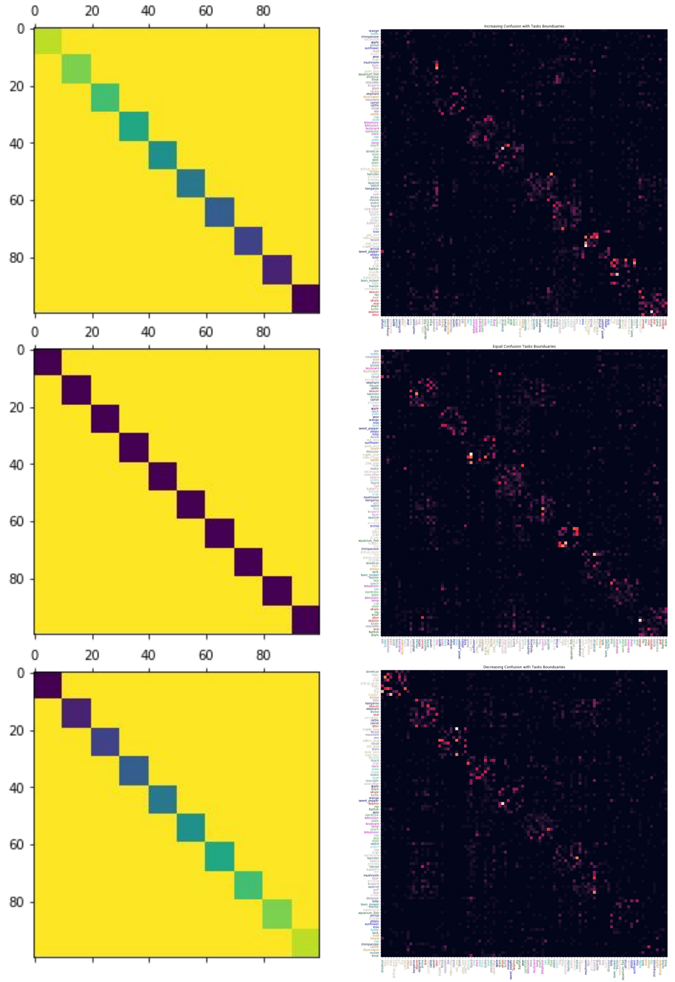
maximize confusion in all tasks around the diagonal of CM with increasing confusion between them.

<u>equal task confusion</u> (eqTaskConf)

similar to max confusion, but introducing task boundaries should cause less confusion between adjacent tasks.

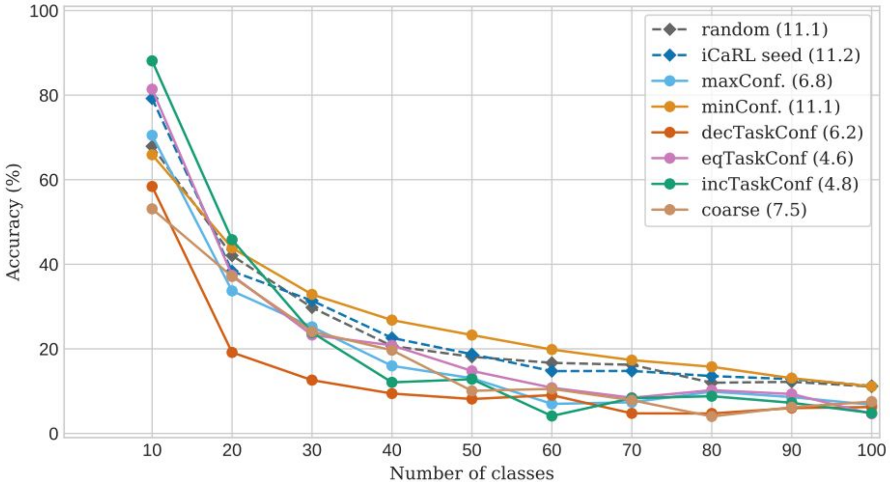
<u>decreasing task confusion</u>(decTaskConf)

maximize confusion in all tasks around the diagonal while decreasing it between them.

