Supplementary Material for Trust, but Verify: Cross-Modality Fusion for HD Map Change Detection

John Lambert[1,2] and **James Hays**[1,2]

[1]Argo AI [2]Georgia Institute of Technology

In this Supplementary Material, we provide additional details about our dataset and experiments. In Section (A), we provide an ablation study on the influence of input crop size on model performance. In Section (B), we discuss additional implementation details about our training, data augmentation, and occlusion-based map rendering process. In Section (C), we discuss the paired positive-negative logs we include. In Section (D), we describe our evaluation metric. In Section (E), we provide additional experimental analysis of different models and rendering viewpoints. In Section (F), we provide additional details about how we generate orthoimagery. In Section (G), we offer additional examples from our test set. In Section (H), we give examples of other types of temporary map changes which we do not annotate or evaluate within our dataset. In Section (I) we provide further analysis of the frequency of map changes. In Section (J), we give additional details about our synthetic map perturbation protocol.

In Section (K), we provide a datasheet for the dataset.

Appendix A: Influence of Input Crop Size

In this section, we perform an ablation on input crop size, as discussed in Section 5.1 of the main text. In the main paper, we set our input crop size to 224×224 px for all experiments mentioned therein. In this section, we present an ablation to measure the influence of input crop size. Again, we find the ego-view model is the best-performing model, as measured on its own field of view.

Perhaps surprisingly, we find that an RGB image at 234×234 px resolution ($\sim$ 164K pixel values/image) is sufficient to capture significant detail. In Table 1, we present an ablation where we find that for BEV models, higher resolution (i.e. 468×468 px) does improve mAcc by 2% mAcc, although requiring almost 4x the GPU memory during training and significantly longer training times. However, for ego-view models, a higher crop size is quite detrimental, reducing visibility-based mAcc by around 7%.

Table 1: Controlled evaluation of the influence of input crop size (for ego-view and BEV).

				Modalities			Visibility-based Eval. @ 20m	BEV proximity Eval. @20m			Visibility-based Eval. @20m		
Resolution	Backbone	Arch.	Viewpoint	RGB	Semantics	Map	val mAcc	test mAcc	Is Changed Acc	No Change Acc	test mAcc	Is Changed Acc	No Change Acc
224x224	ResNet-18	Early Fusion	Ego-View	✓	dropout	dropout	0.8384	**0.6850**	**0.63**	0.74	**0.7342**	**0.72**	0.74
448x448	ResNet-18	Early Fusion	Ego-View	✓	dropout	dropout	0.8713	0.6331	0.38	**0.88**	0.6644	0.45	**0.88**
224x224	ResNet-50	Early Fusion	BEV	✓	no	✓	0.9007	0.6543	0.57	0.74			
448x448	ResNet-50	Early Fusion	BEV	✓	no	✓	**0.9072**	0.6749	**0.63**	0.72			

35th Conference on Neural Information Processing Systems (NeurIPS 2021) Track on Datasets and Benchmarks.

Appendix B: Additional Implementation Details

B.1. Training

We train our models for 90 epochs with the Adam [3] optimizer. We use a polynomial learning rate decay strategy, starting at 1×10^{-3}. We use a batch size of 1024 examples. We start with pretrained ImageNet weights for ResNet-18 or ResNet-50 [2].

We train with multiple negative examples per sensor image, which we found to be more beneficial than randomly sampling a single negative example (i.e. a synthetically perturbed map). In other words, we perform multiple types of perturbations for a given scene, and feed them to the network as separate negative examples (not necessarily in the same mini-batch).

B.2. Data Augmentation

We employ a number of data augmentation techniques to improve the generalization of our models and prevent overfitting. Input images are of dimension 2048×1550 for the front-center camera, and 1550×2048 for all other 6 cameras. For the ego-view models, we first take a square crop from the bottom 1550×1550 of an ego-view image. Afterwards, we resize to 234×234, perform a random horiztonal flip with 50% probability, take a random 224×224 crop, divide pixel intensities by 255, and then normalize both sensor and map RGB channels by the ImageNet mean $(\mu_r, \mu_g, \mu_b) = (0.485, 0.456, 0.406)$ and standard deviation $(\sigma_r, \sigma_g, \sigma_b) = (0.229, 0.224, 0.225)$

For BEV models, we resize input images from 2000×2000 px to 234×234 px, perform a random horizontal and/or vertical flip with 50% probability each (independently), choose a random 224×224 crop, and normalize as described above.

We find other traditional data augmentation techniques from the semantic segmentation literature [9], such as applying a random rotation to the input or randomly blurring the input with a small kernel, to be ineffective.

B.3. Occlusion Reasoning

As discussed in Section 5.1 of the main text, we use map occlusion reasoning when generating the input for our ego-view models. Occluded map elements and map elements that have been removed in the real world ("deleted") are both not visible in camera imagery. While the former is an expected everyday occurrence, and the latter is of interest to us, we use occlusion reasoning in order to separate the two phenomena. We generate a dense depth map from sparse LiDAR returns (see Figure 1) and the depth of map entities is compared against the corresponding depth of its projection in the depth map.

(a) RGB Image (b) Interpolated Depth Map

Figure 1: Example of a dense depth map interpolated from sparse LiDAR returns.

