Appendix

A. Models Hyperparameter Details

We use SGD as optimizer, as it performs better in the continual learning setups [34], with a momentum value of 0.9. For the IIRC-CIFAR experiments, The learning rate starts with 1.0 and is decayed by a factor of 10 on plateau of the performance of the peri-task validation subset that corresponds to the current task. For the IIRC-ImageNet experiments, the learning rate starts with 0.5 in the case of iCaRL-CNN, iCaRL-norm and BiC, and 0.1 in the case of finetune, ER and LUCIR, and is decayed by a factor of 10 on plateau. The number of training epochs per task is 140 for IIRC-CIFAR and 100 for IIRC-ImageNet, with the first task always trained for double the number of epochs (due to its larger size). We set the batch size to 128 and the weight decay parameter to $1e-5$. Moreover, We set the A-GEM memory batch size, which is used to calculate the reference gradient, to 128. For LUCIR, the margin threshold m is set to 0.5, and λ_{base} is set to 5. All the hyperparameters were tuned based on the validation performance in experiments that include only the first four tasks.

During training, data augmentations are applied as follows: for IIRC-ImageNet, a random crop of size (224×224) is sampled from an image, a random horizontal flip is applied, then the pixels are normalized by a pre-calculated per-channel mean and standard deviation. In IIRC-CIFAR, a padding of size 4 is added to each size, then a random crop of size (32×32) is sampled, a random horizontal flip is applied, then the pixels are normalized.

We keep a fixed number of samples per class in the replay buffer (20, except otherwise indicated). Hence, the capacity increases linearly as the model learns more classes. These samples are chosen randomly, except for iCaRL, LUCIR and BiC which use the herding approach. IIRC-CIFAR experiments are averaged over ten task configurations and the each version of IIRC-ImageNet is averaged over 5 task configurations (see 4 for details).

B. Baselines Details

Following are three well known baselines that we used to evaluate in the IIRC setup along other baselines including finetune, joint and incremental joint, Vanilla Experience Re-play (ER), and Experience Replay with infinite buffer (ER-infinite).

iCaRL: iCaRL [38] was among the first deep learning methods to use exemplar rehearsal in order to alleviate the catastrophic forgetting in the class incremental learning setup. iCaRL model updates the model parameters using the distillation loss, where the outputs of the previous network are used as soft labels for the current network. Moreover, it uses the nearest-mean-of-exemplars classifications (NMC) strategy to classify test samples during inference. Since it is difficult to use NMC when the number of labels is variable (not a single label setup), we use the classification layer used during training during inference as well.

Unified learning via rebalancing (LUCIR): LUCIR [20] is a class incremental method that exploits three components to alleviate the catastrophic forgetting and reduce the negative effect of the imbalance between the old and new classes, since the number of samples in the replay buffer is much less than the current task samples. LUCIR uses the cosine normalization to get balanced magnitudes for classes seen so far. It also uses the less forget constraint, where the distillation loss is applied in the feature space instead of the output space, and a margin ranking loss to ensure interclass separation.

A-GEM: A-GEM [9] is an improved version of GEM that is a constrained optimization method in the Replay-based approach. GEM uses memory to constrain gradients so as to update the model parameters to not interfere with previous tasks. GEM is a very computationally expensive approach that is not applicable to the large-scale setup. Hence, Averaged GEM (A-GEM) provides an efficient version of GEM, where it only requires computing the gradient on a random subset of memory examples, and it does not need as well to solve any quadratic program but just an inner product. Since A-GEM is a very well known constrained optimization method that has reasonable guarantees in terms of average accuracy in comparison to GEM, we selected it as a candidate to evaluate its performance in our more realistic large scale setting.

C. IIRC Dataset Statistics

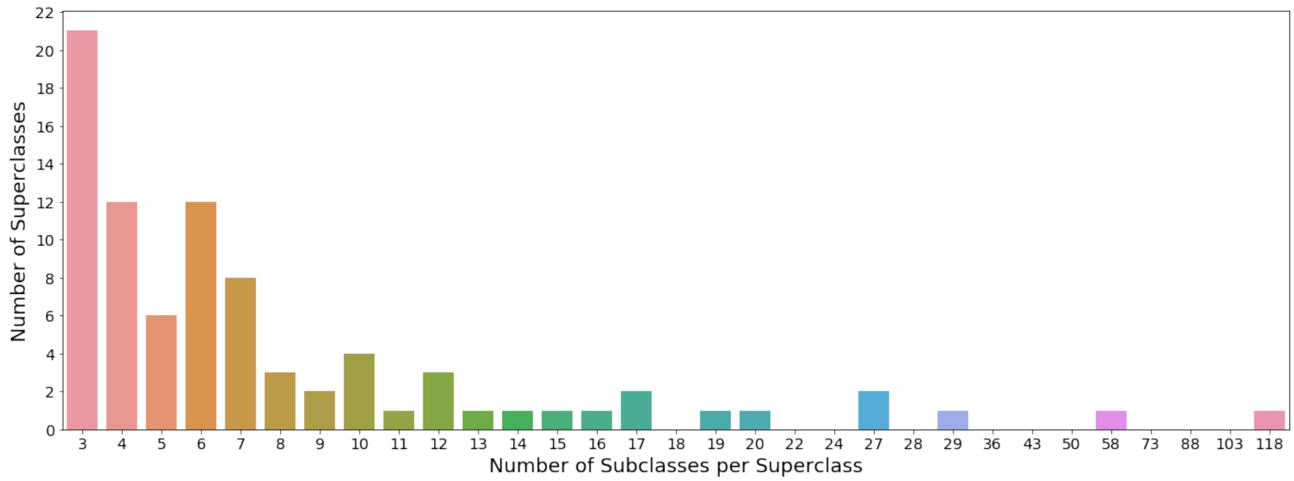


Figure A.7. The distribution of the number of subclasses per superclass on IIRC-ImageNet.

	with duplicates		without duplicates			
dataset	train	in-task validation	train	in-task validation	post-task validation	test
IIRC-CIFAR	46,160	5,770	40,000	5,000	5,000	10,000
IIRC-ImageNet-full	1,215,123	51,873	1,131,966	48,802	51,095	49,900

Table 1. The number of samples for each split of the training set. with duplicates represents the number of samples including the dublicates between some superclasses and their subclasses (the samples that the model see two times with two different labels). This doesn't happen for the post-task validation set and test set as they are in the complete information setup

dataset	superclasses	subclasses (under superclasses)	subclasses (with no superclasses)	total
IIRC-CIFAR	15	77	23	115
IIRC-ImageNet-full	85	788	210	1083

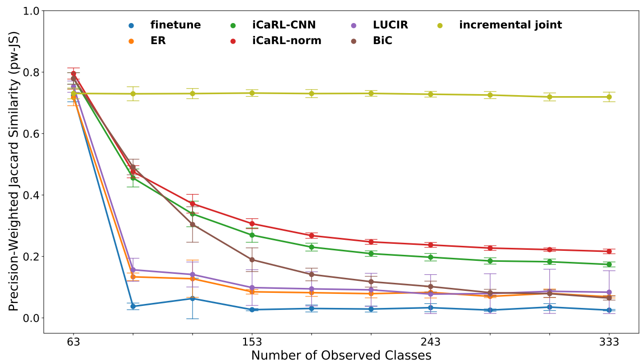
Table 2. For each dataset, these are the number of superclasses, the number of subclasses that belong to these superclasses, the number of subclasses that don't have a superclass, and the total number of superclasses and subclasses

dataset	superclass	num of subclasses	superclass size	subclass	subclass size
IIRC-CIFAR	vehicles	8	1,280	bus	320
IIRC-CIFAR	small mammals	5	800	squirrel	320
IIRC-CIFAR	-	-	-	mushroom	400
IIRC-ImageNet	bird	58	3,762	ostrich	956
IIRC-ImageNet	big cat	6	2,868	leopard	956
IIRC-ImageNet	keyboard instrument	4	1,912	grand piano	956
IIRC-ImageNet	-	-	-	wooden spoon	1,196

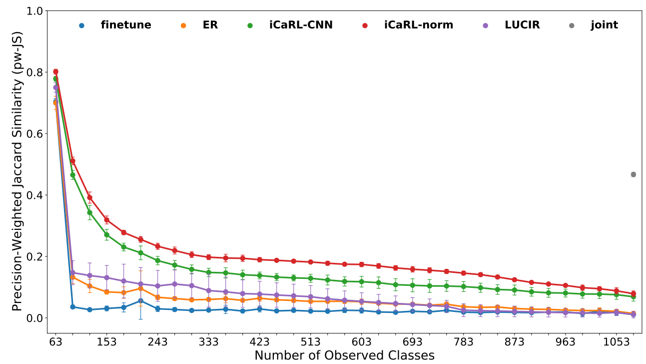
Table 3. Several examples for classes and the number of samples they have in the training set. The subclass on the right is a subclass that belongs to the superclass on the left. The left side is blank for subclasses that have no superclasses.

D. More Figures

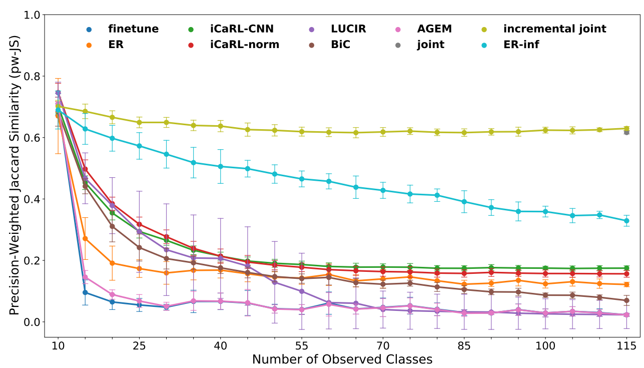
D.1. Main Paper Figures With The Standard Deviation Reported