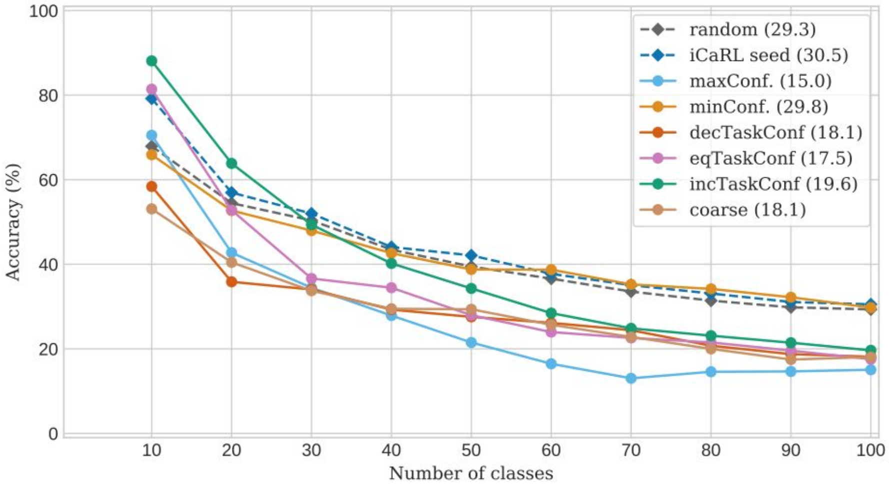


Experimental Results - No Exemplars

Finetuning



LwF[1]



CIFAR-100 (10 tasks with 10 classes each) on ResNet-32 from scratch.

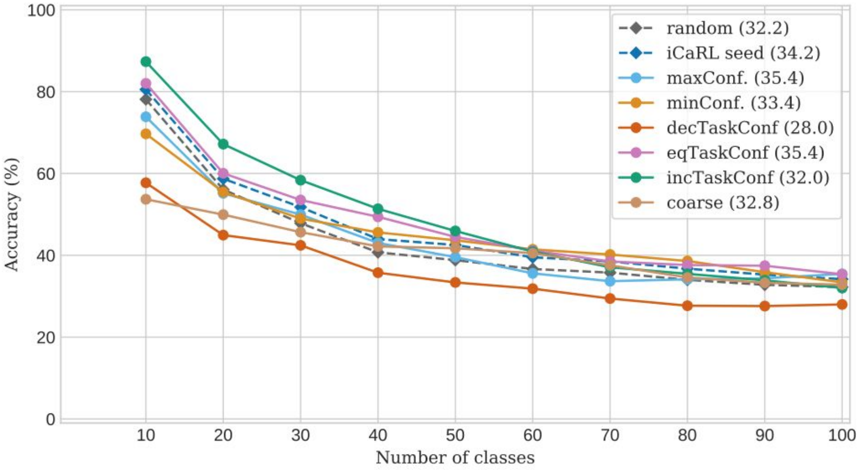
(momentum=0.9, weight decay=2e-4, starting LR=0.1, patience=10, LR factor=⅓, stop condition LR 1e-4 or 200 epochs).

Best trade-off gridsearch done per task with validation split and fixed after 3 first tasks.

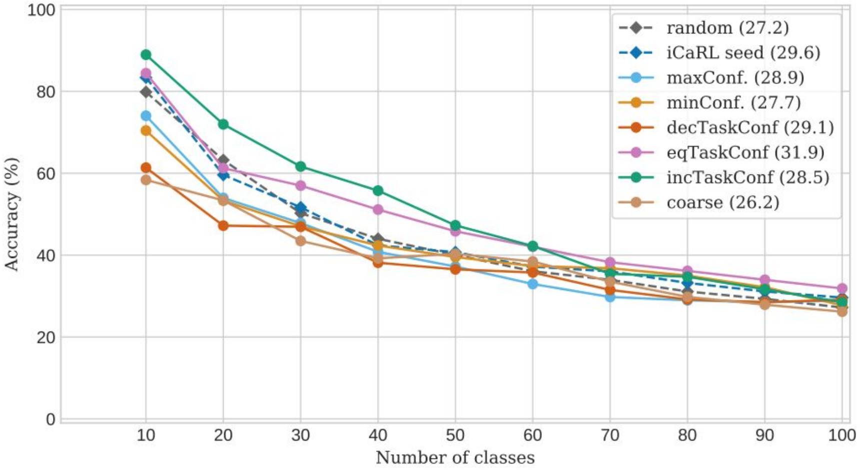
[1] Li, Zhizhong, and Derek Hoiem. "*Learning without forgetting.*" *PAMI,* 2017.

