ArgoTweak: Towards Self-Updating HD Maps through Structured Priors

Supplementary Material

Appendix A. Bijective mapping

This appendix provides additional details and insights into the **ArgoTweak Dataset**, extending the main paper's discussion on its design and application. As introduced in Sec. 4, the dataset is motivated by the need for **systematic, high-fidelity change annotations** in HD map updates. Current datasets used for prior-aided mapping lack structured change annotations, limiting model refinement and evaluation. To address this, we introduce the concept of **bijective change mapping**, a framework that ensures a **one-to-one correspondence** between high-level structural updates and their corresponding atomic changes at the element level.

A.1. Key definitions

As detailed in Sec. 4.1, our framework operates on different map levels. Below, we provide detailed descriptions with examples:

- **Atomic change** a_i: The smallest indivisible edit applied to individual map element (e.g., attribute modifications, vertex adjustments). These edits involve exactly one map element, without influencing the neighboring structures.
- **Structural updates** y_i: Large-scale updates that affect multiple elements, such as adding lanes, modifying intersections, or adjusting entire road layouts. These modifications are explainable through one or multiple atomic changes on one or multiple map elements.
- **Set of atomic changes** x_i: A specific permutationally invariant set of atomic changes, which explains a structural update through applying one or several element-wise atomic edits.
- **Macro-modification** $\hat{y}_i$: Large-scale modifications that form a subset of $\hat{\mathbf{Y}}$, whose combinations approximate $\mathbf{Y}$. These modifications affect shape, appearance, function of map parts, as well as their underlying lane graph and the total lane number in a local patch.
- **Bijective change mapping** f: A novel framework that establishes a direct, one-to-one mapping between structural updates and atomic changes, ensuring consistency and traceability in prior-aided mapping.

A.2. Desiderata for explanation of modifications through atomic change categories

The goal is to ensure a bijective mapping between structural updates on the HD map and their explanation through atomic changes. We define the desiderata for this mapping in Tab. 4.

A.3. Map element properties

The elementary units of our approach are lane segments and pedestrian crossings. Notably, in our model backbone [15], their representation is unified by treating pedestrian crossings as lane segments oriented perpendicular to the driving direction. A lane segment consists of a left border, a right border, and a set of properties. These properties are detailed in Tab. 5, following and complementing the Argoverse 2 map format [28].

A.4. Motivation for bijectivity

From a map annotation perspective, we want to retrieve the richest possible set of change annotations, in the sense that we want to avoid insertions and deletions where possible. This is because an element replacement, *i.e*. deleting an existing map element, and inserting a new one from scratch, "wipes out" potential correspondences between the prior and the ground-truth map. This could limit the ability to leverage structured priors effectively.

In Fig. 6, we illustrate such an example. The structural modification on the shown map patch describes a bike lane turning into an additional vehicle lane. In the upper modification path, we first change the lane type from "BIKE" to "VEHICLE". Next, we modify the geometry of the segments involved. Finally, we alter the lane line marking type for one of the original vehicle lanes. With this approach (which is the one supported by our bijective mapping framework), we maintain correspondence between elements in the prior and their updated versions in the ground truth map. Contrarily, in the lower modification path, we completely remove the bike lane first, and insert a new lane, although the underlying road graph is unchanged. Hence, this is prohibited by our framework.

A.5. The Right-Handside-Rule

Since atomic changes define which edits to apply to a map element, but not necessarily their global location and direction, we standardize insertions and deletions to begin in the driving direction right in cases where it is not immediately clear which element is the inserted one. An example motivating this so-called Right-Handside-Rule can be seen in Fig. 7: The top, central and bottom modification path lead to the same final map state. By applying our bijective mapping, we can discard the bottom path, because deleting the existing structures and replacing them is not justified given the underlying road graph: In both prior and ground-truth-map, the road graph includes at least one lane travelling to the right. However, both central and top path are compati-

