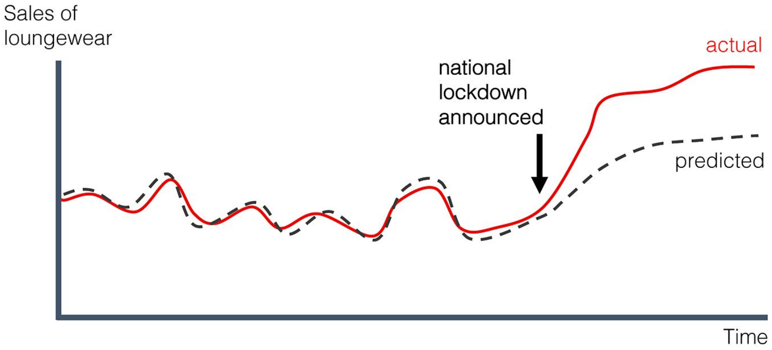
INSTITUTE FOR APPLIED COMPUTATIONAL SCIENCE
AT HARVARD UNIVERSITY

AC 297R: Mosaic ML Milestone 2

Xingyu Liu, Alex Leonardi,
Chris Gilmer-Hill, Lu Yu

Background: Model Retraining

- Motivation: avoid expensive model retraining to **save time and resources**
- Core Problem: Distribution Drift - **changing distributions make models grow less accurate over time** (cf. example)
- If retraining is inevitable, we should focus on making it more efficient



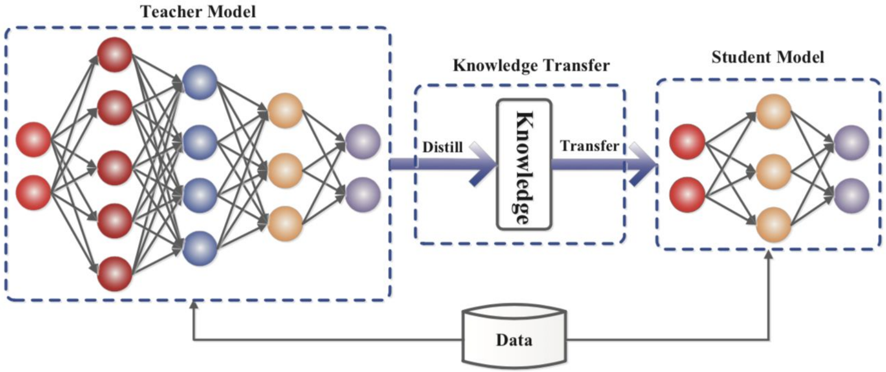
https://static.tildacdn.com/tild6630-3633-4262-b231-626462323634/sudden_concept_drift.png

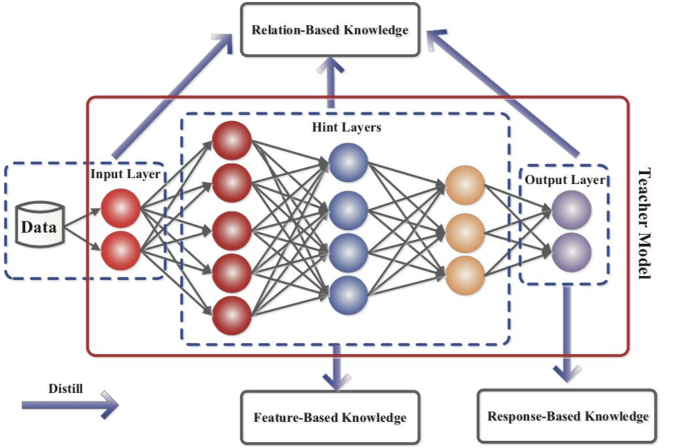
Project Statement:

How can we reuse the computation that was invested in training our initial models to make training future models better?

