Language-Driven Semantic Change Detection in Urban Maps via Multi-Modal Deep Learning

Huaze Liu[1,†], Zihao Gao[1,†], and Adyasha Mohanty[2]

[1]Harvey Mudd College

ABSTRACT

High-integrity maps are essential for safe autonomous navigation in dynamic urban environments, where frequent changes and sensor limitations present significant challenges. This paper introduces a novel, deep-learning-driven framework for continuous map uncertainty monitoring and semantic change detection. Our approach leverages data-driven feature extraction from both vision and LiDAR modalities. A key innovation is the integration of zero-shot semantic segmentation using large pre-trained vision-language models, which provides interpretable, language-driven explanations for detected map inconsistencies. The framework dynamically tracks map consistency using Kullback-Leibler divergence metrics, enabling proactive real-time alerts when deviations occur. By jointly assessing structural and semantic integrity, our approach provides a robust and interpretable mechanism for maintaining high-integrity maps in urban autonomous systems.

INTRODUCTION

Accurate and up-to-date maps are fundamental to the safe and efficient operation of autonomous navigation systems, particularly in dynamic urban environments. Autonomous vehicles (AVs), robots, and other intelligent systems rely on these maps for precise localization, robust trajectory planning, and effective obstacle avoidance. However, system reliability is directly tied to the quality and currency of the underlying map representation. Map discrepancies or outdated information can lead to localization drift, incorrect path planning, and potentially unsafe autonomous decisions.

Map generation typically involves fusing data from various sensors such as Global Navigation Satellite Systems (GNSS), LiDAR, and cameras. Each sensor modality inherently introduces noise and biases, which can result in misaligned features and spatial inaccuracies in the map ((Diehl & Bertram, 2023)). Urban settings pose a unique challenge due to their inherent dynamism, characterized by frequent structural and environmental changes like roadworks, new constructions, and temporary obstructions. These transient conditions are often not captured in pre-built maps, leading to critical divergences between the digital map and the real-world environment ((Gu et al., 2023)). Furthermore, high-rise buildings in urban canyons can severely degrade GNSS signals due to multipath and non-line-of-sight (NLOS) errors, directly impacting both map construction and localization accuracy ((Harithas & Krishna, 2023)).

Despite the availability of multi-modal sensor data, current map generation models often employ heuristic-based methods, Gaussian processes, or neural network approximations to infer missing data. While practical, these approaches can introduce systematic biases, especially in high-density urban areas ((Zou & Sester, 2023)). A significant limitation of many commercial high-definition (HD) maps is their infrequent update cycles, which fail to reflect real-time environmental conditions, rendering them unreliable for navigation in rapidly changing scenarios((Pairet & Lahjanian, 2023)). These challenges highlight a critical need for robust mechanisms to monitor map uncertainty and detect changes in real-time.

Research in map uncertainty quantification has largely focused on static assessments rather than dynamic, real-time estimation. GNSS-based methods analyze signal integrity but often ignore semantic inconsistencies. Other techniques utilize Gaussian Mixture Models (GMM) and Gaussian Processes (GPs) to quantify uncertainty in LiDAR-based maps; however, these methods are computationally intensive and cannot dynamically track uncertainty evolution over time. Simultaneous Localization and Mapping (SLAM)-based approaches, like UrbanFly, improve localization in challenging urban environments using visual-inertial odometry (Harithas & Krishna, 2023). However, SLAM methods typically do not track semantic consistency across different time frames, limiting their effectiveness for long-term map uncertainty monitoring or explicit change detection. Approaches that predict occupancy maps using deep generative networks can infer missing regions probabilistically (Katyal & Hager, 2023) but often lack structured uncertainty tracking and do not integrate zero-shot semantic reasoning, which restricts their ability to provide interpretable explanations for detected inconsistencies. Similarly, online HD map estimation techniques can refine maps dynamically, but their primary focus is on trajectory prediction rather than assessing the uncertainty of the underlying map representation (Gu et al., 2023).