ID	Statement	Requirement	Benefit
A_1	Uniqueness and distinctiveness	Ensure that each atomic change produces a unique, interpretable outcome in the map. No two types of changes should lead to the same map state. Every modification should have a distinct representation.	Prevents ambiguity in training map updating algorithms that rely on clear distinctions between map states.
A_2	Non-redundancy	Avoid overlapping or redundant categories that could confuse the classification process or lead to interpretive flexibility. Each category should capture a unique aspect of the modification, avoiding overlap with other categories.	Reduces confusion for both human operators and automated systems interpreting the changes through streamlined categories.
B	Relevance to functionality	Reflect changes that significantly impact how the map would be used by autonomous systems, particularly for navigation and safety. Categories should capture changes that affect traffic flow, lane usage, directional changes, or functionality (e.g., converting a vehicle lane to a bike lane).	Ensures that the map's usability and safety are prioritized, with critical modifications being flagged and interpreted appropriately.
C	Hierarchical change tracking	Facilitate tracking of both high-level and detailed modifications, allowing for a hierarchical understanding of changes. Changes should be categorized at both coarse and fine levels. For example, a high-level category might indicate "structural change" while a subcategory specifies "lane deletion."	Enhances interpretability for complex modifications and allows for selective filtering of changes based on granularity.
D	Modular change tracking	Maintain accuracy and clarity even in complex scenarios, such as multi-lane adjustments or simultaneous functional and geometric changes, by a modular approach to complex changes. The framework must handle composite changes without ambiguity.	Supports classification of multi-layered modifications, preventing misinterpretation when an element undergoes several modifications.

Table 4. Mapping requirements and benefits.

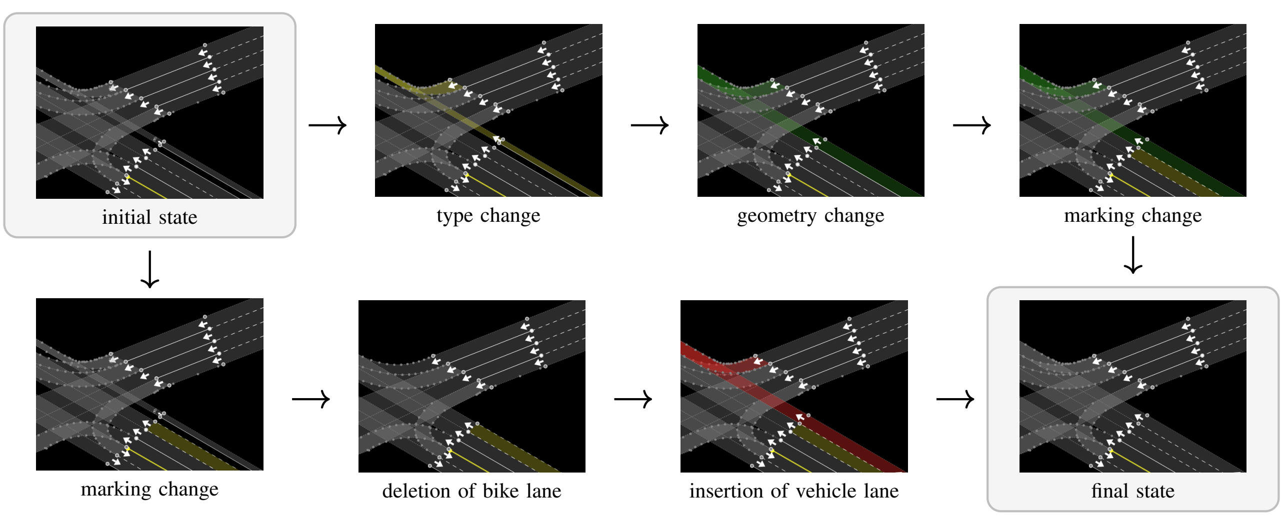


Figure 6. Motivation for bijectivity (Appendix A.4). Green elements have been geometry-edited, red elements are insertions and yellow elements underwent type or marking-related changes.