Background: Knowledge Distillation

- Goal: re-use information encoded in previous model iterations
- Increase efficiency of training process, in terms of:
  - **Time**, in terms of the number of epochs/iterations needed during training
  - **Resources**, in terms of the compute resources required for the training process





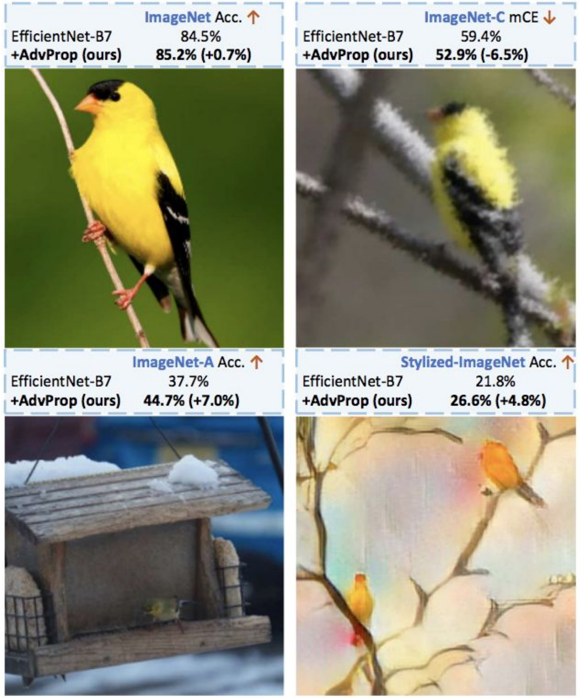
Gou et al 2021

Background: KD + Mixup

- Interpolation between model iterations
- Simplest implementation: train model on linear combination of outputs from prior model iterations
- Expected improvements: increase in time efficiency
  - Decreased number of training epochs required to achieve a given accuracy
  - Increased accuracy after a given number of training epochs

Background: Longer-Term Strategies

- Curriculum Learning:
  - Segment data by difficulty
  - Gradually increase difficulty over time
- Adversarial Recycling:
  - Leverage adversarial inputs from prior model runs
  - Improve robustness to distribution shifts in general



Dataset, Model, & Metrics

Dataset

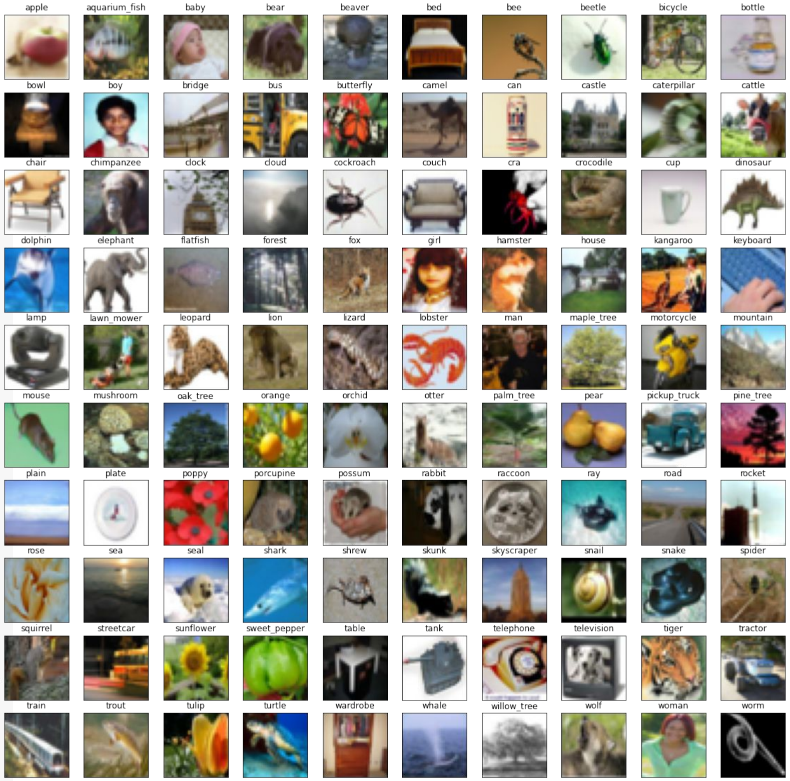
- What is CIFAR?
    - Canadian Institute for Advanced Research, a subset of the Tiny Images dataset
    - CIFAR-100: 50,000 training and 10,000 test images of 20 object classes, along with 100 object subclasses.
    - Each image is an RGB image of size 32x32, 3 channels (RGB).