B.4. Details about Semantic Label Map Input

As discussed in Section 5.1 of the main text, we use semantic label maps generated from the semantic head of a publicly-available seamseg ResNet-50 panoptic segmentation model [5] [1]. We create 5 binary mask channels from the semantic label map, for the 'road', 'bike-lane', 'marking-crosswalk-zebra', 'lane-marking-general', and 'crosswalk-plain' classes. These are optionally provided as

[1] Available at https://github.com/mapillary/seamseg.

Table 2: We describe the statistics of the map deviation data in our test set, and the types of deviations we observe. We define each BEV frame as a pose where the egovehicle has moved at least 5 meters since the previous pose. Lane geometry changes extend over far more frames than crosswalk changes.

	Data Split	
	Train/Val	Test
Num Images	6,991,006	1,008,134
Avg. number of images per log (@20 Hz)	8,129	7,201
Num LiDAR Sweeps	511,208	74,937
Avg. number of LiDAR sweeps per log (@10 Hz)	594	535
Num. rendered BEV frames (once every 5 meters of trajectory)	25,363	4,945
Num. BEV frames with no changes	25,363	2,159
Num. BEV frames with changes	0	2,786
Num. BEV frames with crosswalk changes only	0	201
Num. BEV frames with lane geometry changes only	0	2,105
Num. BEV frames with both lane geometry and crosswalk changes	0	120

additional channels to the 3 RGB sensor channels and 3 RGB map channels via early fusion. Seamseg's semantic label maps on their own do not capture sufficient granularity for the map change detection task we define, since the Mapillary Vistas public dataset's taxonomy does not differentiate between lane color and or different marking types (e.g. double-solid, solid, dashed-solid), which are of interest to autonomous vehicle operation.

Unsuitability of Per-Pixel Semantic Comparison. Directly comparing rendered map and semantic label maps at a *per-pixel* level is not always useful since our HD map representation does not provide paint annotation for every single dashed longitudinal lane marking, but rather provides a description lane marking pattern, polyline boundary, and other corresponding attributes (See Table 3 of the main text). Thus, we can simulate the pattern of dashed lane markings, but not their exact, pixel-perfect location. As the main text shows, the network can abstract away the per-pixel details to provide more meaningful features.

Appendix C: Data Selection

For a subset of the 'negative' logs in our TbV dataset, we provide a corresponding 'positive' log captured before the change occurred. Example images from pair positive-negative logs are provided in Figure 2. This allows for non-learning based approaches (e.g. based upon comparison of 3d reconstructed world models) for a limited amount of the test set.

Appendix D: Evaluation

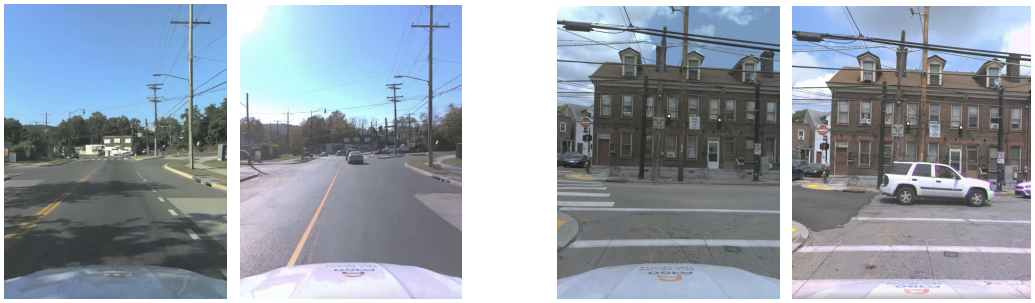
As our primary accuracy metric, we use a mean of class accuracies over two classes. This accounts for both precision and recall. If a confusion matrix is computed with predicted entries on the rows and actual classes as the columns, and normalized by dividing by the sum of each column, 2-class accuracy can be simply calculated as the mean of the diagonal of the confusion matrix.

More formally, let $n_{cl} = 2$ be the number of classes, $\hat{y}_i$ be the prediction for the i'th test example, and y_i be the ground truth label for the i'th test example. We define per-class accuracy (Acc_c) and mean accuracy (mAcc) as:

$$\text{mAcc} = {}^1\!/_{n_{cl}} \sum_{c=0}^{n_{cl}} Acc_c, \quad \text{Acc}_c = \frac{\sum_{i=0}^{N} \mathbb{1}\{\hat{y}_i = y_i\} \cdot \mathbb{1}\{y_i = c\}}{\sum_{i=0}^{N} \mathbb{1}\{y_i = c\}} \tag{1}$$

Appendix E: Additional Experimental Analysis

Advantages of BEV. In principle, the bird's eye view (BEV) representation (orthoimagery) offers two main advantages: a single, dense, accumulated metrically-accurate representation for a single pass through a network, rather than passing in 7 images through 7 separate networks, trained on each frustum, in order to detect changes to the sides and rear of the vehicle. This approach can be costly



(a) Before (WDC) (b) After (WDC) (c) Before (PIT) (d) After (PIT)

(e) Before (MIA) (f) After (MIA) (g) Before (PIT) (h) After (PIT)

(i) Before (PAO) (j) After (PAO) (k) Before (PIT) (l) After (PIT)

(m) Before (PIT) (n) After (PIT) (o) Before (MIA) (p) After (MIA)

(q) Before (WDC) (r) After (WDC) (s) Before (MIA) (t) After (MIA)

Figure 2: For a number of 'negative' logs, our TbV dataset includes corresponding logs captured before the map change was implemented, such that we obtain "before and after" imagery.

at inference time given the number of camera frustums required to achieve a panoramic view with traditional cameras. Second, the BEV is generally free of distortion, compared to the ego-view. The ego-view can be seen as "spoiling" the map data's metric nature.

