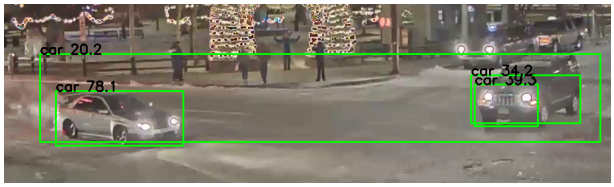
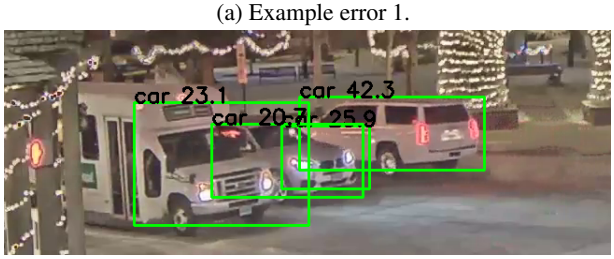
(a) Frame 1 (b) Frame 2

Figure 6. Two example frames from the same scene with an inconsistent attribute (the identity) from the TV news use case.



(a) Example error 1.



(b) Example error 2.

Figure 7. Examples errors when three boxes highly overlap (see `multibox` in Section 5). Best viewed in color.

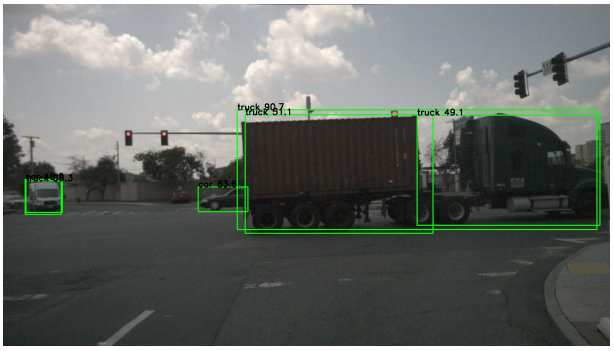
A Examples of Errors Caught by Model Assertions

In this section, we illustrate several errors caught by the model assertions used in our evaluation.

First, we show an example error in the TV news use case in Figure 6. Recall that these assertions were generated with our consistency API (§4). In this example, the identifier is the box's `sceneid` and the attribute is the `identity`.

Second, we show an example error for the visual analytics use case in Figure 7 for the `multibox` assertion. Here, SSD erroneously detects multiple cars when there should be one.

Third, we show two example errors for the AV use case in Figure 8 from the `multibox` and `agree` assertions.



(a) Example error flagged by `multibox`. SSD predicts three trucks when only one should be detected.

(b) Example error flagged by `agree`. SSD misses the car on the right and the LIDAR model predicts the truck on the left to be too large.

Figure 8. Examples of errors that the `multibox` and `agree` assertions catch for the NuScenes dataset. LIDAR model boxes are in pink and SSD boxes are in green. Best viewed in color.

Assertion class	Assertion sub-class	Description	Examples
Consistency	Multi-source	Model outputs from multiple sources should agree	• Verifying human labels (e.g., number of labelers that disagree) • Multiple models (e.g., number of models that disagree)
	Multi-modal	Model outputs from multiple modes of data should agree	• Multiple sensors (e.g., number of disagreements from LIDAR and camera models) • Multiple data sources (e.g., text and images)
	Multi-view	Model outputs from multiple views of the same data should agree	• Video analytics (e.g., results from overlapping views of different cameras should agree) • Medical imaging (e.g., different angles should agree)
Domain knowledge	Physical	Physical constraints on model outputs	• Video analytics (e.g., cars should not flicker) • Earthquake detection (e.g., earthquakes should appear across sensors in physically consistent ways) • Protein-protein interaction (e.g., number of overlapping atoms)
	Unlikely scenario	Scenarios that are unlikely to occur	• Video analytics (e.g., maximum confidence of 3 vehicles that highly overlap), • Text generation (e.g., two of the same word should not appear sequentially)
Perturbation	Insertion	Inserting certain types of data should not modify model outputs	• Visual analytics (e.g., synthetically adding a car to a frame of video should be detected as a car), • LIDAR detection (e.g., similar to visual analytics)
	Similar	Replacing parts of the input with similar data should not modify model outputs	• Sentiment analysis (e.g., classification should not change with synonyms) • Object detection (e.g., painting objects different colors should not change the detection)
	Noise	Adding noise should not modify model outputs	• Image classification (e.g., small Gaussian noise should not affect classification) • Time series (e.g., small Gaussian noise should not affect time series classification)
Input validation	Schema validation	Inputs should conform to a schema	• Boolean features should not have inputs that are not 0 or 1 • All features should be present