- Why we choose CIFAR-100?
    - Our interest: model performance
    - Dataset size: time spent on predicting test data

source: https://www.geeksforgeeks.org/residual-networks-resnet-deep-learning/

Dataset - Intro

```
{0: 'apple', 1: 'aquarium_fish', 2: 'baby', 3: 'bear', 4: 'beaver',
 5: 'bed', 6: 'bee', 7: 'beetle', 8: 'bicycle', 9: 'bottle',
 10: 'bowl', 11: 'boy', 12: 'bridge', 13: 'bus', 14: 'butterfly',
 15: 'camel', 16: 'can', 17: 'castle', 18: 'caterpillar', 19: 'cattle',
 20: 'chair', 21: 'chimpanzee', 22: 'clock', 23: 'cloud', 24: 'cockroach',
 25: 'couch', 26: 'cra', 27: 'crocodile', 28: 'cup', 29: 'dinosaur',
 30: 'dolphin', 31: 'elephant', 32: 'flatfish' ,33: 'forest', 34: 'fox',
 35: 'girl', 36: 'hamster', 37: 'house', 38: 'kangaroo', 39: 'keyboard',
 40: 'lamp', 41: 'lawn_mower', 42: 'leopard', 43: 'lion', 44: 'lizard',
 45: 'lobster', 46: 'man', 47: 'maple_tree', 48: 'motorcycle', 49: 'mountain',
 50: 'mouse', 51: 'mushroom', 52: 'oak_tree', 53: 'orange', 54: 'orchid',
 55: 'otter', 56: 'palm_tree', 57: 'pear', 58: 'pickup_truck', 59: 'pine_tree',
 60: 'plain', 61: 'plate', 62: 'poppy', 63: 'porcupine', 64: 'possum',
 65: 'rabbit', 66: 'raccoon', 67: 'ray', 68: 'road', 69: 'rocket',
 70: 'rose', 71: 'sea', 72: 'seal', 73: 'shark', 74: 'shrew',
 75: 'skunk', 76: 'skyscraper', 77: 'snail', 78: 'snake', 79: 'spider',
 80: 'squirrel', 81: 'streetcar', 82: 'sunflower', 83: 'sweet_pepper', 84: 'table',
 85: 'tank', 86: 'telephone', 87: 'television', 88: 'tiger', 89: 'tractor',
 90: 'train', 91: 'trout', 92: 'tulip', 93: 'turtle', 94: 'wardrobe',
 95: 'whale', 96: 'willow_tree', 97: 'wolf', 98: 'woman', 99: 'worm'}
```