(a) IIRC-ImageNet-lite

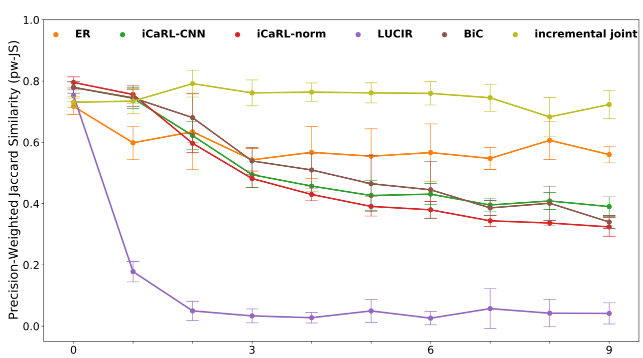


(b) IIRC-ImageNet-full

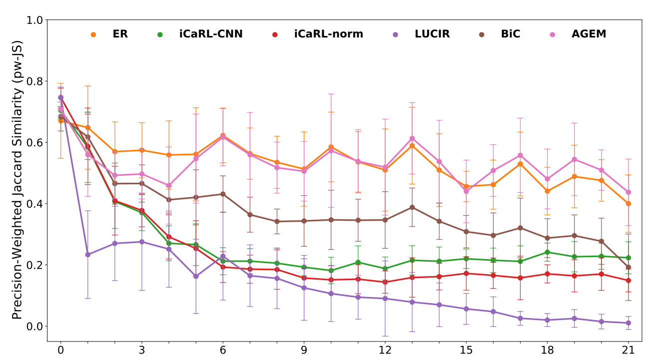


(c) IIRC-CIFAR

Figure A.8. Average performance (Figures 3 and 4) with the standard deviation reported. It was removed from the original figures for more intelligibility



(a) IIRC-ImageNet-full

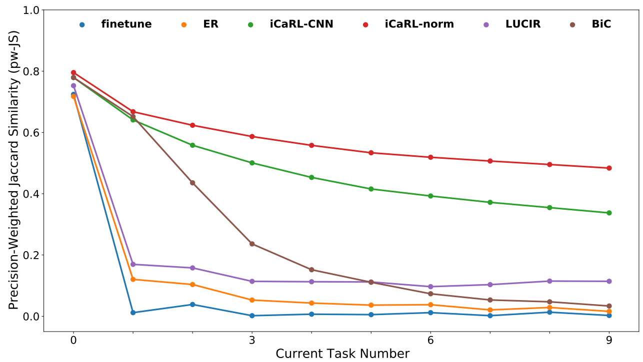


(b) IIRC-CIFAR

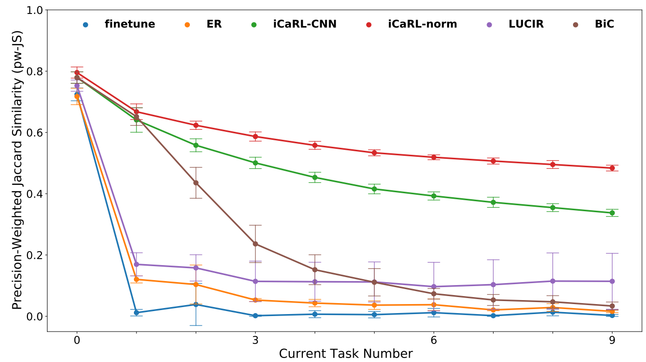
Figure A.9. Per task performance (Figure 5) with the standard deviation reported. It was removed from the original figures for more intelligibility

D.2. Average Performance Over the Superclasses Only

Figures A.10 and A.11 show how much each model is able to predict correctly the superclasses it learned earlier for the samples that come in later tasks (which mostly belong to their subclasses). This is measured by calculating the precision weighted Jaccard similarity solely over the superclasses (only the model predictions that belong to superclasses are taken into account).

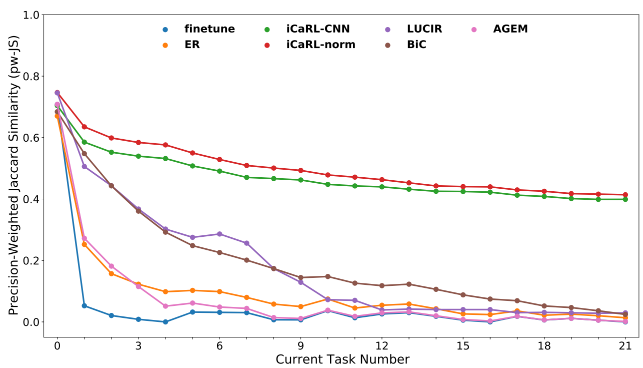


(a) without error bars

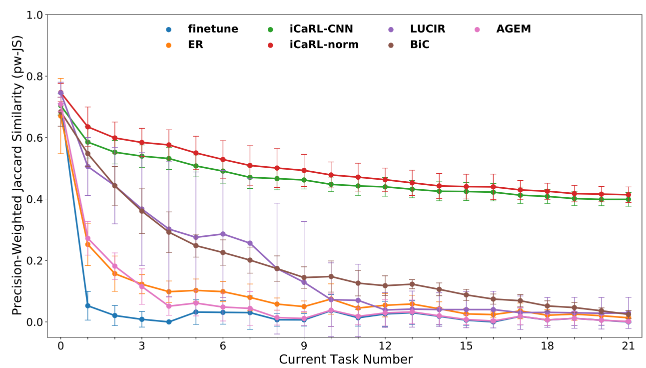


(b) with error bars

Figure A.10. The average performance of IIRC-ImageNet-lite if only the superclasses are taken into account for calculating this performance



(a) without error bars

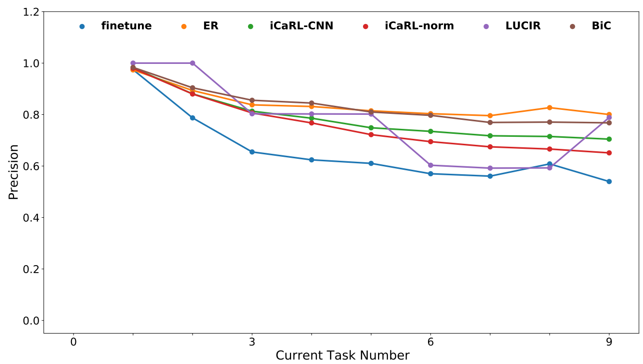


(b) with error bars

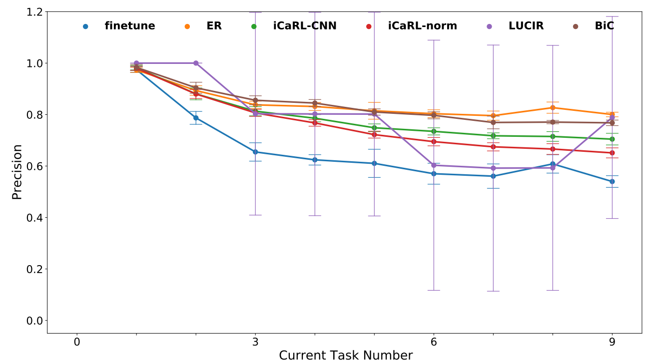
Figure A.11. The average performance of IIRC-CIFAR if only the superclasses are taken into account for calculating this performance

D.3. Average Precision Among Subclass Types

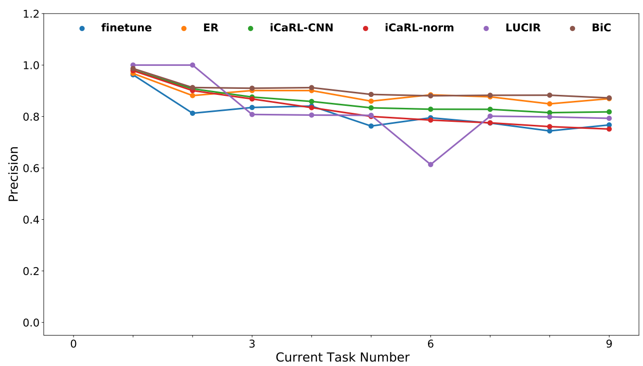
Figures A.12 and A.13 show how much each model is confused between the subclasses that belong to superclasses, which is measured by calculating the precision over only these subclasses (only the model predictions that belong to these subclasses are taken into account). As a baseline, the precision over subclasses which have no superclasses are also plotted. We can see that models collectively tend to be less precise with subclasses that have superclasses vs subclasses which have no superclasses, which is the intuitive result given that subclasses that belong to superclasses are more visually similar and easier to confuse, but also these figures help in highlighting which models are more prone to this kind of confusion between similar subclasses.



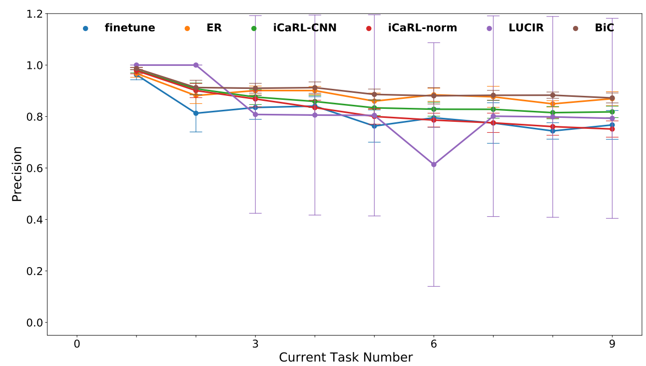
(a) Subclasses under superclasses (without error bars)



(b) Subclasses under superclasses (with error bars)

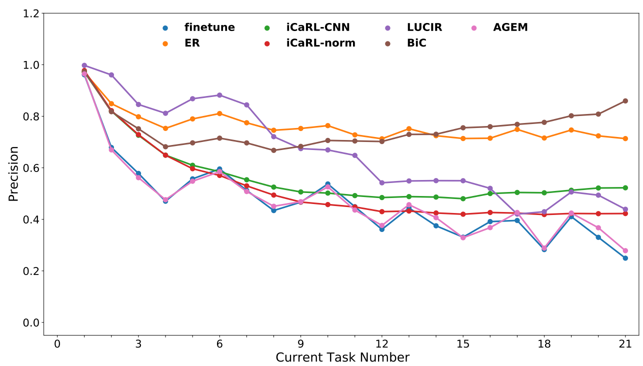


(c) Subclasses with no superclasses (without error bars)

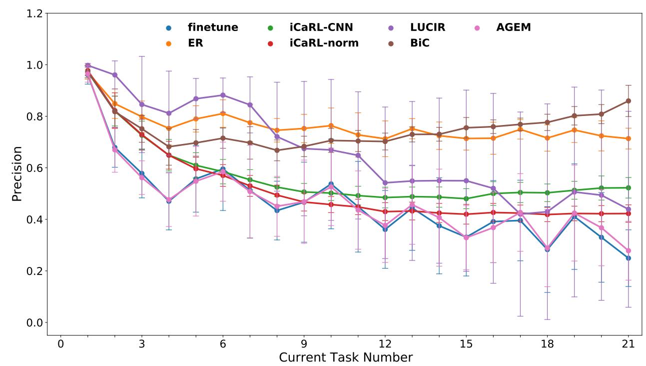


(d) Subclasses with no superclasses (with error bars)