Experimental Results - With Exemplars (20 per class)

iCaRL[1]



LUCIR[2]



CIFAR-100 (10 tasks with 10 classes each) on ResNet-32 from scratch.

(momentum=0.9, weight decay=2e-4, starting LR=0.1, patience=10, LR factor=⅓, stop condition LR 1e-4 or 200 epochs).

Best trade-off gridsearch done per task with validation split and fixed after 3 first tasks.

[1] Rebuffi, Sylvestre-Alvise, et al. "*iCaRL: Incremental classifier and representation learning.*" CVPR, 2017.
[2] Hou, Saihui, et al. "*Learning a unified classifier incrementally via rebalancing.*" CVPR, 2019.

Conclusions

- **Random** obtains among the **highest performances** when used by **non-exemplar** methods.
- Methods with **exemplars decrease difference** between distinct class orderings.
- In Class-IL, **ranking** of the methods **can change due to different orderings.**
- Commonly used **iCaRL seed** seems to be **as good as random.**
- The proposed class orderings based on the **confusion matrix** can be used as a **tool** for checking **robustness** of class-IL approaches.

THANKS!

Experimental Results - With Exemplars (full results)

Table 1. CIFAR-100 results for class-IL with growing memory of 20 exemplars per class (10 runs average and standard deviation). The best score for each task of each method is in bold. Underscore marks the lowest score.

	task	Random	iCaRL seed	coarse	maxConf	minConf	decTaskConf	eqTaskConf	incTaskConf
LwF	2	51.8±5.3	50.8±2.6	45.9±2.7	52.4±2.0	44.6±1.9	41.5±3.2	53.1±3.9	**62.5**±2.7
	5	31.8±3.9	33.2±3.3	36.3±2.2	33.0±2.9	33.8±2.3	33.9±2.6	35.9±3.2	**37.4**±2.9
	10	24.8±2.6	27.3±2.2	26.5±2.0	**32.6**±3.1	25.4±1.4	29.9±2.4	31.6±3.3	29.4±1.9
iCaRL	2	61.4±3.8	58.7±3.6	49.9±3.1	55.1±1.6	55.5±1.3	44.9±2.4	60.0±2.7	**67.2**±3.9
	5	42.0±3.6	42.5±2.7	41.7±2.2	39.6±2.3	43.7±2.4	33.4±2.6	44.5±2.4	**45.9**±3.6
	10	33.8±3.7	34.2±2.6	32.8±1.7	**35.4**±2.6	33.4±2.3	28.0±2.2	35.4±3.0	32.0±3.1
BiC	2	61.2±5.3	56.4±3.8	50.4±3.0	52.8±2.6	51.5±2.7	41.2±2.1	57.9±4.1	**69.7**±2.7
	5	44.8±3.2	45.2±3.0	44.4±2.6	40.7±3.7	47.7±1.3	37.1±2.9	45.3±3.1	**52.4**±2.8
	10	39.3±1.9	40.1±2.8	39.3±2.3	37.5±2.7	**40.1**±1.3	37.2±2.9	38.6±2.0	37.8±2.6
LUCIR	2	63.2±3.4	59.6±3.3	53.4±2.9	54.0±3.2	53.3±3.3	47.2±1.9	61.2±2.0	**72.0**±2.2
	5	40.0±3.4	40.7±3.5	40.3±2.6	37.2±5.6	39.5±2.8	36.5±2.9	45.8±1.4	**47.3**±1.9
	10	27.2±2.7	29.6±3.2	26.2±3.1	28.9±5.8	27.7±1.9	29.1±3.3	**31.9**±2.0	28.5±1.6
IL2M	2	58.2±5.1	52.2±3.7	43.2±5.9	51.2±1.1	51.1±3.0	42.1±2.7	54.4±2.7	**65.3**±2.6
	5	44.2±4.1	41.0±3.7	44.0±2.4	40.6±2.6	47.5±1.7	37.1±3.0	43.6±2.6	**51.0**±2.4
	10	38.2±2.0	37.9±2.0	**38.6**±2.1	38.5±2.7	36.8±1.7	37.4±2.4	38.3±2.3	37.6±2.3