Advantages of Ego-view. However, an ego-view perspective also presents clear advantages over the BEV. Rendering data in the BEV can be seen as "spoiling" the sensor data's texture. Importantly, there is less distraction and less overall content to reason about in the egoview. Therefore, the ego-view task is arguably easier than the BEV task, needing only to detect changes in a 85° f.o.v. instead of 360° f.o.v.

Analysis of Map-Only Baseline. The map-only baseline performs quite poorly when predicting real-world lane geometry changes, slightly over random chance (2% or 3% over random chance in the ego-view and 7% over random change in the BEV). While the map-only stream may seem doomed to fail without access to real-world sensor information, we observe that a certain number of map changes exist to bring the real world into compliance with certain priors, which are already encapsulated in the map. For example, we find that upgrading a 4-way intersection from a single crosswalk to 4 crosswalks, or from a single crosswalk to 0 crosswalks (after repaving) is a common map change, which would agree with priors. Indeed, our experimental results suggest that the map-only baseline, which is completely blind to the real-world, can occasionally succeed at predicting real-world crosswalk changes by learning powerful priors. Inspection via Guided GradCAM demonstrates that the map-only models attends to asymmetric paint patterns along the left and right boundaries of a road, or asymmetric lane subdivisions along two sides of a road; modifications to such map asymmetry which are common real-world map updates.

Analysis of Sensor-Only Baseline. The sensor-only model (see Table 6 of the main paper) sees randomly perturbed labels, with only "positive" training data, and therefore is not a meaningful baseline.

Appendix F: Orthoimagery Generation Implementation Details

In this section, we provide additional details about the orthoimagery generation process described in Sections 4.3 and 5.1 of the main text. In order to create a metrically-accurate sensor data representation that is free of perspective distortion, we generate orthoimagery using ray-casting. Orthoimagery from LiDAR suffers from extreme sparsity, leading to an impoverished representation. To generate dense panoramic orthoimagery, we use a set of high-definition camera sensors with a panoramic field of view, mounted onboard an autonomous vehicle. We generate the BEV representation (i.e. orthoimagery) by ray-casting image pixels to a ground surface triangle mesh. Our ground height maps exploit LiDAR offline, and in this way our ego-view method incorporates the strengths of LiDAR.

CUDA Ray-Casting Routine. We tesselate quads from a ground surface mesh with 1 meter resolution to triangles; rays are cast to triangles up to 25 m away from the egovehicle. For acceleration, we cull triangles outside of the left and right cutting planes of each camera's view frustum. We implement the Moller-Trombore ray-triangle intersection routine [4] in CUDA.

Density. Ray-casting yields a vastly more dense set of image rays than LiDAR, on the order of 2 orders of magnitude greater density; for a 1550×2048 image, one can obtain ~ 3.17 million rays per image, and across 7 camera frustums, this translates to over 22.19 million rays with available RGB values per second. With 20 fps imagery per camera frustum, this amounts to 440 million rays per second. Most conventional 10 Hz LiDAR sensors can provide little more than 100k returns per sweep, and thus at most 1 million rays per second.

Aggregation. In order to prevent holes in the orthoimagery in the area underneath the egovehicle, we aggregate pixels in ring buffer of length 10 sweeps, and wait 10 sweeps before starting rendering. Future sensor data is not used to render the sensor data representation. We use linear interpolation to account for sparsity at range.