Table 6. Example of model assertions. We describe several assertion classes, sub-classes, and concrete instantiations of each class. In parentheses, we describe a potential severity score or an application.

B Classes of Model Assertions

We present a non-exhaustive list of common classes of model assertions in Table 6 and below. Namely, we describe how one might look for assertions in other domains.

Our taxonomization is not exact and several examples will contain features from several classes of model assertions. Prior work on schema validation (Polyzotis et al., 2019; Baylor et al., 2017) and data augmentation (Wang & Perez, 2017; Taylor & Nitschke, 2017) can be cast in the model assertion framework. As these have been studied, we do not focus on these classes of assertions in this work.

Consistency assertions. An important class of model assertions checks the consistency across multiple models or sources of data. The multiple sources of data could be the output of multiple ML models on the same data, multiple sensors, or multiple views of the same data. The output from the various sources should agree and consistency model assertions specify this constraint. These assertions can be generated via our API as described in §4.

Domain knowledge assertions. In many physical domains, domain experts can express physical constraints or unlikely scenarios. As an example of a physical constraint, when predicting how proteins will interact, atoms should not physically overlap. As an example of an unlikely scenario, boxes of the visible part of cars should not highly overlap (Figure 7). In particular, model assertions of unlikely scenarios may not be 100% precise, i.e., will be soft assertions.

Perturbation assertions. Many domains contain input and output pairs that can be perturbed (perhaps jointly) such that the output does not change. These perturbations have been widely studied through the lens of data augmentation (Wang & Perez, 2017; Taylor & Nitschke, 2017) and adversarial examples (Goodfellow et al., 2015; Athalye et al., 2018).

Input validation assertions. Domains that contain schemas for the input data can have model assertions that validate the input data based on the schema (Polyzotis et al., 2019; Baylor et al., 2017). For example, boolean inputs that are encoded with integral values (i.e., 0 or 1) should never be negative. This class of assertions is an instance of preconditions for ML models.

C Hyperparameters

Hyperparameters for active learning experiments. For `night-street`, we used 300,000 frames of one day of video for the training and unlabeled data. We sampled 100 frames per round for five rounds and used 25,000 frames of a different day of video for the test set. Due to the cost of obtaining labels, we ran each trial twice.

For the NuScenes dataset, we used 350 scenes to bootstrap the LIDAR model, 175 scenes for unlabeled/training data for SSD, and 75 scenes for validation (out of the original 850 labeled scenes). We trained for one epoch at a learning rate of 5×10^{-5}. We ran 8 trials.

For the ECG dataset, we train for 5 rounds of active learning with 100 samples per round. We use a learning rate of 0.001 until the loss plateaus, which the original training code did.

Hyperparameters for weak supervision experiments. For `night-street`, we used 1,000 additional frames with 750 frames that triggered `flicker` and 250 random frames with a learning rate of 5×10^{-6} for a total of 6 epochs.

For the NuScenes dataset, we used the same 350 scenes to bootstrap the LIDAR model as in the active learning experiments. We trained with 175 scenes of weakly supervised data for one epoch with a learning rate of 5×10^{-5}.

For the ECG dataset, we use 1,000 weak labels and the same training procedure as in active learning.