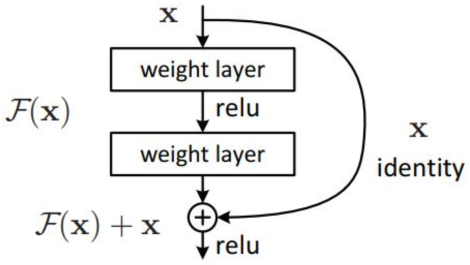


CIFAR-100

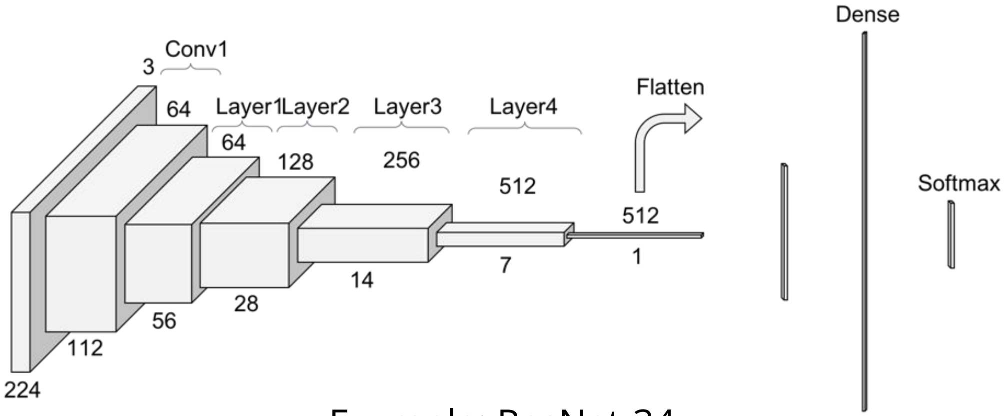
notebook: https://colab.research.google.com/drive/13RMWg3ZJQ5V7QRpgD4INZZ2KU3r1W_Oh?usp=sharing

Baseline Model

- ResNet-56
- Loss function: cross entropy loss
- Optimizer: SGD with exponential learning rate
- Epochs = 200, batch size = 128