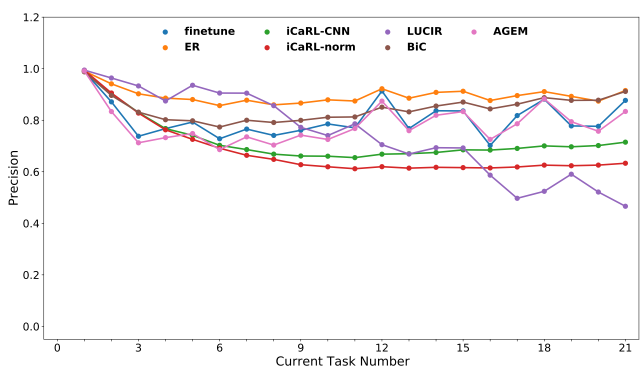
Figure A.12. The average precision of IIRC-ImageNet-lite over each type of subclasses, excluding other types of classes, to measure how much do the models confuse the subclasses as they encounter more related subclasses)



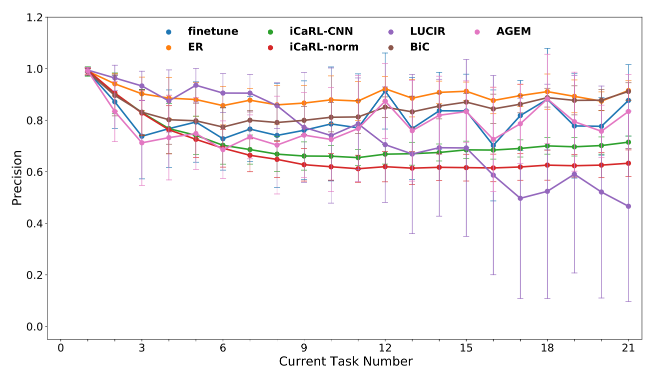
(a) Subclasses under superclasses (without error bars)



(b) Subclasses under superclasses (with error bars)



(c) Subclasses with no superclasses (without error bars)



(d) Subclasses with no superclasses (with error bars)

Figure A.13. The average precision of IIRC-CIFAR over each type of subclasses, excluding other types of classes, to measure how much do the models confuse the subclasses as they encounter more related subclasses)

D.4. pw-Jaccard Similarity vs Jaccard Similarity

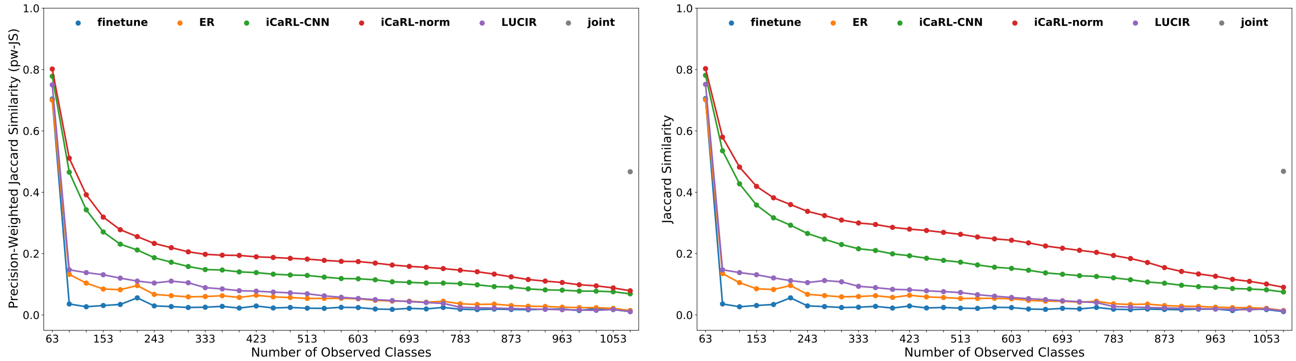


Figure A.14. Average performance on IIRC-CIFAR. (left) the precision-weighted Jaccard Similarity and (right) the Jaccard Similarity.

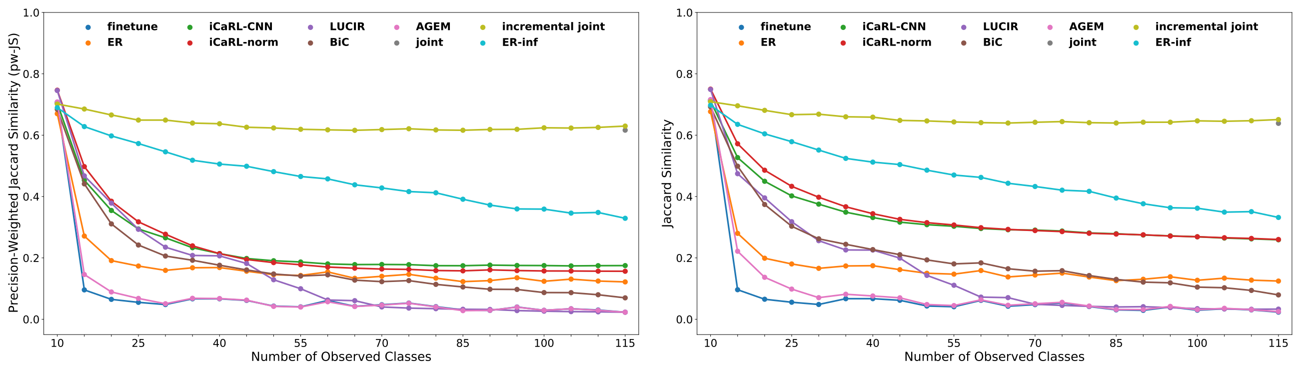
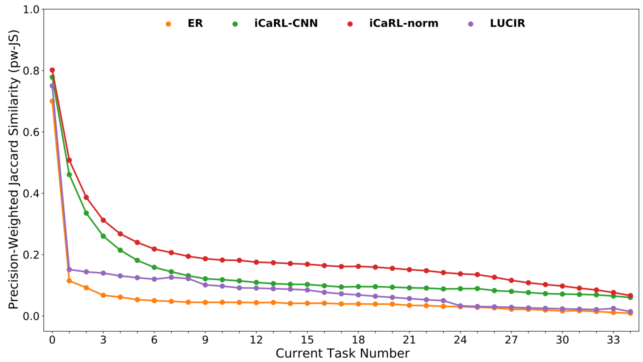
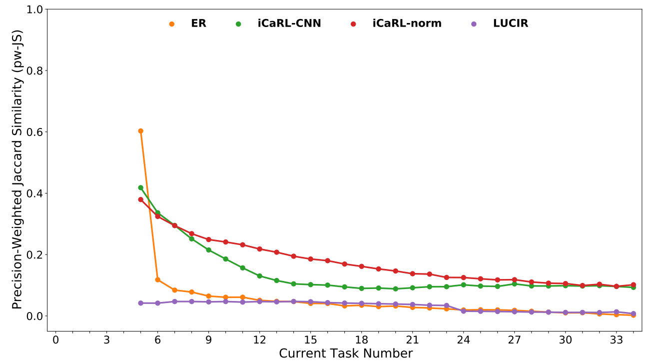


Figure A.15. Average performance on IIRC-CIFAR. (left) the precision-weighted Jaccard Similarity and (right) the Jaccard Similarity.