Property	ls	pc	Description
id	✓	✓	A unique identifier for this lane segment or pedestrian crossing.
is_intersection	✓		True if the lane segment is part of an intersection.
lane_type	✓		Specifies the type of lane (vehicle, bike, bus).
left_lane_boundary	✓	✓	List of points defining the left boundary of the lane.
right_lane_boundary	✓	✓	List of points defining the right boundary of the lane.
centerline	✓	✓	List of points representing the geometric center of the lane or pedestrian crossing.
left_lane_mark_type	✓		Type of marking on the left lane segment boundary with mark type solid/dashed/double-solid/double-dashed/dash-solid/solid-dash/none/unknown and mark color white/yellow/blue/non-visible.
right_lane_mark_type	✓		Type of marking on the right lane segment boundary with mark type solid/dashed/double-solid/double-dashed/dash-solid/solid-dash/none/unknown and mark color white/yellow/blue/non-visible.
successors	✓		Array of IDs for lane segments that follow the segment.
predecessors	✓		Array of IDs for lane segments that precede the segment.
right_neighbor_id	✓		ID of the lane immediately to the right, if it exists.
left_neighbor_id	✓		ID of the lane immediately to the left, if it exists.
is_modified	✓	✓	Indicates whether the segment has been modified compared to the base map.
change_hist	✓	✓	A list of changes applied to the lane segment or pedestrian crossing over time.

Table 5. Lane segment (ls) and pedestrian crossing (pc) object definitions.

ble with our framework. To introduce a uniform map annotation strategy, we apply the Right-Handside-Rule, leaving the top path as the only solution compatible with our framework.

A.6. Function-preserving changes in intersections

We note that within intersections, the road graph is often altered, leading to a large number of deletion+insertion pairs. To prevent overly complex annotations within intersections, we allow element re-routing as a special type of geometry modification. The condition for this special annotation category is that the topological function of the element defined on the road-graph remains unchanged (*e.g.*, turning left from the outermost lane, continuing straight from the center lane).

The concept of these function-preserving road-graph changes is best illustrated through the example in Fig. 8, which presents three different scenarios and how they are annotated in our framework while allowing function-preserving changes in intersections.

In Fig. 8, (1), a new right-turn lane is inserted, changing the total number of lanes in the map. This case is classified as an insertion, as the function of the existing lanes remains unchanged, but the number of lanes in the global road increased, which is accounted for within our framework. Note that the insertion is not constrained by the Right-Handside Rule, as it is immediately clear which is the inserted element, while all existing elements remain unchanged.

In Fig. 8, (2), the right-turn lane is no longer accessible from the outermost lane, but re-routed to the inner lane. While the number of lanes remains unchanged, this re-routing fundamentally alters the functional aspect of the road. Here, the modification is represented by a replacement, *i.e.*, an insertion+deletion pair, as the lane connectivity is altered in a way that impacts traffic behavior.

In Fig. 8, (3), the right-turn option is still removed from the outer lane, but an additional change occurs: the number of lanes in the driving direction is reduced from two to one. In this case, the lane graph structure changes, while the highlighted lane segment in question maintains its original topological function on the new road graph: It allows for right-turns from the right-most lane. This scenario demonstrates a function-preserving road-graph change, where the structure is adjusted but the intended driving options remain clear. Hence, we annotate the segment in question with a geometry change. In our dataset, we mark these elements as

