Model Assertions for Monitoring and Improving ML Models

Daniel Kang*, Deepti Raghavan*, Peter Bailis, Matei Zaharia

DAWN Project, Stanford InfoLab

http://dawn.cs.stanford.edu/

Machine learning is deployed in mission-critical settings with few checks

Tesla's autopilot repeatedly accelerated towards lane dividers

Uber autonomous vehicle involved in fatal crash

- Errors can have life-changing consequences
- No standard way of quality assurance!

Software 1.0 is also deployed in mission-critical settings!

Software powers medical devices, etc.

Important software goes through rigorous engineering / QA process

» Assertions

» Unit tests

» Regression tests

» Fuzzing

» …

Our research:

Can we design QA methods that work across the ML deployment stack?

This talk:

Model assertions

a method for checking outputs of models for both runtime monitoring and improving model quality

Key insight: models can make *systematic* errors

Cars should not flicker in and out of video

Boxes of cars should not highly overlap

(see paper for examples)

We can specify errors in models without knowing root causes or fixes!

THE VERGE

TECH \ TRANSPORTATION \ UBER

Serious safety lapses led to Uber's fatal self-driving crash, new documents suggest

"As the [automated driving system] **changed the classification** of the pedestrian several times—**alternating between vehicle, bicycle, and an other** — the system was unable to correctly predict the path of the detected object," the board's report states.

Model assertions at deployment time

Frame 1

Frame 2

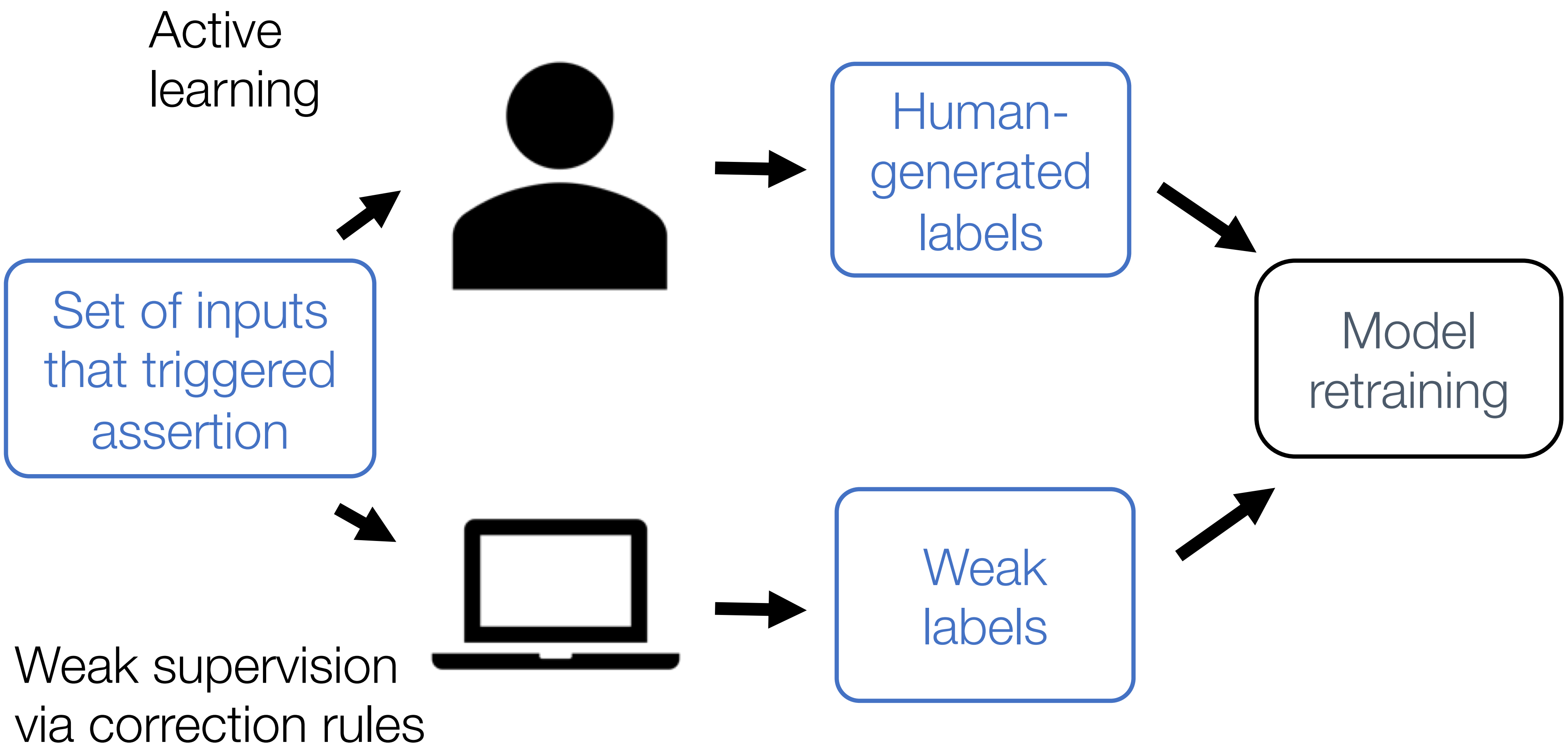
Frame 3


```
assert(cars should not flicker in and out)
```