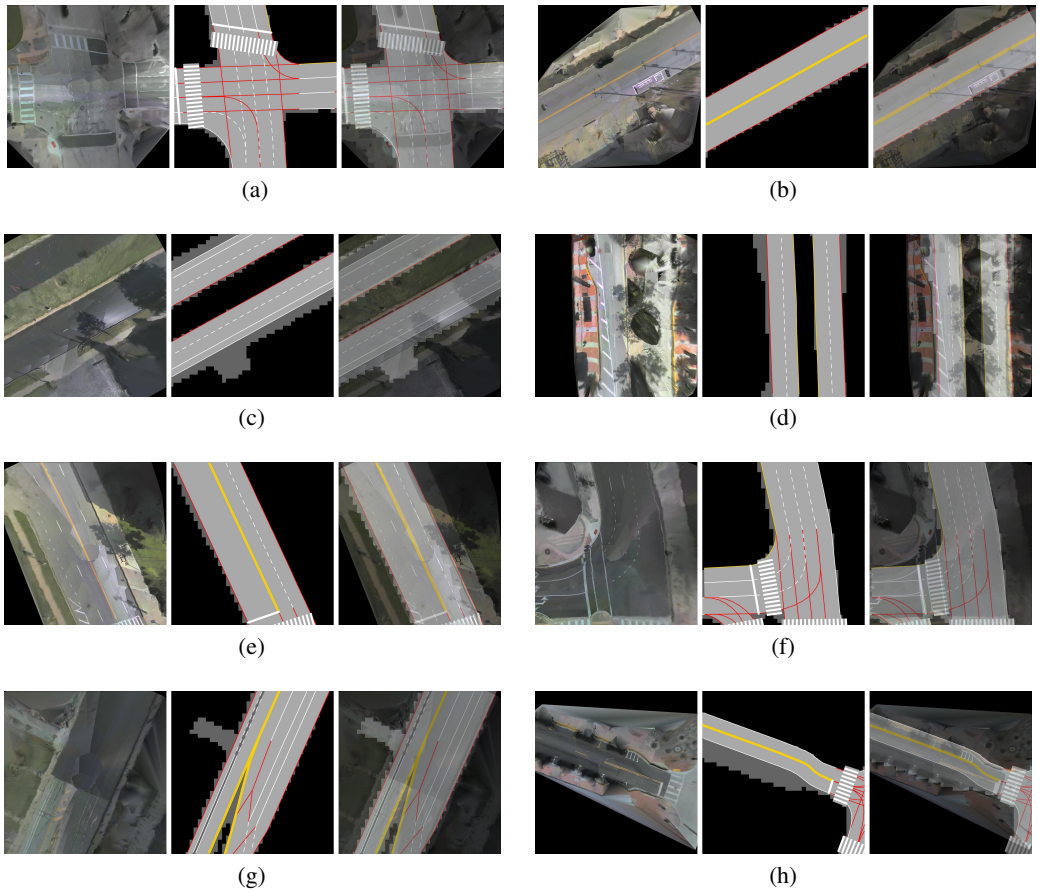


Figure 3: Examples from the test split of our TbV dataset. Left to right: BEV sensor representation, onboard map representation, blended map and sensor representations. Rows, from top to bottom: inserted crosswalks (top row), and painted lane geometry changes (bottom 7 rows).

Comparison with IPM. While Inverse Perspective Mapping (IPM) is the dominant approach in the literature, it is inaccurate as it cannot account for ground surface variation. Geiger [1] model the image-to-ground plane mapping as a homography (IPM) and mosaics together monocular images, but requires scenes with an approximately-planar ground surface. Zhang *et al.*[8] generate orthophoto ground imagery using fisheye cameras and IPM. Rapo [6] explored the use of dashboard-mounted cell phones without access to LiDAR or known calibration, instead relying upon SfM, optical flow, and vanishing point estimation for online calibration and also use IPM for pixel-to-world correspondence.

Appendix G: Additional Examples from Test Set

In Figure 3, we show additional examples from our test set, as seen from a bird's eye view.

Appendix H: Map Changes from Construction

In Figure 4, we show examples of object-centric map changes inside our TbV dataset, which we do not annotate and are not the focus of our work.

Appendix I: Additional Analysis of Map Change Frequency

In Section 3.1 and Table 2 of the main paper, we present an analysis of map change frequency. In this section, we provide additional analysis, an extended table, and derivations of our estimates. Map

(a) Traffic Cones

(b) Jersey Barriers

(c) Type III Traffic Barricades

(d) Fallen Trees

(e) Construction Signs

(f) Traffic Barrels / Drums

(g) Arrowboard Trailers

Figure 4: Scenes with temporary object-related map changes collected in our fleet data. Such scenes are not the focus of our work; rather, we believe such changes should be addressed by onboard object recognition systems.

Table 3: Entities included in our HD map representation.

HD MAP ENTITY	CORRESPONDING ATTRIBUTES
PEDESTRIAN CROSSINGS	2 EDGES ORIENTED ALONG ITS PRINCIPAL AXIS
LANES	BOUNDARIES: 3D LEFT AND RIGHT POLYLINES
	COLOR: YELLOW, WHITE, OR IMPLICIT
	BOUNDARY MARKING TYPE
	CONNECTIVITY
	LANE TYPE: BIKE OR VEHICLE LANE
	IN INTERSECTION: TRUE OR FALSE
DRIVEABLE AREA	POLYGONS
GROUND SURFACE HEIGHT	FLOATING POINT HEIGHT VALUES AT 30 CENTIMETER RESOLUTION

changes occur at random as part of a stochastic process. While some changes are coordinated at a city-administration level, it is still difficult to predict to a specific date or time when construction crews will complete changes. As discussed in the main text, we reason about square spatial areas of size 30 m $\times$ 30 m, which we refer to as tiles, which cover 900 m^2 each.

Derivation: Probability of an Encounter We consider the probability of entering a spatial area that has undergone a crosswalk or lane geometry within it. In other words, it is the probability of encountering a changed area, and thus we name it p_{eca}. In order to estimate the probability of encountering a changed area, rather than computing the ratio $\left(\frac{\text{num change-discovery miles}}{\text{num fleet miles}}\right)$, we compute the ratio $\left(\frac{\text{num. tiles where change is observed}}{\text{num. tiles entered by fleet}}\right)$. We do not require that the autonomous vehicle directly drove over the changed tile, as an observed change can very well still affect driving behavior. We model the probability as a Bernoulli(p) r.v., with $p \approx 0.005517\%$ across the more than 5 North American cities we analyze. A visit would occur once per every 18,124 times a vehicle enters such areas.

While the change percentage may seem inconsequential, one must consider that drivers in the United States are estimated to drive 3.225 trillion miles per year, according to the U.S. Department of Transportation [7]. If one were to consider our rate of change equal to the rate of change of any stretch of road within the United States, this would amount to an *upper bound* of 9B encounters of

Table 4: Across six particular cities, we analyze the probability of change for a $30m \times 30m$ spatial area. Since we can likely only catch changes for spatial areas that are somewhat frequently visited, we require that an area is visited by fleet at least $n = 5$ times. We provide $n = 1$ as well as a lower bound.