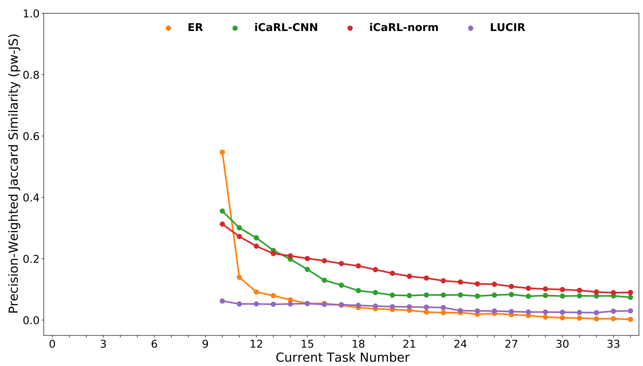
D.5. Performance Per Task



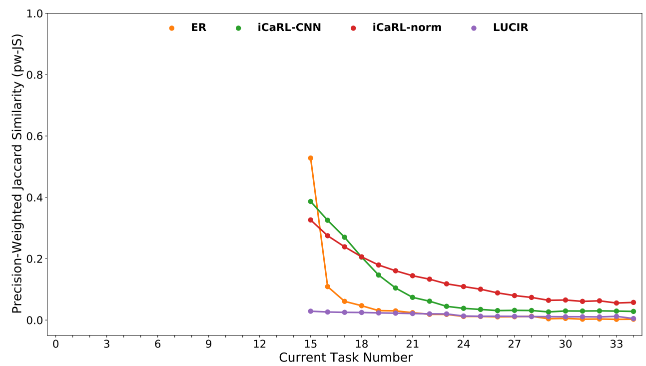
(a) Task 0



(b) Task 5

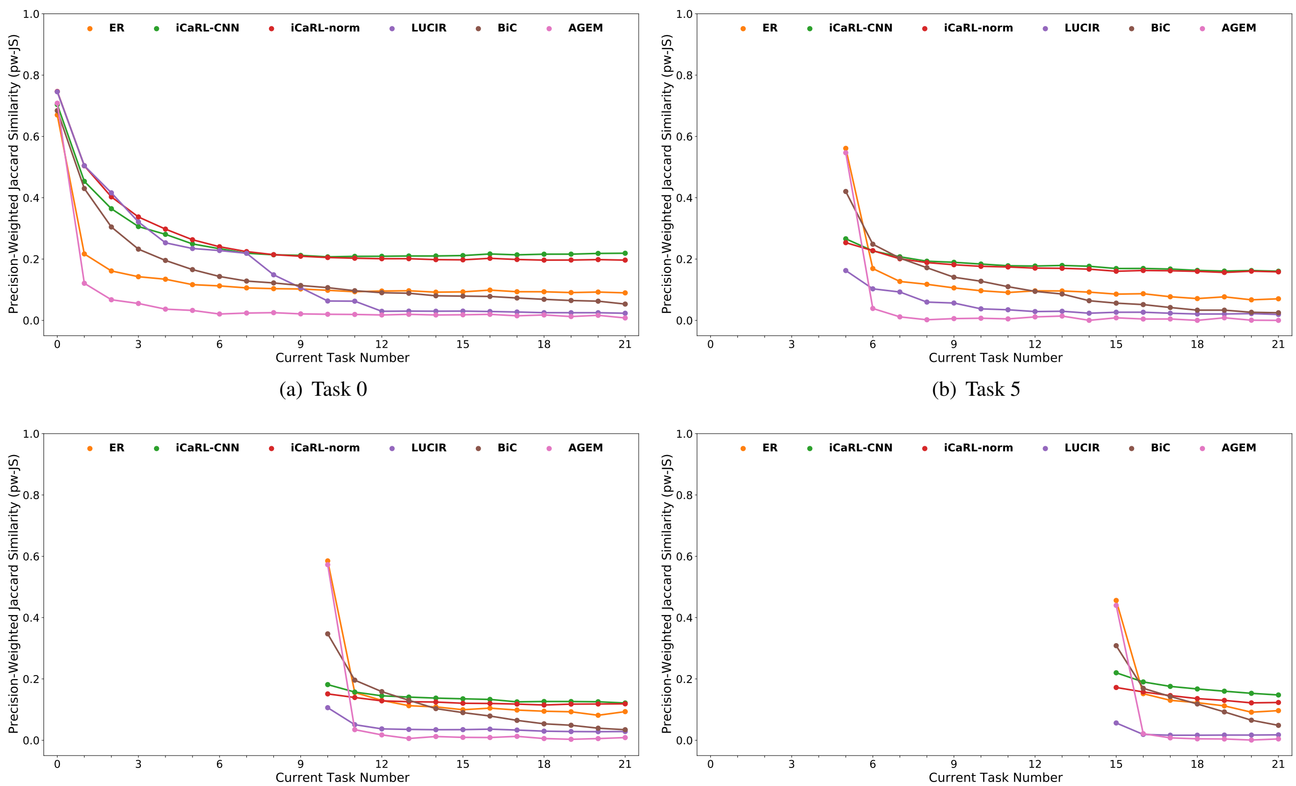


(c) Task 10



(d) Task 15

Figure A.16. IIRC-ImageNet-full Performance on four middle tasks throughout the whole training process, to measure their catastrophic forgetting and backward transfer. Note that a degradation in performance is not necessarily caused by catastrophic forgetting, as a new subclass of a previously observed superclass might be introduced and the model would be penalized for not applying that label retroactively. Experiments are averaged on ten different task configurations with the mean reported.



(a) Task 0

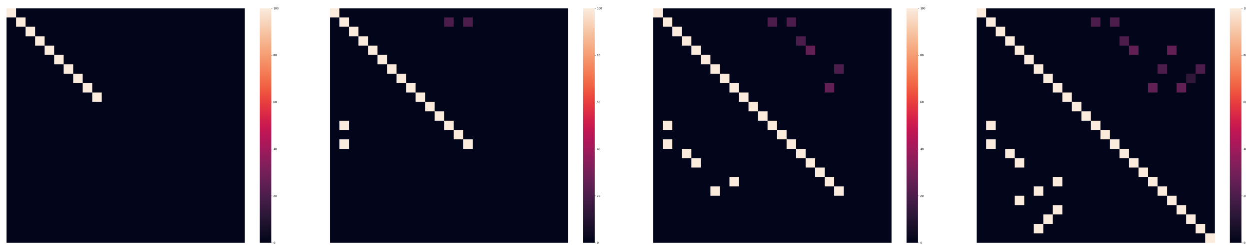
(b) Task 5

(c) Task 10

(d) Task 15

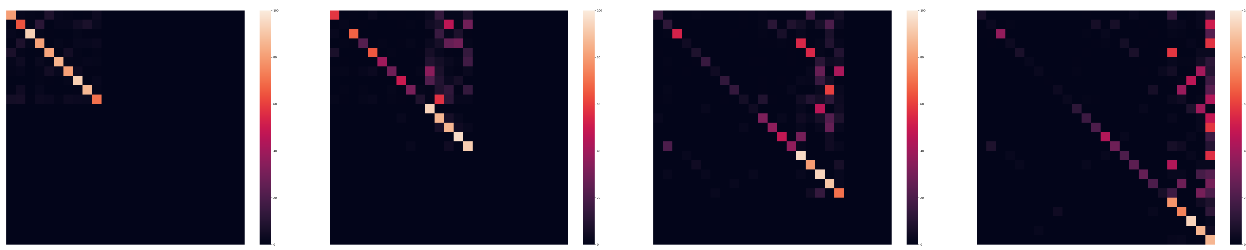
Figure A.17. IIRC-CIFAR Performance on four middle tasks throughout the whole training process, to measure their catastrophic forgetting and backward transfer. Note that a degradation in performance is not necessarily caused by catastrophic forgetting, as a new subclass of a previously observed superclass might be introduced and the model would be penalized for not applying that label retroactively. Experiments are averaged on ten different task configurations with the mean reported.

D.6. Confusion Matrix Over time



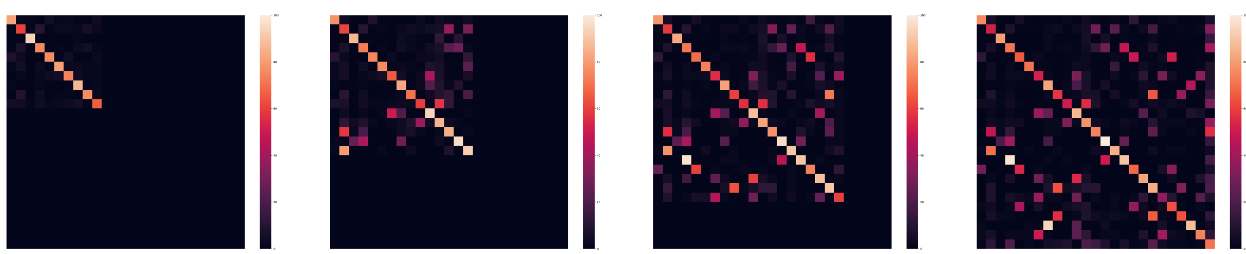
(a) After task 0 (b) After task 1 (c) After task 5 (d) After task 10

Figure A.18. Ground Truth confusion matrix after introducing tasks 0, 1, 5, 10 of IIRC-CIFAR respectively. The y-axis is the correct label (or one of the correct labels). The x-axis is the model predicted labels. Labels are arranged by their order of introduction. Only 25 labels are shown for better visibility.



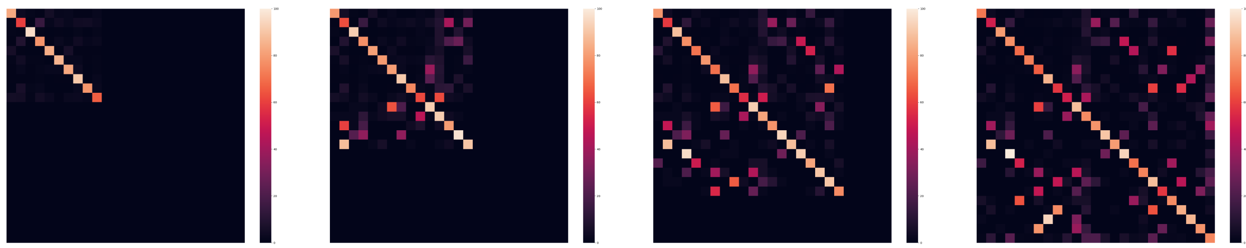
(a) After task 0 (b) After task 1 (c) After task 5 (d) After task 10

Figure A.19. ER confusion matrix after introducing tasks 0, 1, 5, 10 of IIRC-CIFAR respectively. The y-axis is the correct label (or one of the correct labels). The x-axis is the model predicted labels. Labels are arranged by their order of introduction. Only 25 labels are shown for better visibility.



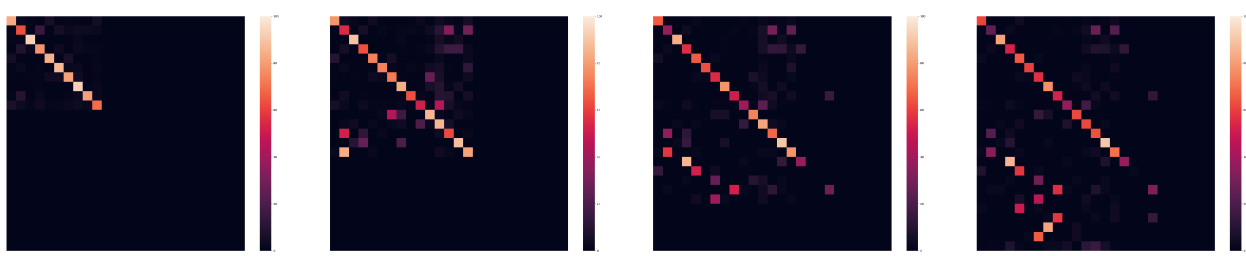
(a) After task 0 (b) After task 1 (c) After task 5 (d) After task 10

Figure A.20. iCaRL-CNN confusion matrix after introducing tasks 0, 1, 5, 10 of IIRC-CIFAR respectively. The y-axis is the correct label (or one of the correct labels). The x-axis is the model predicted labels. Labels are arranged by their order of introduction. Only 25 labels are shown for better visibility.



(a) After task 0 (b) After task 1 (c) After task 5 (d) After task 10

Figure A.21. iCaRL-norm confusion matrix after introducing tasks 0, 1, 5, 10 of IIRC-CIFAR respectively. The y-axis is the correct label (or one of the correct labels). The x-axis is the model predicted labels. Labels are arranged by their order of introduction. Only 25 labels are shown for better visibility.



(a) After task 0 (b) After task 1 (c) After task 5 (d) After task 10

Figure A.22. LUCIR confusion matrix after introducing tasks 0, 1, 5, 10 of IIRC-CIFAR respectively. The y-axis is the correct label (or one of the correct labels). The x-axis is the model predicted labels. Labels are arranged by their order of introduction. Only 25 labels are shown for better visibility.