Runtime monitoring

Corrective action

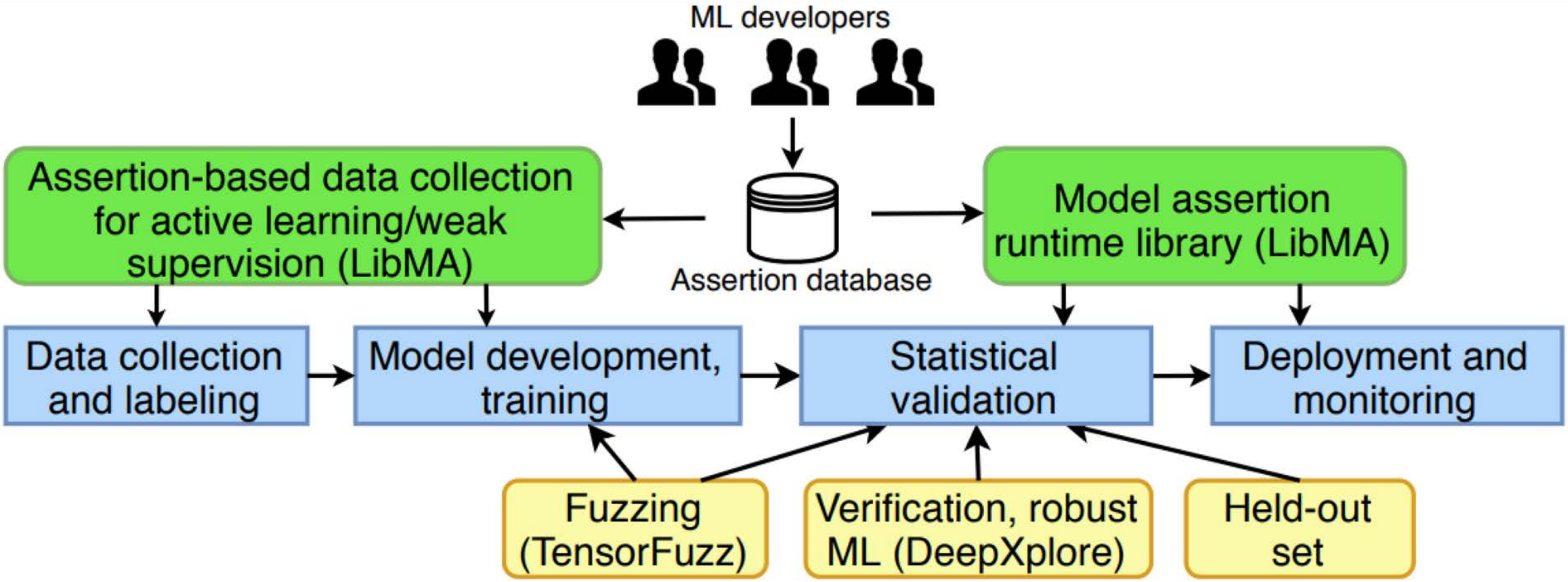
Model assertions at train time



Outline

- Using model assertions
  - Overview
  - For active learning
  - For weak supervision
  - For monitoring
- Model assertions API & examples
- Evaluation of model assertions