Residual learning: a building block



Example: ResNet-34

Metrics

- Fixed # epochs vs. accuracy

  Highest accuracy the model could get to

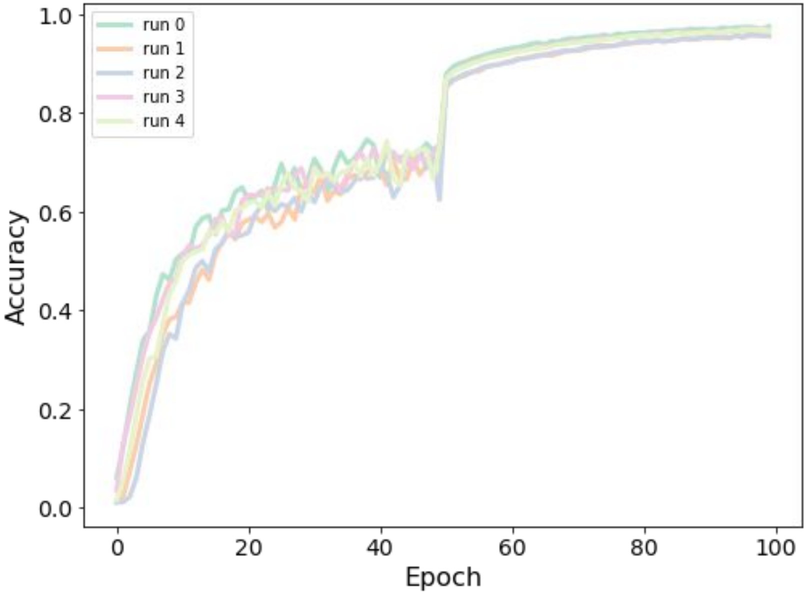
- # epochs vs. fixed accuracy

  Number of epochs the model takes to achieve target accuracy

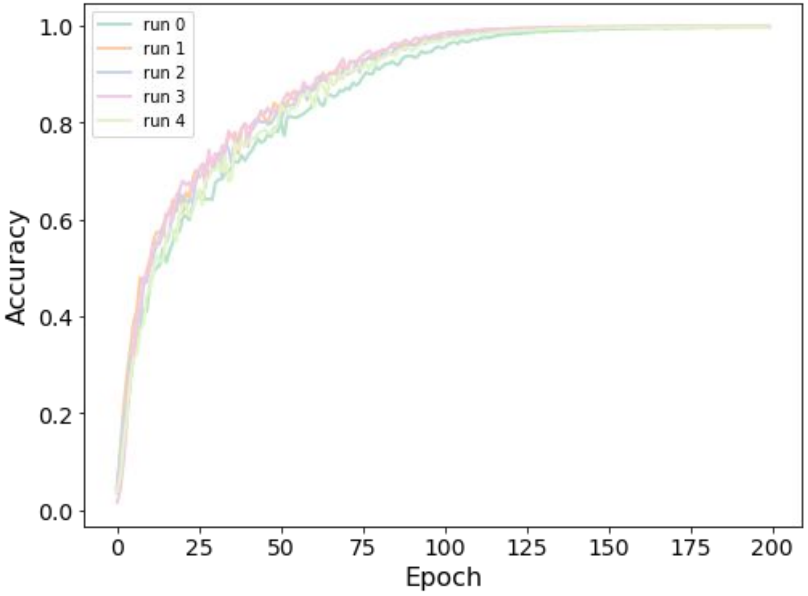
- Training time v.s accuracy

Results

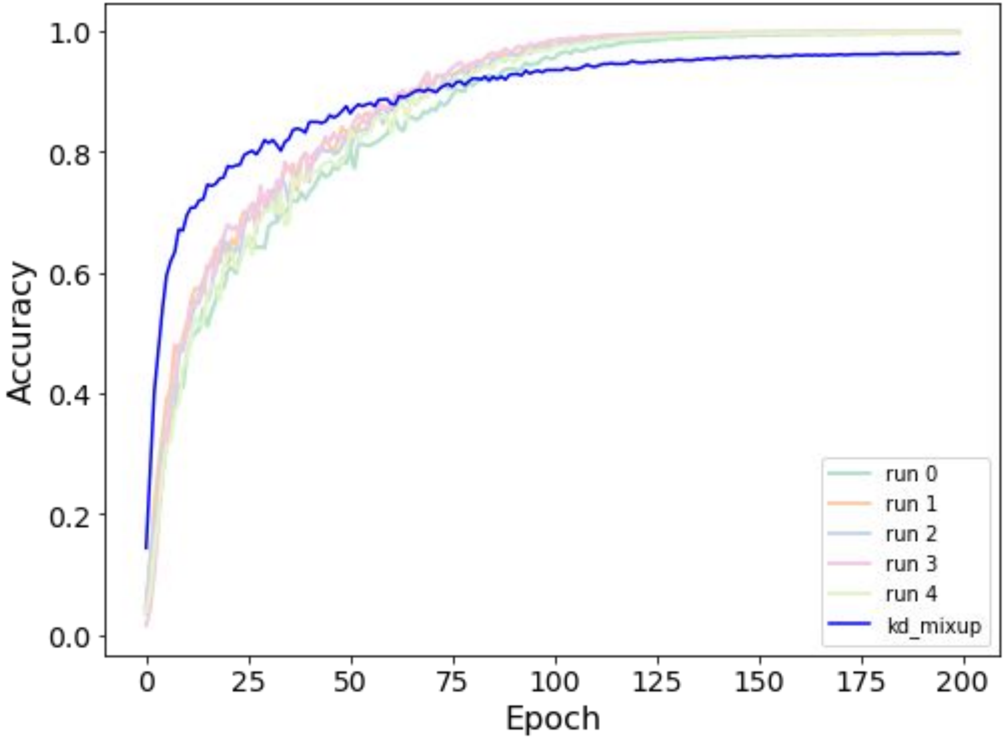
Baseline



Change
Scheduler