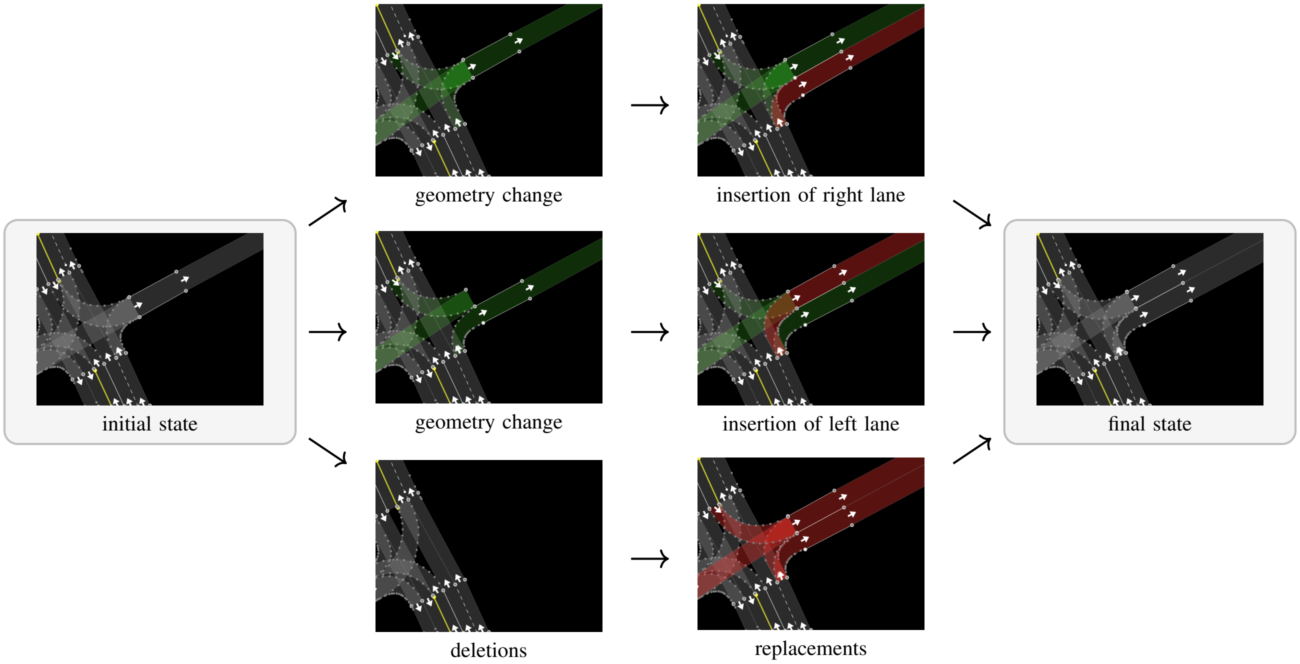


Figure 7. In the above figure, green elements have been geometry-edited, whereas red elements are insertions. The Right-Handside-Rule in Appendix A.5 defines the top path as the only viable option, ruling out the central path. The bottom path is prohibited by our bijective mapping framework, because the underlying road graph does not justify element replacements.

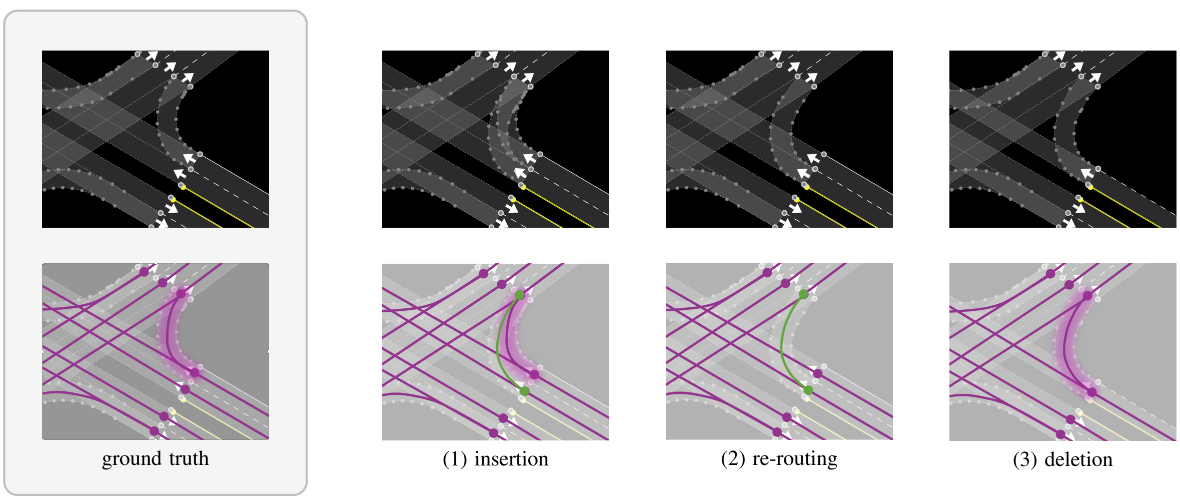


Figure 8. Illustration of different road-graph altering changes. The highlighted pink lane segment defines a right-turn from the outmost lane. Green connections indicate inserted element. A detailed discussion can be found in Appendix A.6.

special geometry edits, allowing for easily converting them back into insertion-deletion pairs if needed.

A.7. Annotations beyond atomic changes

While atomic changes remain indivisible by definition, we introduce additional levels of granularity beyond the five categories of atomic changes, to suffice desideratum C in Tab. 4. This hierarchy should be understood as a refinement in detail rather than a contradiction to their atomic nature. Building upon the lane segment properties outlined in Tab. 5, we structure hierarchical trees that further categorize atomic change categories (Fig. 9). These finer-grained annotations were not utilized in the main paper but are included as part of the dataset release to further enhance interpretability and usability.