Model assertions in context

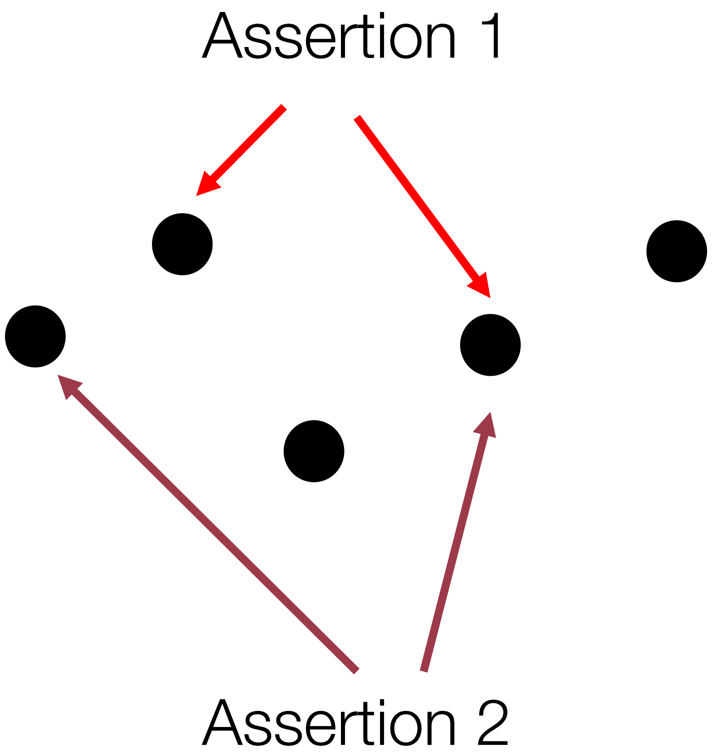


Many users, potentially not the model builders, can collaboratively add assertions

Outline

- Using model assertions
  - Overview
  - For active learning
  - For weak supervision
  - For monitoring
- Model assertions API & examples
- Evaluation of model assertions

How should we select data points to label for active learning?



- Many assertions can flag the same data point
- The same assertion can flag many data points
- Which points should we label?

How should we select data points to label for active learning?

Input: T, B^t, N, R
Output: choice of arms S^t at rounds $1, \ldots, T$

```
for t = 1, ..., T do
    if t = 0 then
        Select data points uniformly at random from
          the d model assertions
    else
        Compute the marginal reduction r_m of the
          number of times model assertion m = 1, ..., d
          triggered;
        for i = 1, ..., B^t do
            Select model assertion m proportional to
              r_m;
            Select x_i that triggers m, sample
              proportional to severity score rank;
            Add x_i to S^t;
        end
    end
end
```

- We designed a bandit algorithm for data selection (BAL)
- Idea: select model assertions with highest reduction in assertions triggered

Outline

- » Using model assertions
  - » Overview
  - » For active learning
  - » For weak supervision
  - » For monitoring
- » Model assertions API & examples
- » Evaluation of model assertions

Correction rules for weak supervision: flickering

Frame 1

Frame 2

Frame 3

Frame two is filled in from surrounding frames

Automatic correction rules: consistency API

Identifier	Time stamp	Attribute 1 (gender)	Attribute 2 (hair color)
1	1	M	Brown
1	2	M	Black
1	4	F	Brown
2	5	M	Grey

Propose ‘M’ as an updated label

Outline

» Using model assertions

» Model assertions API & examples

» Evaluation of model assertions

Specifying model assertions: black-box functions over model inputs and outputs

```
def flickering(
    recent_frames: List[PixelBuf],
    recent_outputs: List[BoundingBox]
) -> Float
```

Model assertion inputs are a history of inputs and predictions

Model assertions output a severity score, where a 0 is an abstension

Predictions from different AV sensors should agree

Assertions can be specified in little code

```
def sensor_agreement(lidar_boxes, camera_boxes):
  failures = 0
  for lidar_box in lidar_boxes:
    if no_overlap(lidar_box, camera_boxes):
      failures += 1
  return failures
```