D.7. Full Resolution Confusion Matrix

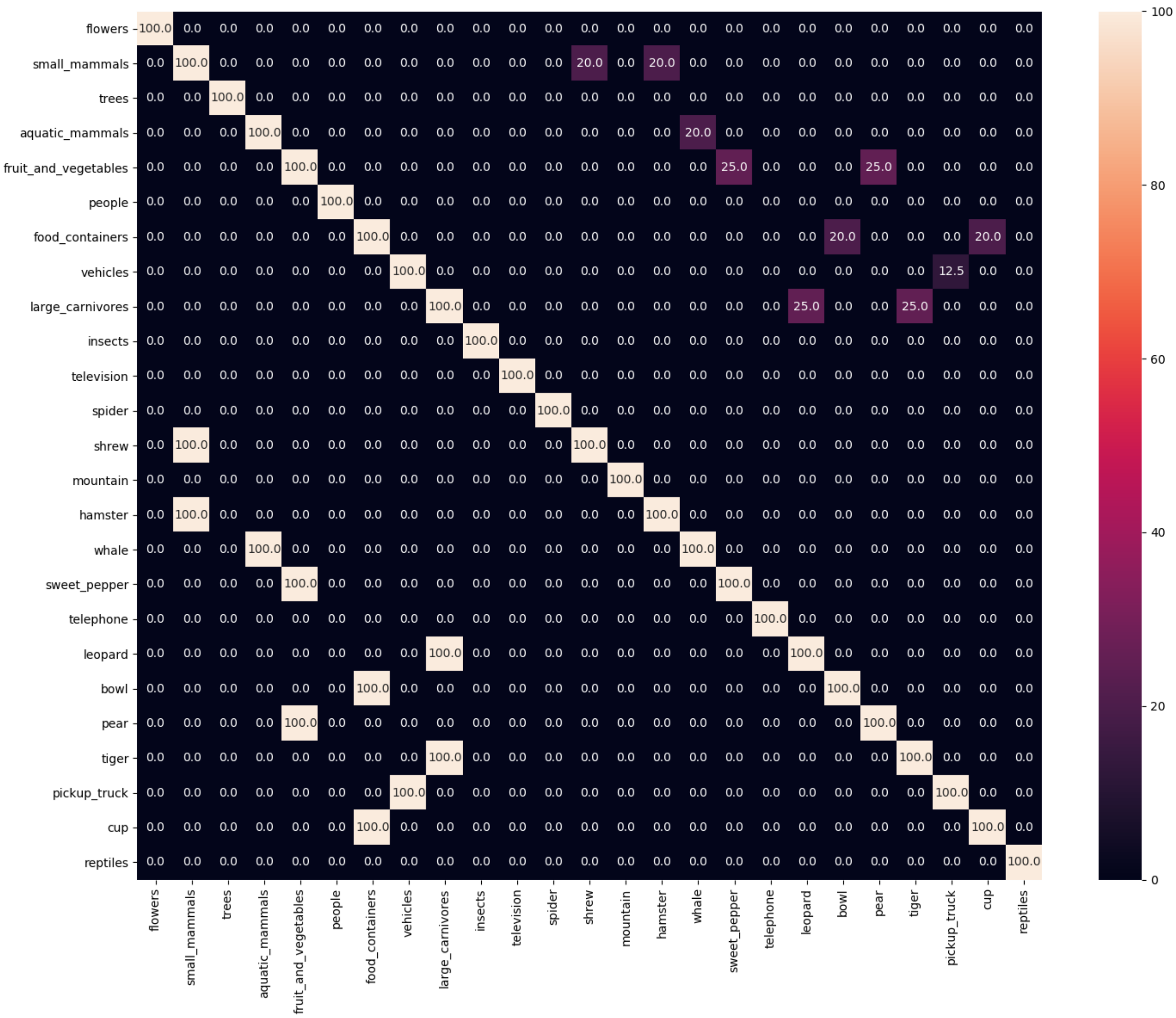


Figure A.23. Confusion matrix (ground truth) after training on task 10 of IIRC-CIFAR. the y-axis is the correct label (or one of the correct labels), the x-axis is the model predicted labels, The classes are arranged by their order of introduction. Only 25 classes are shown for better visibility. The y-axis represents the true label (or one of the true labels), while the x-axis represents the model predictions.

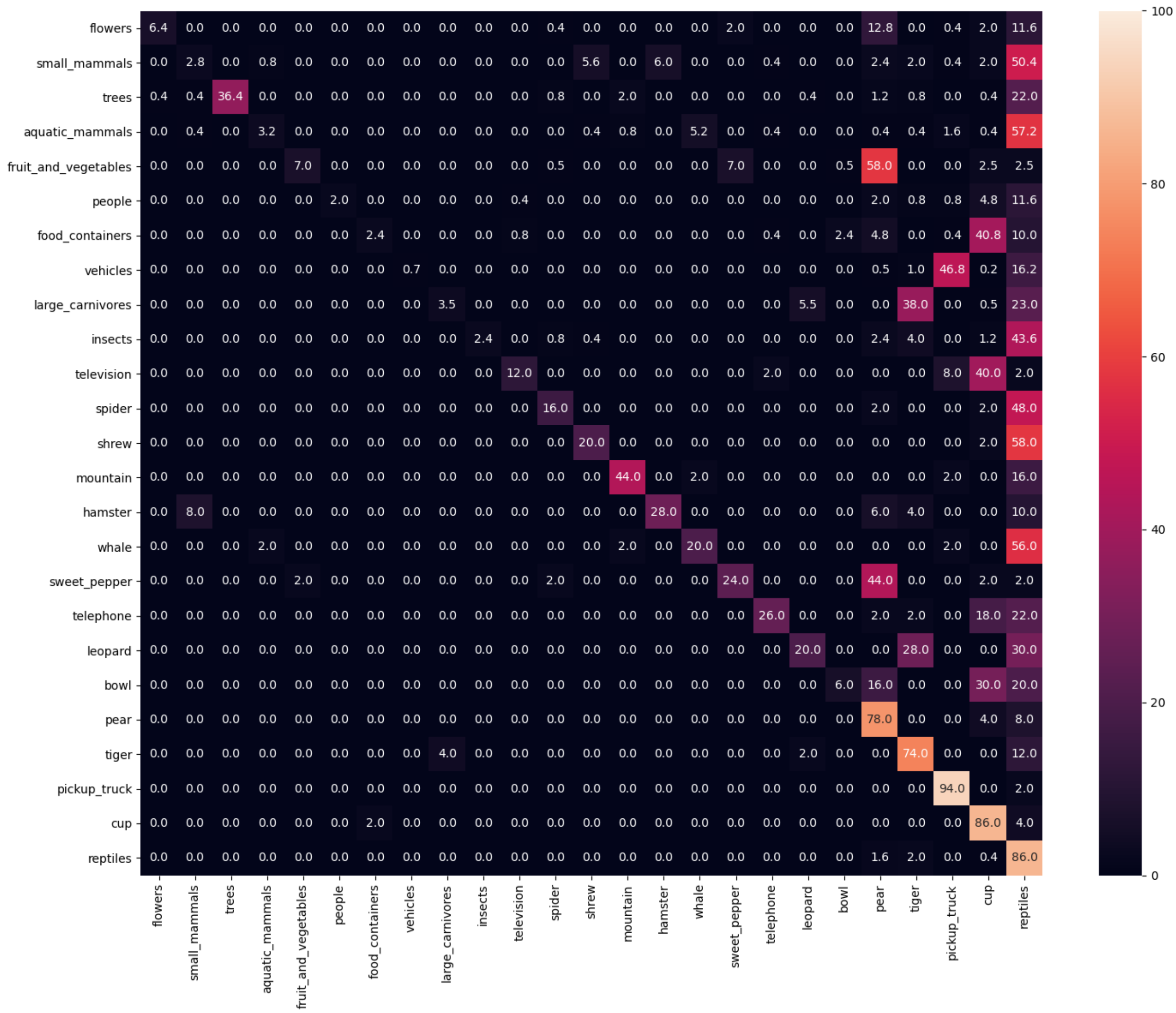


Figure A.24. Confusion matrix (ER) after training on task 10 of IIRC-CIFAR. the y-axis is the correct label (or one of the correct labels), the x-axis is the model predicted labels, The classes are arranged by their order of introduction. Only 25 classes are shown for better visibility. The y-axis represents the true label (or one of the true labels), while the x-axis represents the model predictions.

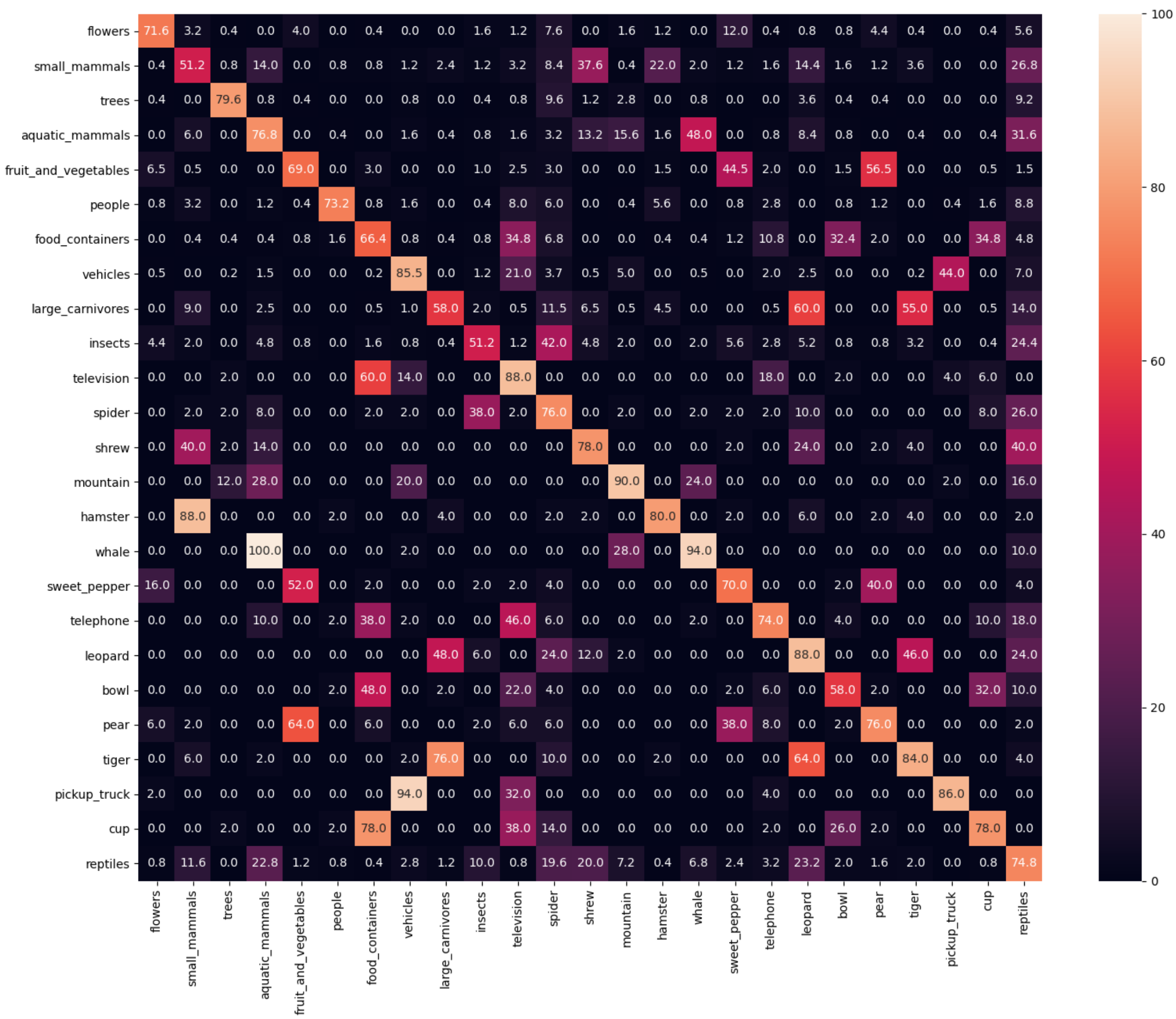


Figure A.25. Confusion matrix (iCaRL-norm) after training on task 10 of IIRC-CIFAR. The x-axis is the model predicted labels, The classes are arranged by their order of introduction. Only 25 classes are shown for better visibility. The y-axis represents the true label (or one of the true labels), while the x-axis represents the model predictions.