Beyond uncertainty quantification, map change detection is another crucial area. Traditional methods often rely on direct

comparison of map versions or sensor data, which can be computationally expensive and sensitive to noise. More recent approaches use deep learning for detecting semantic changes in aerial imagery or street-level views. However, these methods often require extensive labeled datasets for training and may struggle with novel or unseen changes. Furthermore, they typically focus on detecting changes in the environment, not necessarily quantifying the impact of these changes on map uncertainty or providing real-time alerts for navigation systems.

Figure 1: Sources of map data uncertainty in urban driving environments such as a) uncertain road boundary positions, b) lane marking position uncertainty, c) building footprint uncertainty, d) intersection topology uncertainty, e) elevation/3D map uncertainty, and f) temporal map changes. Dashed lines indicate uncertain map features.

In summary, existing approaches to map uncertainty quantification and change detection are often heuristic-based, designed for static environments, or lack the real-time, semantic-aware capabilities required for complex urban settings. Moreover, current methods typically do not provide an on-the-fly mechanism to alert autonomous systems when significant discrepancies arise between the map and the real world, akin to the integrity monitoring systems used in aviation (Blanch et al., 2015; Brown, 1992). To address these critical gaps, we propose a novel, real-time, deep-learning-driven framework for continuous map uncertainty monitoring and semantic change detection. Our framework integrates multi-modal sensor fusion, contrastive learning for robust feature extraction, and dynamic uncertainty quantification, offering several key differentiators:

- We propose a unified framework that fuses camera and LiDAR data, unlike prior single-modality approaches, creating comprehensive feature-rich map representations that capture both geometric and semantic information.
- Instead of relying on hand-crafted heuristics or traditional statistical models, we employ deep learning-based feature embeddings. Specifically, we extract camera spatial features using convolutional neural networks (CNNs), and LiDAR data embeddings using models like PointNet (Qi et al., 2017). This data-driven approach allows for learning a more nuanced representation of map uncertainty directly from the sensor data.
- By leveraging large pre-trained vision-language models (VLMs), our approach performs zero-shot semantic segmentation to classify and explain detected discrepancies without requiring extensive retraining on new datasets. This allows for understanding what has changed semantically, a capability largely missing in existing heuristic-based or purely geometric integrity verification methods.
- Our approach provides a more comprehensive understanding of map consistency over time, moving beyond simple geometric deviations to include meaningful semantic changes.

I. PROPOSED FRAMEWORK

As shown in Figure 2, our framework detects map anomalies in urban driving scenes by jointly analysing camera imagery and pseudo-LiDAR point clouds derived from depth maps. On the vision side, Grounding DINO v2 (Ding et al., 2024) localises four critical object classes (traffic lights, traffic signs, guard-rails, poles), while the Segment Anything Model (Kirillov et al., 2023) refines each bounding box into a high-resolution mask. Pairwise comparison of before/after masks yields a per-frame change distribution whose Kullback–Leibler (KL) divergence (Kullback & Leibler, 1951) measures the severity of visual disruptions.

On the geometric side, depth images are back-projected into 3D, range filtered, and normal augmented, and then fed to a PointNet segmentation network. Removing key classes in variant scenes lets us quantify how their absence perturbs the predicted semantic distribution; the resulting KL term captures geometry-aware change.

Finally, a sensor-fusion layer combines the vision and LiDAR divergence scores through a weighted sum, producing a single scalar that flags map changes and encodes confidence. Heuristic weights balance modality trust under clear versus degraded conditions; future work will learn these adaptively so that the system can recalibrate online as lighting, weather, or sensor quality evolves.