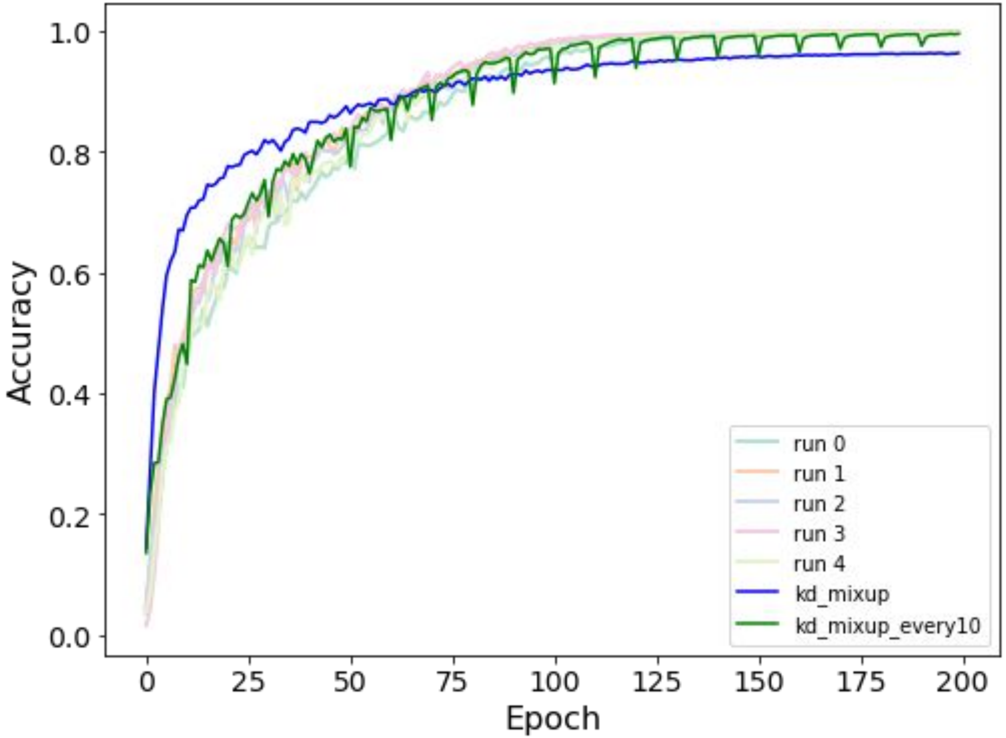
Knowledge Distillation and MixUp



Use Knowledge Distillation and MixUp for all 200 epochs

```
Loss =
cross_entropy(student_out, target) +
mse(student_out,teachers_mixup_out)
```

Knowledge Distillation and MixUp



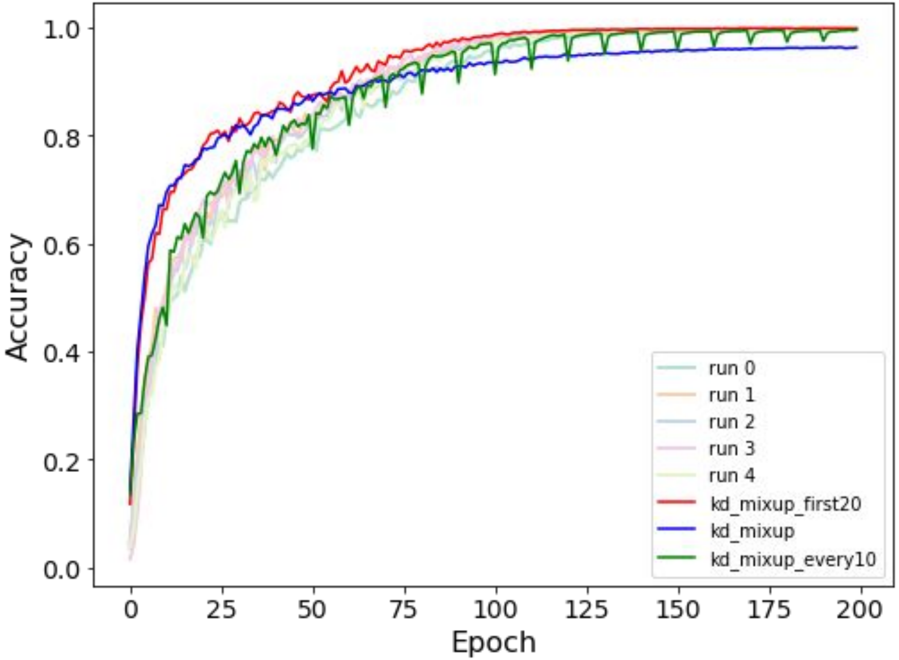
Use Knowledge Distillation and MixUp every 10 epochs

```
Loss =
cross_entropy(student_out, target) +
mse(student_out,teachers_mixup_out)
```