City Name	≥ 5 visits by fleet		≥ 1 visit by fleet	
	Probability of Change per Tile	Up to T tiles in a thousand will change in 5 months	Probability of Change per Tile	Up to T tiles in a thousand will change in 5 months
Pittsburgh	0.0068	7	0.0052	5
Detroit	0.0056	6	0.0049	5
Washington, D.C.	0.0046	5	0.0037	4
Miami	0.0038	4	0.0027	3
Austin	0.0009	0.9	0.0006	0.6
Palo Alto	0.0007	0.7	0.0006	0.6

spatial areas with changed lane geometry or crosswalks, per year:

$$\frac{3.225 \cdot 10^{12}\text{miles}}{1 \text{ year}} \cdot \frac{1609 \text{ m}}{1 \text{ mile}} \cdot \frac{1 \text{ tile}}{30 \text{ m}} \cdot \frac{5.517 \cdot 10^{-5} \text{ changes}}{1 \text{ tile}} \approx 9.5B \tag{2}$$

This derivation assumes that all roads (including highways) are changed as often as urban roads (a generous estimate).

Derivation: Probability per Spatial Area We next estimate the probability of each unique tile in a city seeing a crosswalk or lane geometry change, which we also model as a Bernoulli(p) random variable, with p estimated as:

$$p = \frac{\text{\# unique changed tiles in city}}{\text{\# unique tiles in city visited at least n times by fleet}} \tag{3}$$

where the numerator and denominator are both measured over k months.

In Table 4, we analyze the probability of change for a $30m \times 30m$ spatial area across six particular cities. Since we can likely only catch changes for spatial areas that are somewhat frequently visited, we require that an area is visited by fleet at least $n = 5$ times over $k = 5$ months.

Appendix J: Synthetic Map Perturbation Technique

Table 5: Training dataset statistics and types of synthetic changes generated from 800 logs. Not all scenes can support all synthetic change types. For example, in order to delete a crosswalk from a local map, a crosswalk must be present of local vicinity of the egovehicle.

Change Category	Description of Change	Quantity of Examples
BEV Sensor Images	N/A	25,393
No change	None	25,263
Lane Geometry Changes	Delete Lane Marking	19,870
	Change Lane Marking Color	25,098
	Change Lane Boundary Dash-Solid	19,875
	Add Bike Lane	21,529
Crosswalk Changes	Delete Crosswalk	9,627
	Insert Crosswalk	23,166

In Section 4.2, Table 4, and Figure 4 of the main text, we enumerate a number of hand-designed priors we use to generate realistic-appearing synthetic maps. In this section, we provide detailed descriptions of the generation process.

J.1. Priors on the Crosswalk Perturbation Procedure

Our main observations from studying mapped data are that crosswalks are generally located near intersections, are orthogonal to lane segment tangents, and have little to no area overlap with other crosswalks. Accordingly, we first sample a random lane segment which will be spanned by the generated, synthetic crosswalk. We perform this random sampling from a biased but normalized probability distribution; lane segments within intersections achieve 4.5x the weight of non-intersection lane segments. In order to determine the orientation of the synthesized crosswalk's principal axis, we compute the normal to the centerline of the sampled lane segment at a randomly sampled waypoint. This waypoint is sampled from 50 waypoints that we interpolate along the centerline. We ensure that the sampled waypoint is not within the outermost 1/8 of pixels along any border of the rendered map image (i.e. within 15 m according to ℓ_∞ norm from the egovehicle). This measure is to allow some perturbation of the random crop for data augmentation, without losing visibility of the changed entity.

Next, in order to determine how many total lane segments the crosswalk must cross in order to span the entire road, we must determine the road extent. We approximate it as the union of all nearby lane segment polygons. The line representing the principal axis of the crosswalk may intersect with this road polygon in more than two locations, since it is often non-convex. We choose the shortest possible length segment that spans the road polygon to be valid, and thus find the closest two intersections to the sampled centerline waypoint. We randomly sample a crosswalk width w in meters from a normal distribution $w \sim \mathcal{N}(\mu = 3.5, \sigma = 1)$, but clip to the range $w \in [2, 4]$ meters afterwards, in accordance to our empirical observations of the real-world distribution.

If the rendered synthetic crosswalk has overlap with any other real crosswalk above a threshold of IoU $= 0.05$, we continue to sample until we succeed. The crosswalk is rendered as a rectangle, bounded between two long edges both extending along the principal axis of the crosswalk. We use alternating parallel strips of white and gray to color the object. Crosswalks are deleted by simply not rendering them in the rasterized image.

J.2. Lane Geometry Perturbation Procedure

Our main observations from studying real-world map changes are that lane changes generally occur over a chain of lane segments, with combined length often over tens or hundreds of meters, although at times the combined length is far shorter. Accordingly, we use the directed lane graph to sample random connected sequences of lane segments, respecting valid successors. We then manipulate either the left or the right boundary only (not both) of this lane sequence.

Our general procedure is to start this sequence at a random lane well-within the field of view of the BEV image. As before, we ensure that the sampled marking is not entirely contained within the outermost 1/8 of pixels along any border of the rendered map image (i.e. within 15 m according to ℓ_∞ norm from the egovehicle).

When deleting lane boundaries, we sample only painted yellow or white lane boundary markings. When changing the color or structure of lane boundaries, we sample lane boundary markings of any color (including those that are implicit). When adding a bike lane, we sample a sequence of 5 lane segments. For marking deletion and changes to lane marking color and structure, we sample a sequence of length 3.