Specifying model assertions: consistency API

Identifier	Time stamp	Attribute 1 (gender)	Attribute 2 (hair color)
1	1	M	Brown
1	2	M	Black
1	4	F	Brown
2	5	M	Grey

Transitions cannot happen too quickly

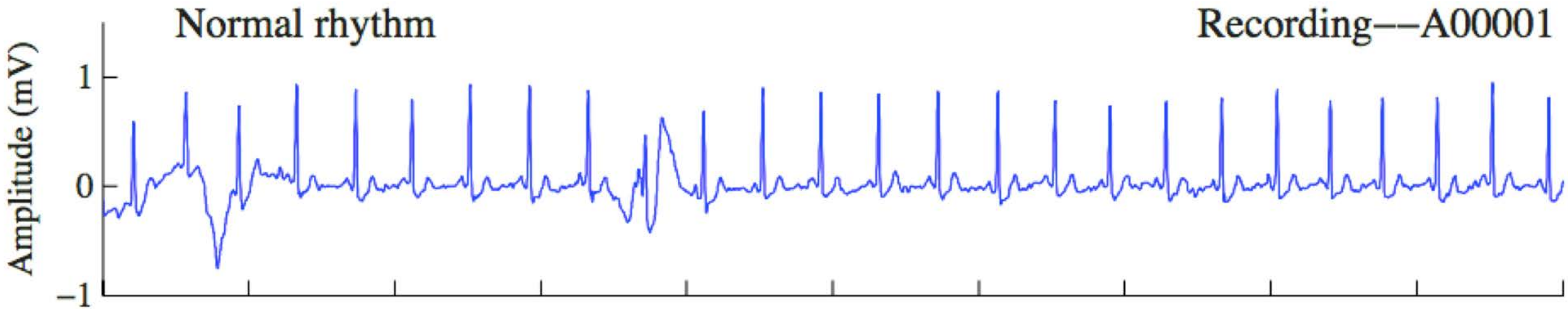
Attributes with the same identifier must agree

Model assertions for TV news analytics

Overlapping boxes in the same scene should agree on attributes

Automatically specified via consistency assertions

Model assertions for ECG readings



Normal AF Normal

Classifications should not change from normal to AF and back within 30 seconds

Automatically specified via consistency assertions

Outline

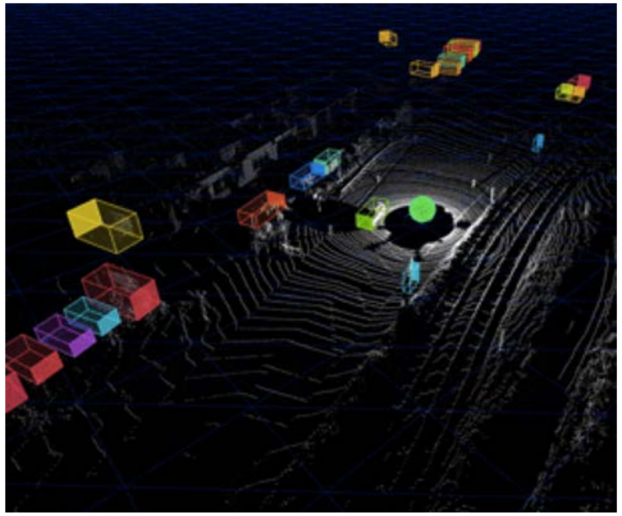
- » Using model assertions
- » Model assertions and examples
- » Evaluation of model assertions
    - » **Evaluation setup**
    - » Evaluating the precision of model assertions (monitoring)
    - » Evaluating the accuracy gains from model assertions (training)