Otherwise use the original loss

```
Loss = cross_entropy(student_out,
target)
```

Knowledge Distillation and MixUp



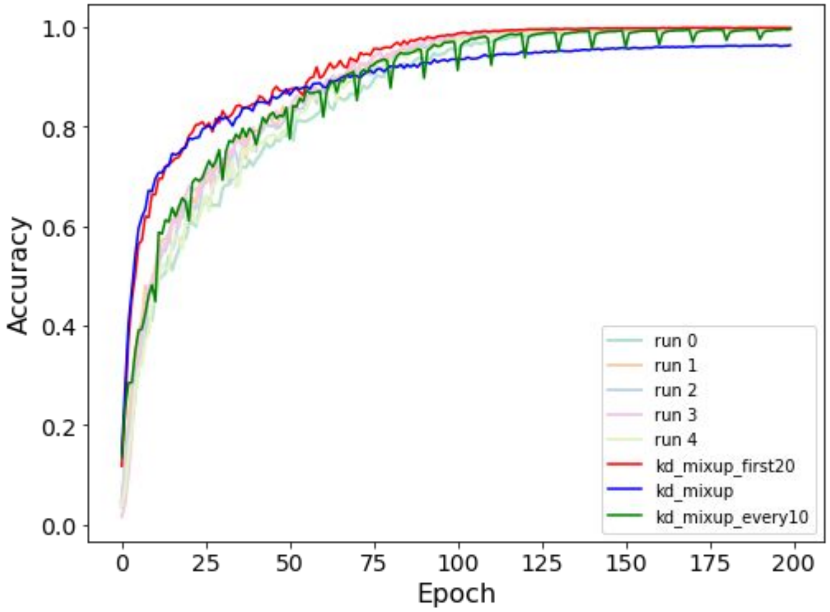
Use Knowledge Distillation and MixUp
for the first 20 epochs

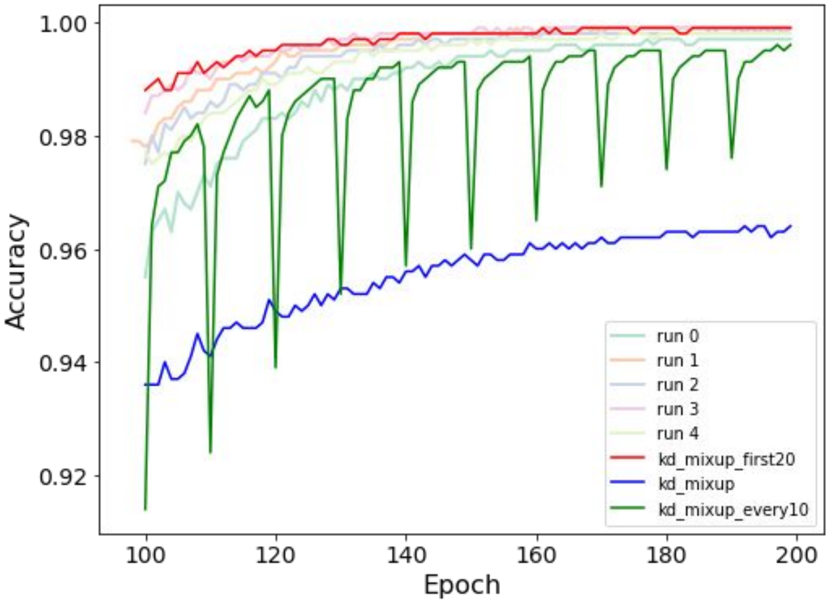
```
Loss =
cross_entropy(student_out, target) +
mse(student_out,teachers_mixup_out)
```