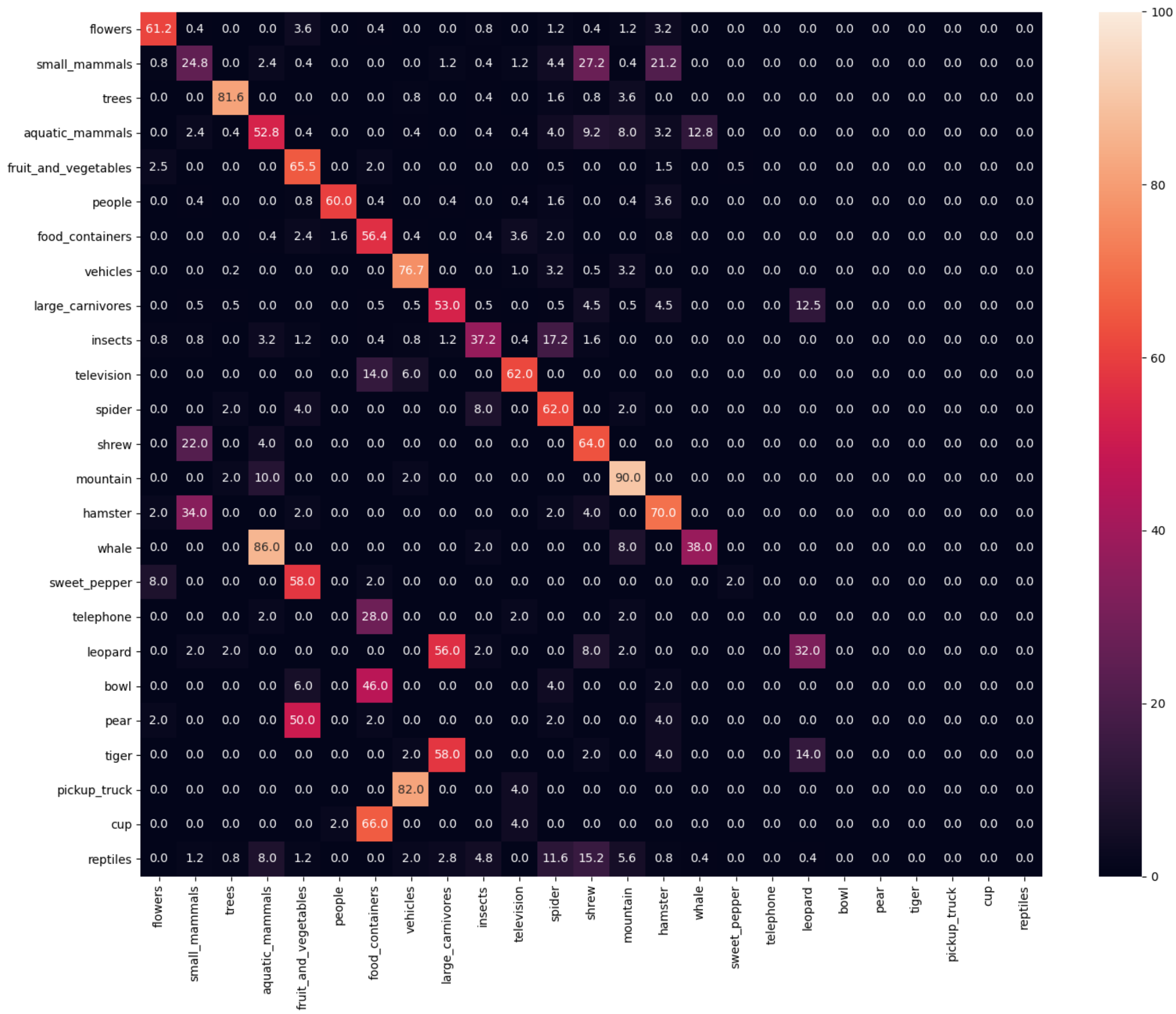


Figure A.26. Confusion matrix (LUCIR) after training on task 10 of IIRC-CIFAR. The x-axis is the model predicted labels, The classes are arranged by their order of introduction. Only 25 classes are shown for better visibility. The y-axis represents the true label (or one of the true labels), while the x-axis represents the model predictions.

E. Effect of Buffer

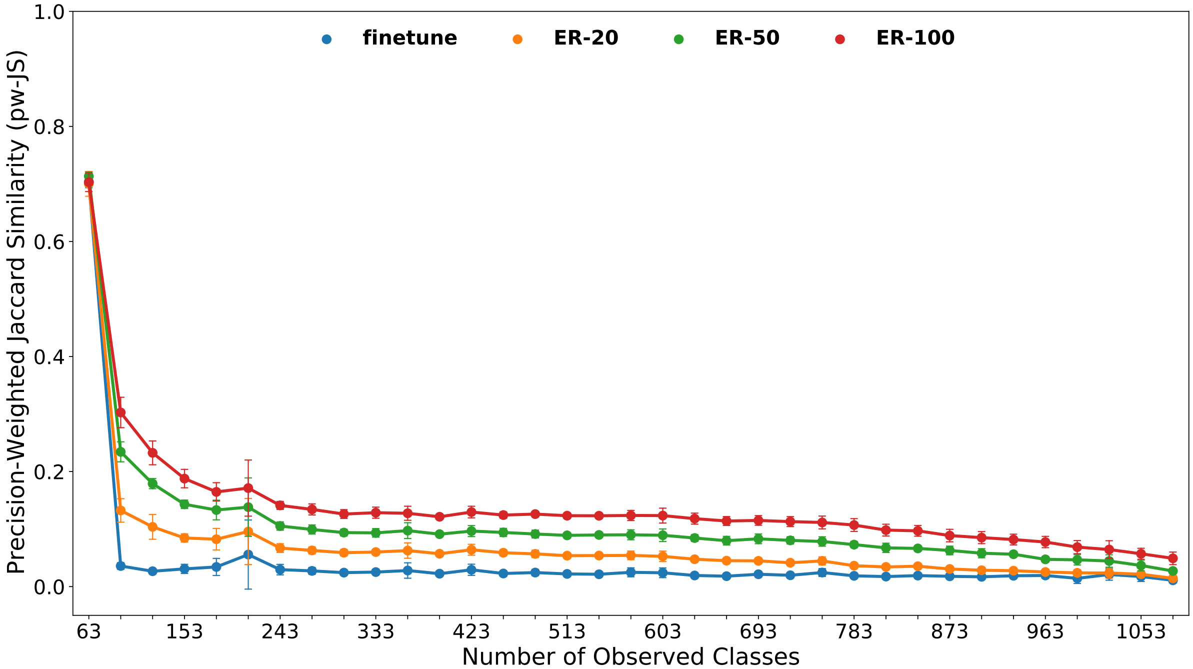


Figure A.27. ImageNet average performance using different buffer sizes, The number next to ER indicates the number of samples per class used for the replay buffer

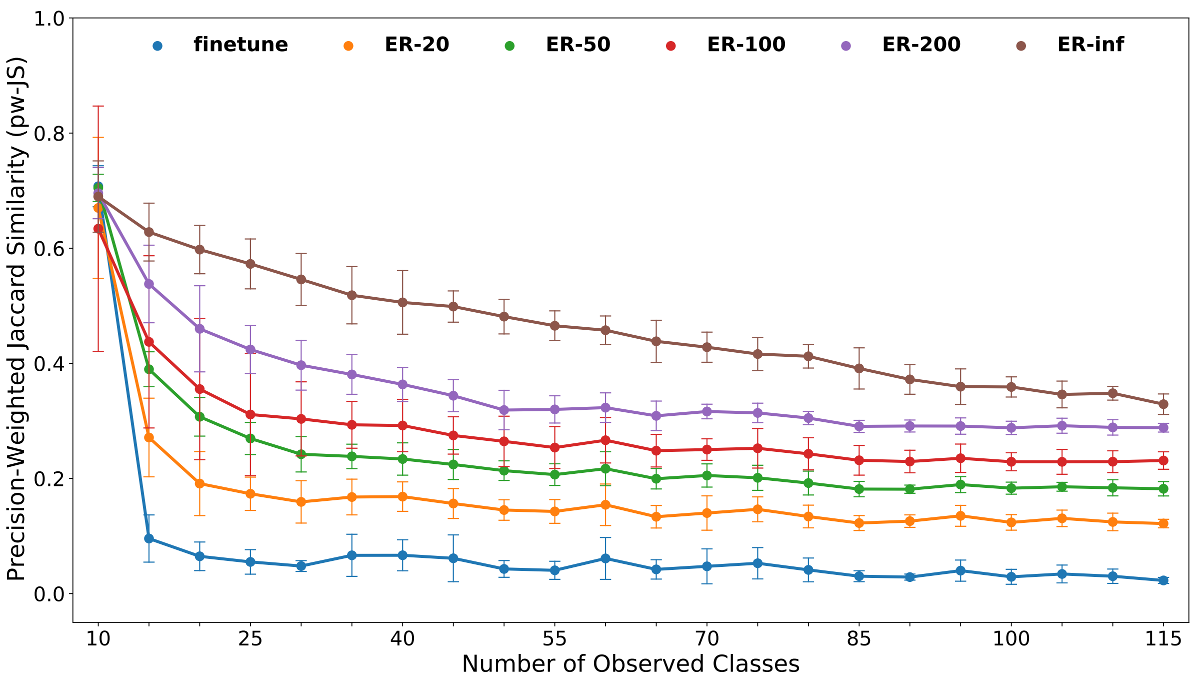


Figure A.28. CIFAR average performance using different buffer sizes, The number next to ER indicates the number of samples per class used for the replay buffer