Evaluation setup: datasets and tasks

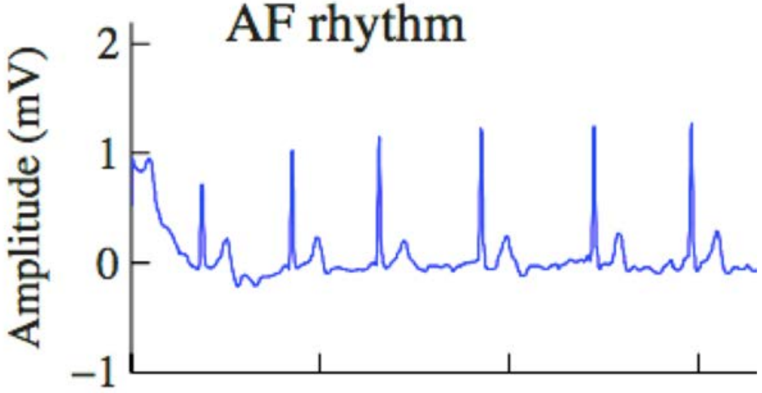
Setting	Task	Model	Assertions
Visual analytics	Object detection	SSD	Flicker, appear, multibox
Autonomous vehicles	Object detection	SSD, VoxelNet	Consistency, multibox
ECG analysis	AF detection	ResNet-34	Consistency
TV news	Identifying TV news hosts	Several	Consistency

Evaluation Setup: Examples

Security camera footage, original SSD



Point cloud data (NuScenes)



Medical time series data

Outline

- Model assertions and examples
- Using model assertions
- Evaluation of model assertions
  - Evaluation setup
  - **Evaluating the precision of model assertions (monitoring)**
  - Evaluating the accuracy gains from model assertions (training)

Evaluating Model Assertion Precision:
Can assertions catch mistakes?

Assertion	True Positive Rate
Flickering	96%
Multibox	100%
Appearing	88%
LIDAR	100%
ECG	100%

Outline

- Model assertions and examples
- Using model assertions
- Evaluation of model assertions
  - Evaluation setup
  - Evaluating the precision of model assertions (monitoring)
  - Evaluating the accuracy gains from model assertions (training)