What sets our approach apart is the principled integration of semantic change detection from both vision and LiDAR modalities using normalized KL divergence as a unifying metric. Unlike traditional voxel-level or object-counting methods that are sensitive to sampling density, occlusion, or spatial alignment, our framework interprets map change as a shift in probabilistic semantic distributions, which provides a modality-agnostic and resolution-invariant signal. Moreover, by aligning before/after scenes in both modalities under strictly controlled virtual conditions (same viewpoint, trajectory, lighting), we isolate the true semantic impact of missing infrastructure without confounding variables. This not only improves anomaly detection accuracy in diverse environments (e.g., fog, rain) but also enables fine-grained interpretability through patch-level heatmaps and per-class divergence trends.

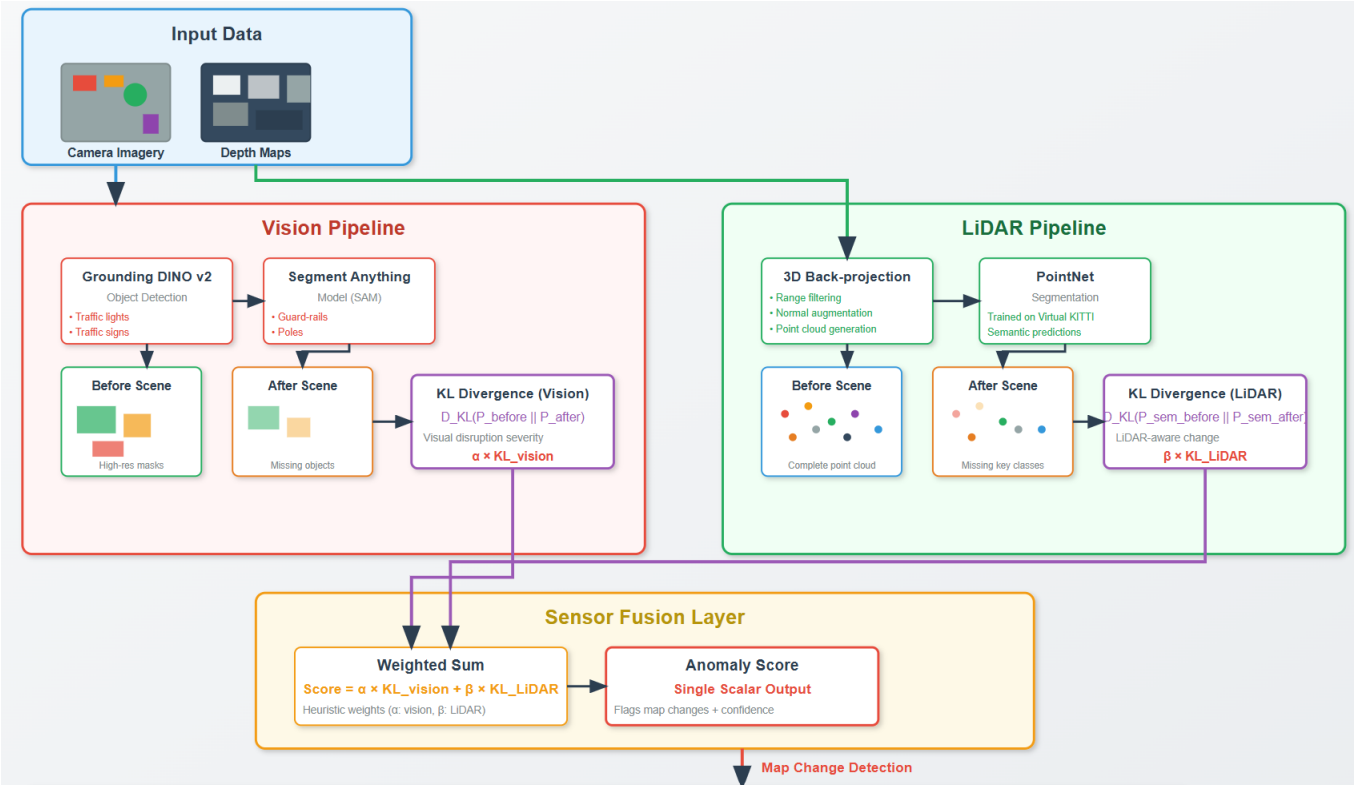


Figure 2: Our proposed multi-modal anomaly detection framework for map uncertainty.