Otherwise use the original loss

```
Loss = cross_entropy(student_out,
target)
```

Knowledge Distillation and MixUp





Knowledge Distillation and MixUp

Baseline	Acc. when epoch=200
Run0	0.997
Run1	0.999
Run2	0.999
Run3	0.999
Run4	0.998
Ours(avg.)	**0.999**

Epochs used to reach acc	ACC= 0.95	ACC= 0.96	ACC= 0.97	ACC= 0.98	ACC= 0.99
Run0	97	101	109	117	135
Run1	82	88	94	102	114
Run2	85	89	95	103	118
Run3	81	84	88	97	107
Run4	87	93	100	108	122
Ours(avg.)	**76**	**82**	**84**	**93**	**105**

Learned Lessons

Learned Lessons So Far

- ❏ Good Harness Design is important

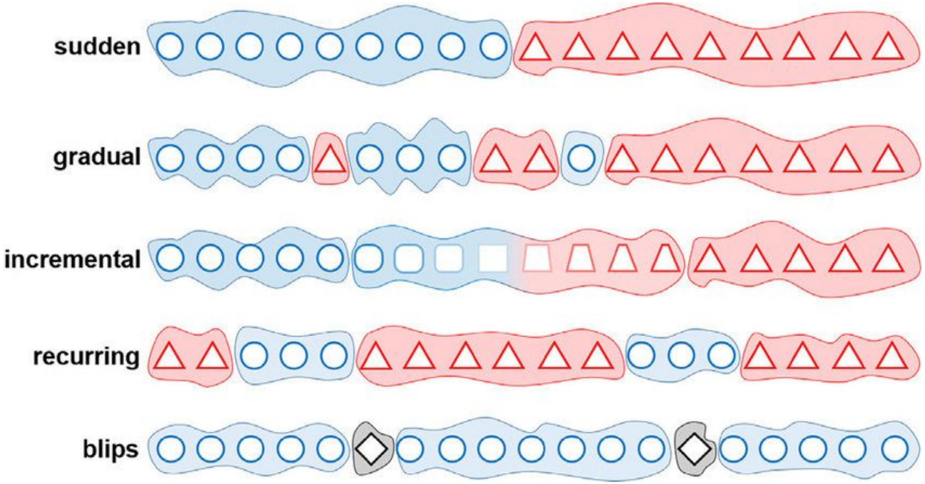
- ❏ Scheduling jobs is helpful

- ❏ Focus on the simple setting rather than the complex setting first

Next Steps:

Short-Term: KD + Mixup

- Distribution drift
  - Extend harness to simulate broad range of distribution drift scenarios
  - Extend metrics to measure “robustness” to drift
- Asymptotic convergence: does the student always necessarily lag behind the teacher?
- Explore variations in implementation: scheduling, weighting, etc.



Krawczyk and Cano (2018)

Longer-Term: Alternate Strategies

Curriculum Learning

- Leverage prior loss calculations
- Expected improvements:
  - Increased training **efficiency**
  - Equivalently, improve the tradeoff between model **accuracy** and training **speedup** (i.e. number of epochs)
- Implementation: simplify and store loss metrics **per-input** over the course of model training

Adversarial Recycling

- Leverage recycled adversarial examples
- Expected improvements:
  - Increased **robustness**
  - Equivalently, **smaller losses in efficiency** when the distribution changes
- Implementation: explicitly calculate and store adversarial examples over the course of the model's lifetime (i.e. **across iterations**)

Longer-Term: Alternate Applications

- Goal: harness and framework that can be easily used for an arbitrary model/dataset training setup
- Build-in further robustness for harness:
  - Check teacher outputs explicitly
  - Explore strategies for KD-switching
- Expand testing beyond CIFAR + ResNet-56