Evaluating Model Quality after Retraining: Metrics

» Video analytics: box mAP

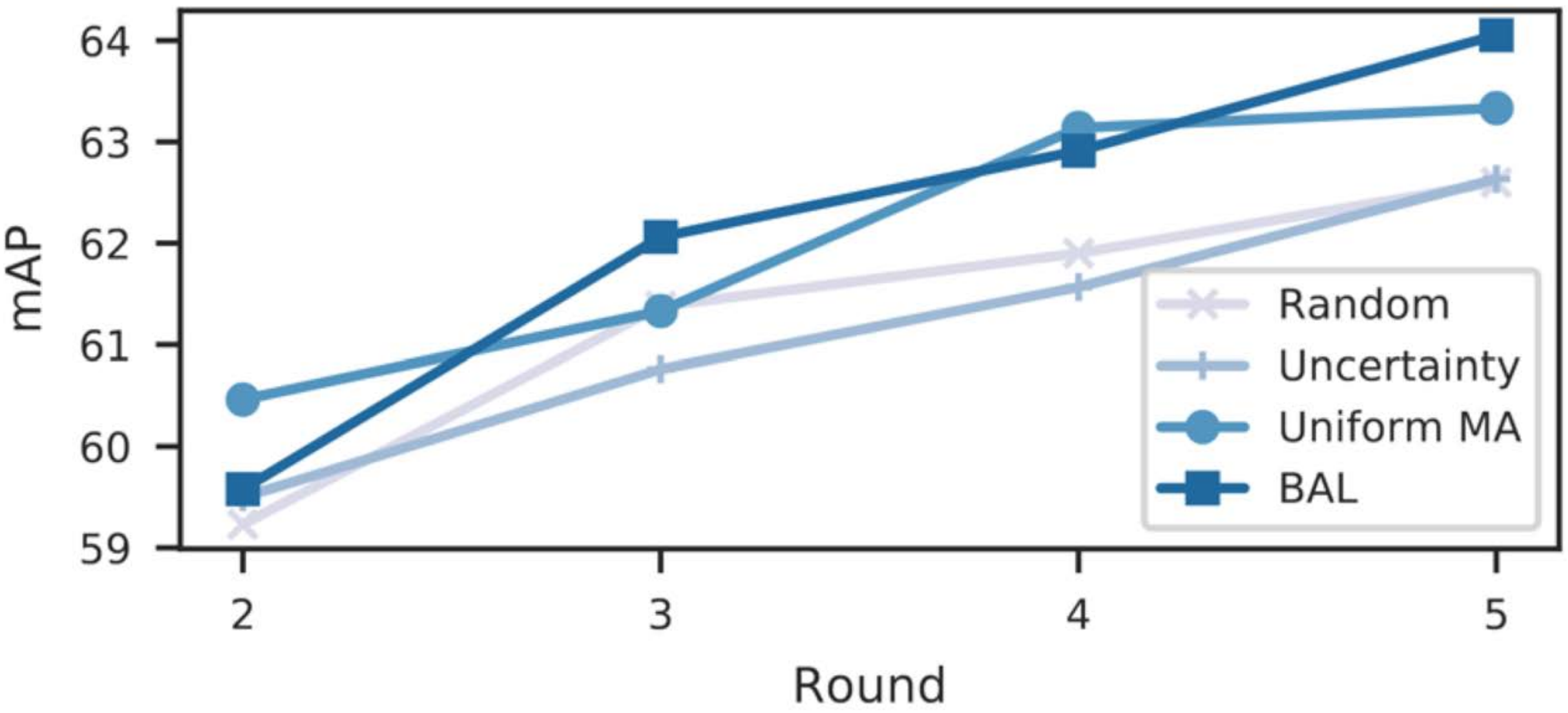
» Autonomous vehicle sensing: box mAP

» AF classification: accuracy

Evaluating Model Quality after Retraining (multiple assertions): Can collecting training data via assertions improve model quality via active learning?

- Finetuned model with 100 examples each round
- 3 assertions to choose frames from:
  - Flickering
  - Multibox
  - Appearing
- Compare against:
  - Random sampling
  - Uncertainty sampling
  - Randomly sampling from assertions

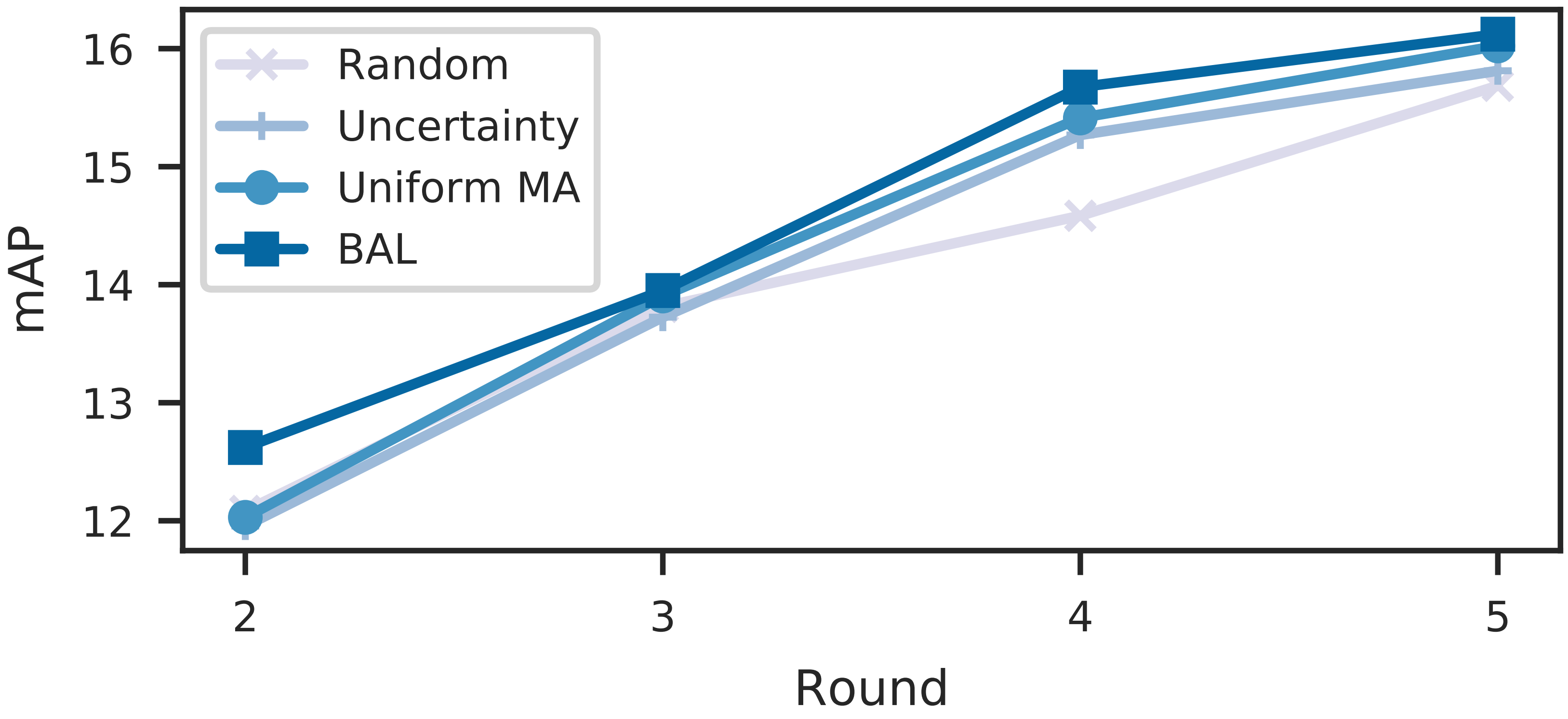
Model assertions can be used for active learning more efficiently than alternatives (video analytics)