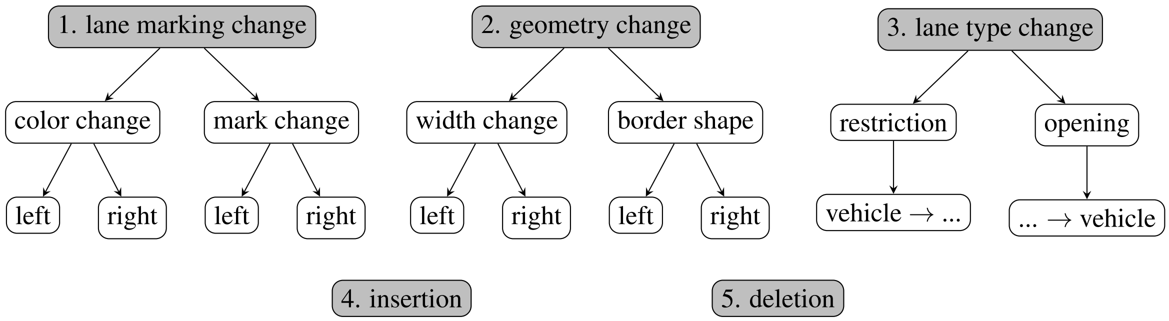


Figure 9. Hierarchical annotation granularity within atomic changes. Notably, insertions and deletions cannot be categorized further, as they do not have any correspondences between prior and ground-truth map.

Category		train			val			test		
		global	frame	element	global	frame	element	global	frame	element
total		697	77670	1368387	102	11287	212695	111	10078	189331
of which changed		697	43536	240849	102	5545	32555	104	5879	50459
ls	geometry	336	18027	68266	50	2043	7773	63	3491	21686
ls	mark	411	19789	42387	59	2359	5361	78	4243	14352
ls	insertion	325	13842	49909	58	2318	8054	37	1862	5354
ls	deletion	212	8052	16171	37	1213	2419	27	1227	3121
ls	topology	149	5497	6767	25	635	829	36	1488	2527
ls	type	23	1539	4276	7	258	1035	6	416	1172
pc	geometry	169	5298	8591	25	823	1044	4	141	266
pc	insertion	324	9945	17382	43	1258	2228	16	511	638
pc	deletion	483	15541	27100	80	2238	3812	20	790	1343

Table 6. Comparison of global, frame-wise, and element-wise change annotations across train, validation, and test sets. The change-class wise annotations are divided by lane segments (ls) and pedestrian crossings (pc).

Appendix B. Dataset statistics

In Tab. 6 we provide the details on annotations in ArgoTweak on global, frame and element-level. By global level, we mean the complete HD map for a specific driving sequence, whereas a frame indicates a $50 \times 50\text{m}^2$ map patch around the local vehicle pose. We include the information for all atomic change categories, while in the experiments, type information has not been used. Topology denotes special geometric changes within intersections that maintain function on the road graph (see Appendix A.6 for details).

Appendix C. Rule-based prior generation

C.1. Pedestrian crossing perturbation procedure

The implementation in our script largely follows the methodology described in [12], with some notable deviations. Below, we summarize the similarities:

- **Sampling of lane segments:** The probability distribution is biased in favor of intersection lane segments, with a 4.5x higher weight, ensuring pedestrian crossings are more likely placed near intersections.
- **Waypoint sampling and orientation:** We interpolate the centerline of the sampled lane and select a random point, using the normal at that point to define the principal axis of the pedestrian crossing.
- **Road extent determination:** We compute the road polygon extent and identify the shortest valid span as [12] suggests.
- **Width sampling:** A width value is sampled from a normal distribution $w \sim \mathcal{N}(3.5, 1)$ and clipped to $[2, 4]$ meters.
- **Overlap avoidance:** We ensure that no significant intersection with existing pedestrian crossings occurs, maintaining an IoU below 0.05.
- **Rendering of pedestrian crossings:** The pedestrian crossing is rendered as a buffered rectangle.

Despite the similarities, some deviations from the described methodology exist:

- **Sampling iteration limit:** Instead of indefinitely sampling until success, our implementation enforces a maximum of 20 iterations for the global map, preventing infinite loops.