F. Pseudo Codes

Algorithm 1: IncrementalTrain

```
Require: tasks        // A list of the classes to-be-introduced at each task
trainSet, validSet_inTask, validSet_postTask, testSet ← LoadDatasets()
model ← CreateModel()
/* create an empty buffer                                              */
buffer ← CreateBuffer()
for task in tasks do
    model ← TrainOnTask(model, buffer, trainingSet, validSet_inTask)
    /* add randomly selected samples to buffer                         */
    buffer ← AddToBuffer (buffer, trainingSet)
    PostTaskEvaluate(model, validSet_postTask, testSet)
end
```

Algorithm 2: LoadDatasets

```
input  : rawData_train       // The default single-label full dataset (train)
input  : rawData_test        // The default single-label full dataset (test)
input  : classHierarchy      // A dictionary that maps each superclass to its constituent subclasses
multilabeledData_train ← AddSuperclassLabels(rawData_train, classHierarchy)
multilabeledData_test ← AddSuperclassLabels(rawData_test, classHierarchy)
multilabeledData_train, multilabeledData_valid_inTask, multilabeledData_valid_postTask ←
  SplitData(multilabeledData_train)
trainSet ← IncompleteInfoIncrementalDataset(multilabeledData_train)
validSet_inTask ← IncompleteInfoIncrementalDataset(multilabeledData_valid_inTask)
validSet_postTask ← CompleteInfoIncrementalTestDataset(multilabeledData_valid_postTask)
testSet ← CompleteInfoIncrementalTestDataset(multilabeledData_test)
output : trainSet            // The incomplete information incremental learning training set
output : validSet_inTask     // The incomplete information incremental learning validation set (for in-task
         performance)
output : validSet_postTask   // The complete information incremental learning validation set (for post-task
         performance)
output : testSet             // The complete information incremental learning test set
```

Algorithm 3: IncompleteInfoIncrementalDataset

input : $multilabelData$ // A list of samples with each sample in the form of ($image$, ($superclassLabel,\ subclassLabel$)) or ($image$, ($subclassLabel$)))
input : $superclassToSubclass$ // a mapping that maps superclasses to their constituent subclasses
input : $tasks$ // The classes to-be-introduced at each task
Require: $subclasses$ // All refined subclasses (those who have a superclass as well as those who don't)
output : a dataset object with the data changing along the tasks

Initialization:
$classToDataIndices \leftarrow$ EmptyDictionary
$currentTaskId \leftarrow 0$
$dataIndices_{task} \leftarrow$ []
for *subclass in subclasses* **do**
  /* get the indices of the samples which correspond to this subclass */
  $dataIndices_{subclass} \leftarrow$ GetSamplesIndices($mtultilabelData,\ subclass$)
  if *subclass has superclass* **then**
    $dataSubsetLength_{superclass} \leftarrow$ 0.4 * Length($dataIndices_{subclass}$)
    $dataSubsetLength_{subclass} \leftarrow$ 0.8 * Length($dataIndices_{subclass}$)
    $dataIndices_{subclass} \leftarrow$ Shuffle($dataIndices_{subclass}$)
    $dataSubsetIndices_{subclass} \leftarrow dataIndices_{subclass}$[:$dataSubsetLength_{subclass}$]
    $dataSubsetIndices_{superclass} \leftarrow dataIndices_{subclass}$[-$dataSubsetLength_{superclass}$:]
    $classToDataIndices[subclass] \leftarrow dataIndices_{subclass}$
    $classToDataIndices[superclass] \leftarrow classToDataIndices[superclass] \cup dataSubsetIndices_{superclass}$
  end
  else if *subclass has no superclass* **then**
    $classToDataIndices$[subclass] $\leftarrow dataIndices_{subclass}$
  end
end

IncrementTask:
$currentTaskId \leftarrow currentTaskId + 1$
$dataIndices_{task} \leftarrow$ []
for *class in tasks[currentTaskId]* **do**
  $dataIndices_{task} \leftarrow dataIndices_{task} \cup classToDataIndices[class]$
end

GetItem:
Require: $classes_{task}$ // The classes present in the current task
input : $index$ // an index in the range of length of $dataIndices_{task}$
$image,\ labels \leftarrow multilabelData[dataIndices_{task}[index]]$
$label \leftarrow labels \cap classes_{task}$
output : $image$ // The sample image
output : $label$ // The label corresponding to this image that exists in the current task

Algorithm 4: CompleteInfoIncrementalTestDataset

input : $multilabelData$ // A list of samples with each sample in the form of ($image$, ($superclassLabel,\ subclassLabel$)) or ($image$, ($subclassLabel$)))
input : $superclassToSubclass$ // a mapping that maps superclasses to their constituent subclasses
input : $tasks$ // The classes available at each task
Require: $subclasses$ // All refined subclasses (those who have a superclass as well as those who don't)
output : a test dataset object which keeps collecting data along the tasks

Initialization:

$classToDataIndices \leftarrow$ empty_dictionary
$classes_{observed} \leftarrow []$
$dataIndices_{accessible} \leftarrow []$
for *subclass in subclasses* **do**
  /* get the indices of the samples which correspond to this subclass */
  $dataIndices_{subclass} \leftarrow$ GetSamplesIndices($multilabelData,\ subclass$)
  $classToDataIndices[subclass] \leftarrow dataIndices_{subclass}$
  if *subclass has superclass* **then**
    $classToDataIndices[superclass] \leftarrow classToDataIndices[superclass] \cup classToDataIndices[subclass]$
  end
end

LoadTask

input : $taskId$ // The index of the task to load
$dataIndices_{accessible} \leftarrow []$
$classes_{observed} \leftarrow classes_{observed} \cup tasks[taskId]$
for *class in tasks[taskId]* **do**
  $dataIndices_{accessible} \leftarrow dataIndices_{accessible} \cup classToDataIndices[class]$
end

LoadAllObservedData

Require: $classes_{observed}$ // All classes observed till now in all previous tasks
$dataIndices_{accessible} \leftarrow []$
for *class in* $classes_{observed}$ **do**
  $dataIndices_{accessible} \leftarrow dataIndices_{accessible} \cup classToDataIndices[class]$
end
$dataIndices_{accessible} \leftarrow$ RemoveDuplicates($dataIndices_{accessible}$)

GetItem

Require: $classes_{observed}$ // All classes observed till now in all previous tasks
input : $index$ // an index in the range of task_data_indices
$image,\ labels \leftarrow multilabelData[dataIndices_{accessible}[index]]$
$labels \leftarrow labels \cap classes_{observed}$
output : $image$ // The sample image
output : $labels$ // The labels corresponding to this image that exist in the $classes_{observed}$

G. IIRC Datasets Hierarchies

G.1. IIRC-CIFAR Hierarchy

superclass	subclasses
aquatic mammals	beaver, dolphin, otter, seal, whale
fish	aquarium fish, flatfish, ray, shark, trout
flowers	orchid, poppy, rose, sunflower, tulip
food containers	bottle, bowl, can, cup, plate
fruit and vegetables	apple, orange, pear, sweet pepper
household furniture	bed, chair, couch, table, wardrobe
insects	bee, beetle, butterfly, caterpillar, cockroach
large carnivores	leopard, lion, tiger, wolf
large omnivores and herbivores	bear, camel, cattle, chimpanzee, elephant, kangaroo
medium sized mammals	fox, porcupine, possum, raccoon, skunk
people	baby, boy, girl, man, woman
reptiles	crocodile, dinosaur, lizard, snake, turtle
small mammals	hamster, mouse, rabbit, shrew, squirrel
trees	maple tree, oak tree, palm tree, pine tree, willow tree
vehicles	bicycle, bus, motorcycle, pickup truck, train, streetcar, tank, tractor
-	mushroom, clock, keyboard, lamp, telephone, television, bridge, castle, house, road, skyscraper, cloud, forest, mountain, plain, sea, crab, lobster, snail, spider, worm, lawn mower, rocket

G.2. IIRC-ImageNet Hierarchy

superclass	subclasses
dog	dalmatian, basenji, pug, Leonberg, Newfoundland, Great Pyrenees, Mexican hairless, Brabancon griffon, Pembroke, Cardigan, Chihuahua, Japanese spaniel, Maltese dog, Pekinese, Shih-Tzu, toy terrier, papillon, Blenheim spaniel, Rhodesian ridgeback, boxer, bull mastiff, Great Dane, Saint Bernard, Eskimo dog, Tibetan mastiff, French bulldog, malamute, Siberian husky, Samoyed, Pomeranian, chow, keeshond, toy poodle, miniature poodle, standard poodle, Afghan hound, basset, beagle, bloodhound, bluetick, redbone, Ibizan hound, Norwegian elkhound, otterhound, Saluki, Scottish deerhound, Weimaraner, black-and-tan coonhound, Walker hound, English foxhound, borzoi, Irish wolfhound, Italian greyhound, whippet, Bedlington terrier, Border terrier, Kerry blue terrier, Irish terrier, Norfolk terrier, Norwich terrier, Yorkshire terrier, Airedale, cairn, Australian terrier, Dandie Dinmont, Boston bull, Scotch terrier, Tibetan terrier, silky terrier, soft-coated wheaten terrier, West Highland white terrier, Lhasa, Staffordshire bullterrier, American Staffordshire terrier, wire-haired fox terrier, Lakeland terrier, Sealyham terrier, German short-haired pointer, vizsla, kuvasz, schipperke, Doberman, miniature pinscher, affenpinscher, Brittany spaniel, clumber, cocker spaniel, Sussex spaniel, English springer, Welsh springer spaniel, Irish water spaniel, English setter, Irish setter, Gordon setter, flat-coated retriever, curly-coated retriever, golden retriever, Labrador retriever, Chesapeake Bay retriever, miniature schnauzer, giant schnauzer, standard schnauzer, Greater Swiss Mountain dog, Bernese mountain dog, Appenzeller, EntleBucher, briard, kelpie, komondor, Old English sheepdog, Shetland sheepdog, collie, Border collie, Bouvier des Flandres, Rottweiler, German shepherd, groenendael, malinois
bird	cock, hen, ostrich, bee eater, hornbill, hummingbird, jacamar, toucan, coucal, quail, partridge, peacock, black grouse, ptarmigan, ruffed grouse, prairie chicken, water ouzel, robin, bulbul, jay, magpie, chickadee, brambling, goldfinch, house finch, junco, indigo bunting, black swan, European gallinule, goose, drake, red-breasted merganser, pelican, albatross, king penguin, spoonbill, flamingo, limpkin, bustard, white stork, black stork, American coot, oystercatcher, red-backed sandpiper, redshank, dowitcher, ruddy turnstone, little blue heron, bittern, American egret, African grey, macaw, sulphur-crested cockatoo, lorikeet, vulture, kite, bald eagle, great grey owl
garment	suit, abaya, kimono, cardigan, feather boa, stole, jersey, sweatshirt, poncho, brassiere, jean, gown, military uniform, pajama, apron, academic gown, vestment, bow tie, Windsor tie, fur coat, lab coat, trench coat, hoopskirt, miniskirt, overskirt, sarong, cloak
beverage	espresso, red wine, cup, eggnog
aircraft	airship, balloon, airliner, warplane, wing, space shuttle
bear	brown bear, American black bear, ice bear, sloth bear
fox	red fox, kit fox, Arctic fox, grey fox
wolf	timber wolf, white wolf, red wolf, coyote
bag	backpack, mailbag, plastic bag, purse, sleeping bag
footwear	clog, cowboy boot, Loafer, running shoe, sandal
toiletry	hair spray, lotion, perfume, face powder, sunscreen, lipstick
box	carton, chest, crate, mailbox, pencil box, safe
rodent	hamster, porcupine, marmot, beaver, guinea pig, fox squirrel
bottle	beer bottle, pill bottle, pop bottle, water bottle, wine bottle, water jug, whiskey jug
fabric	velvet, wool, bib, dishrag, handkerchief, bath towel, paper towel