We render these boundaries as colored polylines; we use red for implicit boundaries, and yellow and white for lane markings of their respective color. Lane boundary markings are deleted by simply not rendering them in the rasterized image.

Bike lanes generally represent the rightmost lane in the United States. Accordingly, we synthesize a valid location for a new bike lane by iterating through the lane graph until there is no right neighbor; by dividing this rightmost lane into half, we can create two half-width lanes in place of one. We use solid white lines to represent their boundaries.

Appendix K: Datasheet for TbV

In this appendix, we answer the questions laid out in *Datasheets for Datasets* by Gebru *et al.*[2].

[2] https://arxiv.org/abs/1803.09010

Motivation

For what purpose was the dataset created? Was there a specific task in mind? Was there a specific gap that needed to be filled?

TbV was created to allow the community to improve the state of the art in machine learning tasks related to mapping, that are vital for self-driving.

To our knowledge, no prior datasets has ever been publicly released for HD map change detection. It is also one of the largest sensor datasets ever released, paired with HD maps, allowing for new exploration of the synergies between the sensor data and map data.

Who created this dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organization)?

The TbV dataset was created by researchers at Argo AI.

Who funded the creation of the dataset?

The creation of this dataset was funded by Argo AI.

Composition

What do the instances that comprise the dataset represent (e.g., documents, photos, people, countries)? Are there multiple types of instances (e.g., movies, users, and ratings; people and interactions between them; nodes and edges)?

The core instances for TbV are brief "scenarios" or "logs" of 30-90 seconds that represent a continuous observation of a scene around a self-driving vehicle.

Each scenario has an HD map representing lane boundaries, crosswalks, drivable area, etc. They also contain a raster map of ground height at 0.3 meter resolution.

How many instances are there in total (of each type, if appropriate)?

The TbV Dataset has 1000 30-90 second scenarios.

Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set? If the dataset is a sample, then what is the larger set? Is the sample representative of the larger set (e.g., geographic coverage)? If so, please describe how this representativeness was validated/verified. If it is not representative of the larger set, please describe why not (e.g., to cover a more diverse range of instances, because instances were withheld or unavailable).

The scenarios in the dataset are a sample of the set of observations made by a fleet of self-driving vehicles. The data is not uniformly sampled.

The "negative" instances in the dataset were chosen to include specific examples where an HD map has become out-of-date, due to real-world changes.

The "positive" instances in the dataset were chosen to include interesting behavior (e.g. cars making unexpected maneuvers), to contain interesting weather (e.g. rain and snow), and to be geographically diverse (spanning 6 cities – Pittsburgh, Detroit, Austin, Palo Alto, Miami, and Washington D.C.).

What data does each instance consist of? "Raw" data (e.g., unprocessed text or images) or features? In either case, please provide a description.

Each TbV scenario is 30-90 seconds in duration. Each scenario has 20 fps video from 7 ring cameras, 20 fps video from two forward-facing stereo cameras, and 10 Hz LiDAR returns from two out-of-phase 32-beam LiDARs. The ring cameras are synchronized to fire when either LiDAR sweeps through their field of view. Each scenario contains vehicle pose over time and calibration data to relate the various sensors.

The HD map associated with each scenario contains polylines describing lanes, crosswalks, and drivable area. Lanes form a graph with predecessors and successors, e.g. a lane that splits can have two successors. Lanes have precisely localized lane boundaries that include paint type (e.g. double solid yellow). Drivable area, also described by a polygon, is the area where it is possible (but not necessarily legal) to drive without damaging the vehicle. It includes areas such as road shoulders.

Is there a label or target associated with each instance?

Yes. For the logs found in the train and validation splits, an up-to-date HD map serves as a label, as these are "positive" logs, where the map and sensor data are in agreement.

For the logs found in the test split, 3d coordinates of polygons or polylines are manually annotated for areas where the map has changed, for lane paint and crosswalks, specifically.

In addition, the LiDAR depth estimates can act as ground truth for monocular depth estimation. The vehicle pose data could be considered ground truth labels for visual odometry. The evolving point cloud itself can be considered ground truth for point cloud forecasting.

Is any information missing from individual instances? If so, please provide a description, explaining why this information is missing (e.g., because it was unavailable). This does not include intentionally removed information, but might include, e.g., redacted text.

No. To our knowledge, all instances should be complete.

Are relationships between individual instances made explicit (e.g., users' movie ratings, social network links)? If so, please describe how these relationships are made explicit.

Each instance of the datasets (a vehicle "log") is disjoint. Each carries their own HD map for the region around their scenario. These HD maps may overlap spatially, though. For example, they may be captured at the same intersection, but separated in time by several months. If a user of the dataset wanted to recover the spatial relationship between scenarios, they could do so through our development kit.

Are there recommended data splits (e.g., training, development/validation, testing)? If so, please provide a description of these splits, explaining the rationale behind them.

We define splits of the TbV dataset. The train, validation, and test set include 720/80/200 logs each.

Are there any errors, sources of noise, or redundancies in the dataset?

Every sensor used in the dataset – ring cameras and lidar – has noise associated with it. Pixel intensities, lidar intensities, and lidar point 3D locations all have noise. Lidar points are also quantized to float16 which leads to roughly a centimeter of quantization error. 6 degree of freedom vehicle pose also has noise. The calibration specifying the relationship between sensors can be imperfect.