- **Width validation:** While the width is normally sampled and clipped, an additional height condition $(h > 2m)$ is enforced, which is not explicitly mentioned in the prior work.
- **Random crop constraints:** The original procedure ensures sampled waypoints avoid the outermost $\frac{1}{8}$ of the image; our script does not explicitly enforce this constraint, because this is not necessarily the case in real-world scenarios. Instead, we enforce that the pedestrian crossing intersects with a buffered area of 15 m around the ego vehicle's trajectory.

C.2. Lane geometry perturbation procedure

We implement lane modifications in accordance with the prior methodology, with some adaptations. Below, we outline the similarities and deviations.

- **Altering lane markings:** We modify lane boundary markings through transitions between solid and dashed, as well as between visible and non-visible markings.
- **Modifying lane boundaries over multiple segments:** We iterate through connected lane segments, modifying three consecutive segments for marking changes and five for bike lanes.
- **Adding bike lanes:** We identify the rightmost lane and divide it in half, introducing a new bike lane with solid white boundaries, as described in the original methodology.
- **Deleting or modifying lane boundaries:** We selectively delete or alter painted lane boundary markings while ensuring that implicit boundaries remain unchanged.

We report the following deviations:

- **Ensuring perturbations stay within the field of view:** The original procedure enforces a constraint to avoid perturbations near the outermost $\frac{1}{8}$ of the rendered image, which we do not explicitly enforce. Instead, we again enforce that the modified segment intersects with a buffered area of 15 m around the ego vehicle's trajectory.
- **Change frequency**: We insert a maximum number of 2 bike lanes per global map, and attempt to change four three-segment sequences of lane border markings.

Appendix D. Additional results

In Tab. 7, we provide the evaluation of our model on the test split of ArgoTweak with primary and secondary change assessment heads active. This complements Tab. 3 in the main paper while providing the AP_c values for lane segments and pedestrian crossings objects separately.

In Fig. 10, we present an analysis of the sim-to-real gap based on mAP_c. While the reduction in the sim-to-real gap is clearly observable in this figure, the effects of map generation and change detection are intertwined, making interpretation more complex. The task is inherently more challenging for mAP_c than for mAcc_c, as the model must accurately assess the change status while simultaneously capturing the correct geometric representation. For mAP_c calculation, we use all elements, while for mAcc_c, we threshold at a confidence score of 0.5.

In Fig. 11, we provide more qualitative examples.

