Problem

How can we fastly train video classification models?
Approach

- Development of video-handling condensation
- Changed data loader
  - Can get data from Newton cluster
- Changed model architecture
  - Architecture with video backbone
Distribution Matching

- Distribution matching – directly matching the distributions of the synthesized and the real data
- Idea: to find model parameters that minimize the discrepancy between the generated distribution and the true data distribution
Distribution Matching: Experiment

- CIFAR10, 10 images per class in synthetic dataset with ConvNet
  - Testing accuracy: 49.18%
  - Testing loss: 3.6520

- Paper Results
  - Testing accuracy: 48.9%

- CIFAR10, 50 images per class in synthetic dataset with ConvNet
  - Testing accuracy: 62.81%
  - Testing loss: 1.4082

- Paper Results
  - Testing accuracy: 63.0%
Examples of synthesized images via DM
Utilizes: CIFAR10, ConvNet, 50 images per class
Approach

- Development of video-handling condensation
- Changed data loader
  - Can get data from Newton cluster
- Changed model architecture
  - Architecture with video backbone
Baseline Model: Design

- Condensation with coreset technique: Random
- Dataset: UCF-101
- Architecture: ResNet-18 3D
- Videos per class for smaller dataset: 50
Baseline Model: Design

- UCF-101
  - 13,320 videos
  - 101 action categories:
    - Rope Climbing
    - Apply Eye Makeup
    - etc.
  - Aim: encourage research into action recognition
ResNet 2D versus ResNet 3D

• 2D
  ◦ Typically used for images
  ◦ Dimensions of 3D tensor:
    ▪ Input: $b \times c \times t \times h \times w$
      • $b =$ batches, $c =$ channels, $h =$ height, $w =$ width
    ▪ Output: $b \times nc$
      • $nc =$ number of classes

• 3D
  ◦ Typically used for videos
  ◦ Dimensions of 3D tensor:
    ▪ Input: $b \times c \times t \times h \times w$
      • $t =$ time
    ▪ Output: $b \times nc$
Baseline Model: Progress

- **Successes**
  - Data loader for UCF-101
    - Can get dataset's videos and labels
  - Model architecture initialization
    - e.g., ResNet, ResNext, ShuffleNet
- **Areas for Improvement**
  - Training and testing functions
    - Tensor-model issues
  - Add dataset condensation
Next Steps

- Fix issues and get baseline results for UCF-101
- Finetune learning rate
- Add temporal consistency
- Experiment with model to get results for:
  - More coreset techniques, gradient matching, and distribution matching
  - More architectures
  - More small datasets on Newton (e.g., JRDB)
Thank You

Any Questions?