The HD map for each scenario can contain noise, both in terms of lane boundary locations and precise ground height.

Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g., websites, tweets, other datasets)? If it links to or relies on external resources, a) are there guarantees that they will exist, and remain constant, over time; b) are there official archival versions of the complete dataset (i.e., including the external resources as they existed at the time the dataset was created); c) are there any restrictions (e.g., licenses, fees) associated with any of the external resources that might apply to a future user? Please provide descriptions of all external resources and any restrictions associated with them, as well as links or other access points, as appropriate.

The data itself is self-hosted, and we will maintain public links to all previous versions of the dataset in case of updates.

Does the dataset contain data that might be considered confidential (e.g., data that is protected by legal privilege or by doctor-patient confidentiality, data that includes the content of individuals non-public communications)?

No.

Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety?

No.

Does the dataset relate to people?

Yes, the dataset contains images and behaviors of thousands of people on public streets.

Does the dataset identify any subpopulations (e.g., by age, gender)?

No.

Is it possible to identify individuals (i.e., one or more natural persons), either directly or indirectly (i.e., in combination with other data) from the dataset? If so, please describe how.

We do not believe so. Image data has been anonymized via blurring. Faces and license plates are obfuscated by replacing their corresponding bounding box with a 5×5 grid, where each grid cell is the average color of the original pixels in that grid cell. The anonymization is done manually. For example, a person sitting on their front porch 10 meters from the road would have their face obscured.

Does the dataset contain data that might be considered sensitive in any way (e.g., data that reveals racial or ethnic origins, sexual orientations, religious beliefs, political opinions or union memberships, or locations; financial or health data; biometric or genetic data; forms of government identification, such as social security numbers; criminal history)?

N/A.

Collection Process

How was the data associated with each instance acquired? Was the data directly observable (e.g., raw text, movie ratings), reported by subjects (e.g., survey responses), or indirectly inferred/derived from other data (e.g., part-of-speech tags, model-based guesses for age or language)? If data was reported by subjects or indirectly inferred/derived from other data, was the data validated/verified? If so, please describe how.

The sensor data was directly acquired by a fleet of autonomous vehicles.

Over what timeframe was the data collected? Does this timeframe match the creation timeframe of the data associated with the instances (e.g., recent crawl of old news articles)? If not, please describe the timeframe in which the data associated with the instances was created.

The data was collected from May 2020 to March 2021.

What mechanisms or procedures were used to collect the data (e.g., hardware apparatus or sensor, manual human curation, software program, software API)? How were these mechanisms or procedures validated?

The Trust but Verify (TbV) data comes from Argo 'Z1' fleet vehicles. These vehicles use Velodyne lidars and traditional RGB cameras. All sensors are calibrated by Argo. HD maps are created and validated through a combination of computational tools and human annotations. Map change labels are created through human annotation.

If the dataset is a sample from a larger set, what was the sampling strategy (e.g., deterministic, probabilistic with specific sampling probabilities)?

The dataset scenarios were chosen from a larger set through manual review. The test set scenarios were selected to illustrate unambiguous map changes.

Who was involved in the data collection process (e.g., students, crowdworkers, contractors) and how were they compensated (e.g., how much were crowdworkers paid)?

Argo employees and Argo interns curated the data. Data collection and data annotation was done by Argo employees. Crowdworkers were not used.

Were any ethical review processes conducted (e.g., by an institutional review board)?

No.

Does the dataset relate to people?

Yes.

Did you collect the data from the individuals in question directly, or obtain it via third parties or other sources (e.g., websites)?

The data is collected from vehicles on public roads, not from a third party.

Were the individuals in question notified about the data collection? If so, please describe (or show with screenshots or other information) how notice was provided, and provide a link or other access point to, or otherwise reproduce, the exact language of the notification itself.

No, but the data collection was not hidden. The Argo fleet vehicles are well-marked and have obvious cameras and LiDAR sensors. The vehicles only capture data from public roads.

Did the individuals in question consent to the collection and use of their data? If so, please describe (or show with screenshots or other information) how consent was requested and provided, and provide a link or other access point to, or otherwise reproduce, the exact language to which the individuals consented.

No. People in the dataset were in public settings and their appearance has been anonymized. Drivers, pedestrians, and vulnerable road users are an intrinsic part of driving on public roads, so it is important that datasets contain people so that the community can develop more accurate perception systems.

If consent was obtained, were the consenting individuals provided with a mechanism to revoke their consent in the future or for certain uses? If so, please provide a description, as well as a link or other access point to the mechanism (if appropriate).

N/A.

Has an analysis of the potential impact of the dataset and its use on data subjects (e.g., a data protection impact analysis) been conducted? If so, please provide a description of this analysis, including the outcomes, as well as a link or other access point to any supporting documentation.

No.

Preprocessing/cleaning/labeling

Was any preprocessing/cleaning/labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)? If so, please provide a description. If not, you may skip the remainder of the questions in this section.

Yes. Images are reduced from their full resolution, and are JPEG compressed. 3D point locations are quantized to float16. Ground height maps are quantized to 0.3 meter resolution from their full resolution. HD map polygon vertex locations are quantized to 0.01 meter resolution.

Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data (e.g., to support unanticipated future uses)? If so, please provide a link or other access point to the "raw" data.

Yes, but such data is not public.