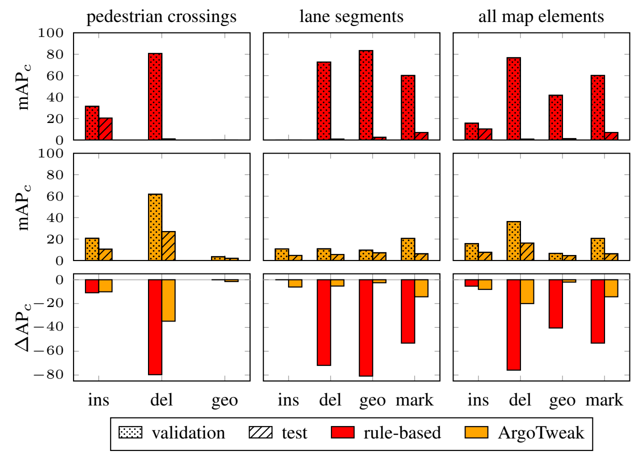
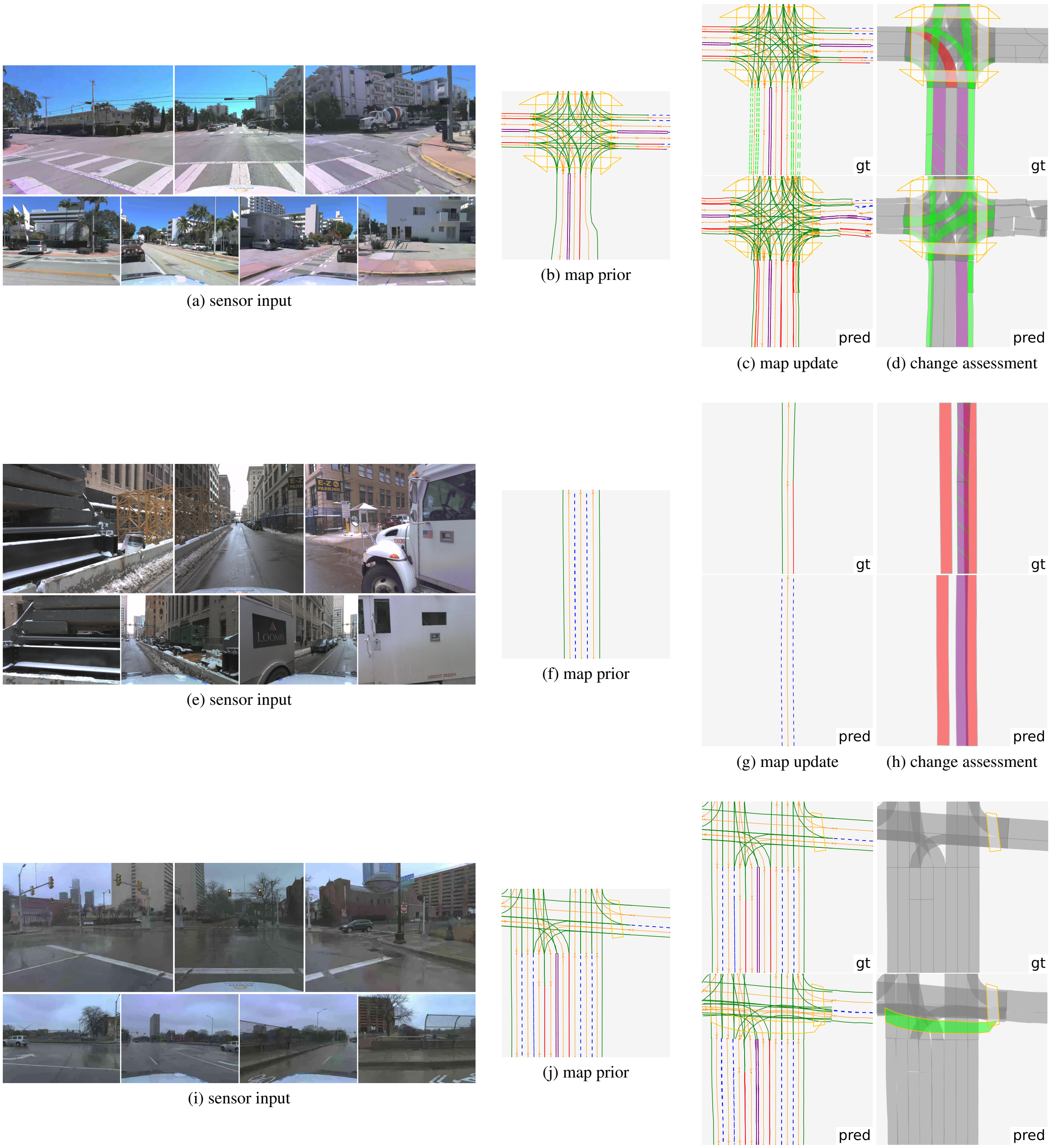


Figure 10. Sim2real gap computed on mAP_c.

all categories	AP_c^{ls}	AP_c^{pc}	mAP
total	75.4	82.2	78.8
change	9.0	19.0	14.0
no change	74.5	82.5	78.5
insertion	4.7	10.6	7.6
deletion	5.6	27.0	16.3
geometry	7.2	2.0	4.6
mark	6.3	–	6.3

Table 7. Performance of our model trained on ArgoTweak and evaluated on the ArgoTweak test set.



(a) sensor input (b) map prior (c) map update (d) change assessment

(e) sensor input (f) map prior (g) map update (h) change assessment

(i) sensor input (j) map prior (k) map update (l) change assessment

Figure 11. Qualitative examples of our ArgoTweak-trained model. For change assessment, purple denotes lane marking changes, light green insertions, red deletions, dark green geometry changes. Striped elements indicate multiple changes per segment. In the first example, the model correctly detects the newly added bike lanes. In the second example, the road shape is correctly updated, but the appearance is not. In the third example, we observe how the model correctly classifies all lane segments as unchanged and reproduces them with high accuracy. However, the white stopline in front of the vehicle is mistaken for a pedestrian crossing insertion.