Using assertions outperforms uncertainty and random sampling

Our bandit algorithm outperforms uniformly sampling from assertions

Model assertions also outperform on autonomous vehicle datasets (NuScenes)



Using assertions outperforms uncertainty and random sampling

Evaluating Model Quality after Retraining:

Can correction rules improve model quality without human labeling via weak supervision?

Domain	Pretrained	Weakly supervised
Video analytics (mAP)	34.4	49.9
AVs (mAP)	10.6	14.1
ECG (% accuracy)	70.7	72.1

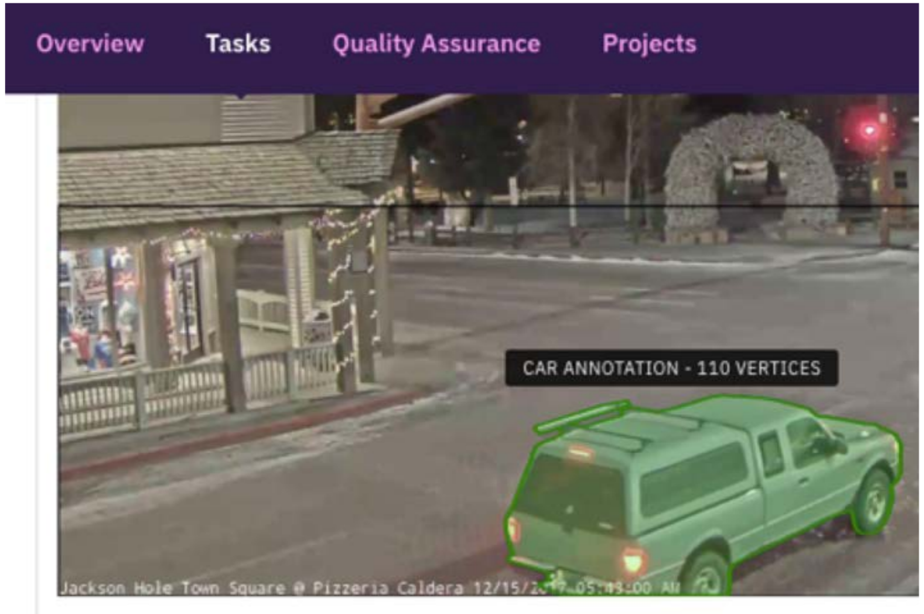
Full experimental details in paper

Further results in paper

» Model assertions can find high confidence errors

» Model assertions for validating human labels (video analytics)

» Active learning results with a single model assertion (ECG)



Incorrect annotation
from Scale AI

Future work

» What is the language to specify model assertions?

» How can we choose thresholds in model assertions automatically?

» How can we apply model assertions to other domains such as text?

Conclusion: Assertions can be Useful in ML!

No standard way of doing quality assurance for ML

- Model assertions can be used for:
  - Monitoring ML at deployment time
  - Improving models at train time
- Preliminary results show significant model improvement

ddkang@stanford.edu