Is the software used to preprocess/clean/label the instances available?

No.

Uses

Has the dataset been used for any tasks already?

Yes, this manuscript benchmarks a novel HD map change detection method on the TbV dataset.

Is there a repository that links to any or all papers or systems that use the dataset?

Yes, at https://github.com/johnwlambert/tbv. We plan to add a leaderboard for the HD map change detection task using the test split of the TbV dataset.

What (other) tasks could the dataset be used for?

The TbV dataset could be used for research on visual odometry, lane detection, synthetic HD map generation, map automation, self-supervised learning, scene flow, and point cloud forecasting.

Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses? For example, is there anything that a future user might need to know to avoid uses that could result in unfair treatment of individuals or groups (e.g., stereotyping, quality of service issues) or other undesirable harms (e.g., financial harms, legal risks) If so, please provide a description. Is there anything a future user could do to mitigate these undesirable harms?

No.

Are there tasks for which the dataset should not be used?

The dataset should not be used for tasks which depend on faithful appearance of faces or license plates since that data has been obfuscated. For example, running a face detector to try and estimate how often pedestrians use crosswalks will not result in meaningful data.

Distribution

Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created?

Yes, the dataset is hosted on https://www.argoverse.org/. Our dataset requires no user registration for access. The dataset's metadata page will include structured metadata.

In addition to long term hosting on Argoverse.org, the Creative Commons license enables rehosting by any repository. The authors will ensure that the dataset is accessible.

How will the dataset will be distributed (e.g., tarball on website, API, GitHub) Does the dataset have a digital object identifier (DOI)?

The TbV dataset is distributed as a series of tar.gz files. The files are broken up to make the process more robust to interruption (e.g. a single 1 TB file failing after 3 days would be frustrating) and to allow easier file manipulation (an end user might not have 1 TB free on a single drive, and if they do, they might not be able to decompress the entire file at once).

The dataset can be read with the Argoverse API. See https://github.com/argoai/argoverse-api for details on usage.

When will the dataset be distributed?

The data is currently available for download, at the time of NeurIPS 2021.

Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)? If so, please describe this license and/or ToU, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms or ToU, as well as any fees associated with these restrictions.

Yes, the dataset is released under the same Creative Commons license as Argoverse 1.0 (CC BY-NC-SA 4.0). The authors are responsible for the contents of the dataset and are responsible for any possible violation of rights.

Have any third parties imposed IP-based or other restrictions on the data associated with the instances?

No.

Do any export controls or other regulatory restrictions apply to the dataset or to individual instances?

No.

Maintenance

Who will be supporting/hosting/maintaining the dataset?

Argo AI.

How can the owner/curator/manager of the dataset be contacted (e.g., email address)?

The TbV team will respond through the Github page https://github.com/johnwlambert/tbv/issues (where training code and pre-trained models have been made available).

For privacy concerns, contact information may be found here: https://www.argoverse.org/about.html#privacy.

Is there an erratum?

No.

Will the dataset be updated (e.g., to correct labeling errors, add new instances, delete instances)? If so, please describe how often, by whom, and how updates will be communicated to users (e.g., mailing list, GitHub)?

It is possible that the TbV 1.0 Dataset will be updated to correct errors. Updates will be communicated on Github and through a mailing list we will create.

If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances (e.g., were individuals in question told that their data would be retained for a fixed period of time and then deleted)? If so, please describe these limits and explain how they will be enforced.

No.

Will older versions of the dataset continue to be supported/hosted/maintained? If so, please describe how. If not, please describe how its obsolescence will be communicated to users.

Yes. If we ever deprecate TbV 1.0, we will continue to host it, although we will declare it "deprecated."

If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so? If so, please provide a description. Will these contributions be validated/verified? If so, please describe how. If not, why not? Is there a process for communicating/distributing these contributions to other users? If so, please provide a description.

Yes. The Creative Commons license we use for TbV ensures that the community can do the same thing without needing Argo's permission.

We do not have a mechanism for these contributions/additions to be incorporated back into the 'base' TbV Dataset. Our preference would generally be to keep the 'base' dataset as is, and to give credit to noteworthy additions by linking to them.

Environmental Impact Statement
Amount of Compute Used: We estimate 5000 CPU hours and 3000 GPU hours for all of the data extraction, preparation and experiments.

References

[1] Andreas Geiger. Monocular road mosaicing for urban environments. In *2009 IEEE Intelligent Vehicles Symposium*, pages 140–145. IEEE, 2009. 6

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 2

[3] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[4] Tomas Möller and Ben Trumbore. Fast, minimum storage ray-triangle intersection. *Journal of graphics tools*, 2(1):21–28, 1997. 5

[5] Lorenzo Porzi, Samuel Rota Bulò, Aleksander Colovic, and Peter Kontschieder. Seamless scene segmentation. In *CVPR*, June 2019. 2

[6] Lauri Rapo. Generating road orthoimagery using a smartphone. Master's thesis, Lappeenranta University of Technology, 2018. 6

[7] U.S. Department of Transportation. Strong economy has americans driving more than ever before. Press Release, March 2019. https://www.fhwa.dot.gov/pressroom/fhwa1905.cfm. 7

[8] H. Zhang, M. Yang, C. Wang, X. Weng, and L. Ye. Lane-level orthophoto map generation using multiple onboard cameras. In *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, pages 855–860, 2014. 6

[9] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, July 2017. 2