cup	beer glass, goblet, cocktail shaker, measuring cup, pitcher, beaker, coffee mug
fungus	coral fungus, gyromitra, stinkhorn, earthstar, hen-of-the-woods, bolete, agaric
musteline	weasel, mink, polecat, black-footed ferret, otter, skunk, badger
truck	fire engine, garbage truck, pickup, tow truck, trailer truck, moving van, police van, recreational vehicle, forklift, harvester, snowplow, tractor
headdress	crash helmet, football helmet, bearskin, bonnet, cowboy hat, sombrero, bathing cap, mortarboard, shower cap, pickelhaube
ball	baseball, basketball, croquet ball, golf ball, ping-pong ball, punching bag, rugby ball, soccer ball, tennis ball, volleyball
car	ambulance, beach wagon, cab, convertible, jeep, limousine, Model T, racer, sports car, minivan, grille, golfcart
measuring instrument	barometer, scale, odometer, rule, sundial, digital watch, hourglass, parking meter, stopwatch, analog clock, digital clock, wall clock
tool	hammer, plunger, screwdriver, shovel, cleaver, letter opener, can opener, corkscrew, hatchet, chain saw, plane, scabbard, power drill, carpenter's kit
watercraft	schooner, catamaran, trimaran, fireboat, gondola, canoe, yawl, lifeboat, speedboat, pirate, wreck, container ship, liner, aircraft carrier, submarine, amphibian, paddle
dish	Petri dish, mixing bowl, soup bowl, tray
bus	minibus, school bus, trolleybus
cart	horse cart, jinrikisha, oxcart
tracked vehicle	snowmobile, half track, tank
lamp	candle, spotlight, jack-o'-lantern, lampshade, table lamp
optical instrument	binoculars, projector, sunglasses, lens cap, loupe, Polaroid camera, reflex camera
gymnastic apparatus	balance beam, horizontal bar, parallel bars
swine	hog, wild boar, warthog
rabbits	hare, wood rabbit, Angora
echinoderm	starfish, sea urchin, sea cucumber
wild dog	dingo, dhole, African hunting dog
pouched mammal	wombat, wallaby, koala
aquatic mammal	dugong, grey whale, killer whale, sea lion
person	ballplayer, scuba diver, groom
mollusk	chiton, chambered nautilus, conch, snail, slug, sea slug
weapon	bow, projectile, cannon, missile, rifle, revolver, assault rifle, holster
bovid	bison, water buffalo, ram, ox, bighorn, ibex, hartebeest, impala, gazelle
salamander	European fire salamander, common newt, eft, spotted salamander, axolotl
frog	tree frog, tailed frog, bullfrog
big cat	leopard, snow leopard, jaguar, lion, tiger, cheetah
domestic cat	tabby, tiger cat, Persian cat, Siamese cat, Egyptian cat
cooking utensil	spatula, frying pan, wok, Crock Pot, Dutch oven, caldron, coffeepot, teapot
primate	Madagascar cat, indri, gibbon, siamang, orangutan, gorilla, chimpanzee, marmoset, capuchin, howler monkey, titi, spider monkey, squirrel monkey, guenon, patas, baboon, macaque, langur, colobus, proboscis monkey
fish	barracouta, electric ray, stingray, hammerhead, great white shark, tiger shark, sturgeon, gar, puffer, rock beauty, anemone fish, lionfish, eel, tench, goldfish, coho
lizard	banded gecko, common iguana, American chameleon, whiptail, agama, frilled lizard, alligator lizard, Gila monster, green lizard, African chameleon, Komodo dragon

turtle	mud turtle, terrapin, box turtle, loggerhead, leatherback turtle
spider	black and gold garden spider, barn spider, garden spider, black widow, tarantula, wolf spider, spider web
insect	ringlet, sulphur butterfly, lycaenid, cabbage butterfly, monarch, admiral, dragonfly, damselfly, lacewing, cicada, leafhopper, cockroach, mantis, walking stick, grasshopper, cricket, bee, ant, fly, tiger beetle, ladybug, ground beetle, long-horned beetle, leaf beetle, weevil, dung beetle, rhinoceros beetle
green groceries	acorn, hip, ear, fig, pineapple, banana, jackfruit, custard apple, pomegranate, strawberry, orange, lemon, Granny Smith, buckeye, rapeseed, corn, cucumber, artichoke, cardoon, mushroom, bell pepper, mashed potato, zucchini, spaghetti squash, acorn squash, butternut squash, broccoli, cauliflower, head cabbage
keyboard instrument	accordion, organ, grand piano, upright
percussion instrument	chime, drum, gong, maraca, marimba, steel drum
stringed instrument	banjo, acoustic guitar, electric guitar, cello, violin, harp
wind instrument	ocarina, harmonica, flute, panpipe, bassoon, oboe, sax, cornet, French horn, trombone
crustacean	isopod, crayfish, hermit crab, spiny lobster, American lobster, Dungeness crab, rock crab, fiddler crab, king crab
pen	ballpoint, fountain pen, quill
display	desktop computer, laptop, notebook, screen, television, monitor
electronic equipement	cassette player, CD player, modem, oscilloscope, tape player, iPod, printer, joystick, dial telephone, pay-phone, cellular telephone, mouse, hand-held computer
snake	sea snake, horned viper, boa constrictor, rock python, Indian cobra, green mamba, diamondback, sidewinder, thunder snake, ringneck snake, hognose snake, green snake, king snake, garter snake, water snake, vine snake, night snake
geological formation	cliff, geyser, lakeside, seashore, valley, promontory, alp, volcano, coral reef, sandbar
food	dough, guacamole, chocolate sauce, carbonara, French loaf, bagel, pretzel, plate, trifle, ice cream, ice lolly, pizza, potpie, burrito, consomme, hot pot, hotdog, cheeseburger, meat loaf
white home appliances	dishwasher, refrigerator, washer, stove
kitchen appliances	microwave, toaster, waffle iron, espresso maker
wheel	car wheel, paddlewheel, pinwheel, potter's wheel, reel, disk brake
seat	toilet seat, studio couch, park bench, barber chair, folding chair, rocking chair, throne
baby bed	bassinet, cradle, crib
cabinet	medicine chest, wardrobe, china cabinet, bookcase, chiffonier, file, entertainment center, plate rack
table	desk, pool table, dining table
bridges	steel arch bridge, suspension bridge, viaduct
fence	chainlink fence, picket fence, stone wall, worm fence
long structures	beacon, obelisk, totem pole
movable homes	mountain tent, mobile home, yurt
building	planetarium, barn, cinema, boathouse, palace, monastery, castle, dome, church, mosque, stupa, bell cote, thatch, tile roof, triumphal arch
body armor	chain mail, cuirass, bulletproof vest, breastplate
mask	mask, oxygen mask, gasmask, ski mask
curtain-screen	window shade, shower curtain, theater curtain
bike	moped, bicycle-built-for-two, tricycle, unicycle, mountain bike, motor scooter

train	passenger car, freight car, electric locomotive, bullet train, streetcar, steam locomotive
swimsuit	bikini, maillot, swimming trunks
socks mittens	Christmas stocking, mitten, sock
keyboard	computer keyboard, space bar, typewriter keyboard
-	African crocodile, American alligator, triceratops, trilobite, harvestman, scorpion, tick, centipede, tusker, echidna, platypus, jellyfish, sea anemone, brain coral, flatworm, nematode, crane, hyena, cougar, lynx, mongoose, meerkat, sorrel, zebra, hippopotamus, Arabian camel, llama, armadillo, three-toed sloth, Indian elephant, African elephant, lesser panda, giant panda, abacus, altar, apiary, ashcan, bakery, Band Aid, bannister, barbell, barbershop, barrel, barrow, bathtub, binder, birdhouse, bobsled, bolo tie, bookshop, bottlecap, brass, breakwater, broom, bucket, buckle, butcher shop, car mirror, carousel, cash machine, cassette, chain, cliff dwelling, coil, combination lock, confectionery, crutch, dam, diaper, dock, dogsled, doormat, drilling platform, drumstick, dumbbell, electric fan, envelope, fire screen, flagpole, fountain, four-poster, gas pump, go-kart, greenhouse, grocery store, guillotine, hair slide, hamper, hand blower, hard disc, home theater, honeycomb, hook, iron, jigsaw puzzle, knee pad, knot, ladle, lawn mower, library, lighter, loudspeaker, lumbermill, magnetic compass, manhole cover, matchstick, maypole, maze, megalith, microphone, milk can, mortar, mosquito net, mousetrap, muzzle, nail, neck brace, necklace, nipple, oil filter, packet, padlock, paintbrush, parachute, patio, pedestal, pencil sharpener, photocopier, pick, pier, piggy bank, pillow, plow, pole, pot, prayer rug, prison, puck, quilt, racket, radiator, radio, radio telescope, rain barrel, remote control, restaurant, rotisserie, rubber eraser, safety pin, saltshaker, scoreboard, screw, seat belt, sewing machine, shield, shoe shop, shoji, shopping basket, shopping cart, ski, slide rule, sliding door, slot, snorkel, soap dispenser, solar dish, space heater, spindle, stage, stethoscope, strainer, stretcher, sunglass, swab, swing, switch, syringe, teddy, thimble, thresher, tobacco shop, torch, toyshop, tripod, tub, turnstile, umbrella, vacuum, vase, vault, vending machine, wallet, washbasin, water tower, whistle, wig, window screen, wooden spoon, web site, comic book, crossword puzzle, street sign, traffic light, book jacket, menu, hay, bubble, daisy, yellow lady's slipper, toilet tissue