1. Vision Module

We construct a vision pipeline centered on Grounding DINO v2 for class-level localization (Ding et al., 2024) and the Segment Anything Model (SAM) (Kirillov et al., 2023) for mask refinement. Each image pair is aligned with its corresponding LiDAR frame in terms of viewpoint, trajectory, lighting, and weather conditions. This controlled setup isolates the impact of

object removal by holding all other environmental factors constant. The complete vision-based processing pipeline is illustrated in Figure 3.

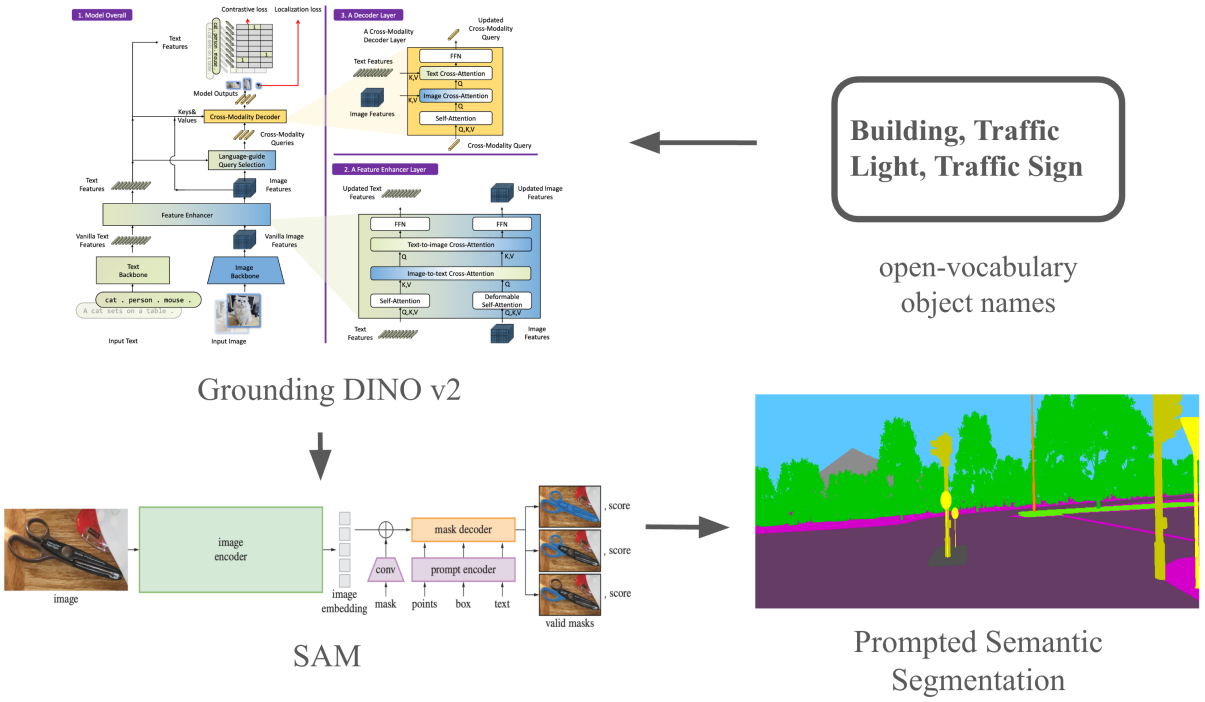


Figure 3: Vision-based change–detection pipeline.

a) Image–level Pre-processing

RGB frames are resized to 640×360 and converted to `float32` tensors. A fixed set of text prompts ("`traffic light`", "`traffic sign`", "`building`") is supplied to Grounding DINO v2, which predicts class logits and bounding boxes. Boxes whose confidence is below 0.25 are discarded.[1]. For every box that remains, we obtain a high-resolution binary mask from the SAM model. Masks are then rasterized to the original image grid so that follow-up change maps share a pixel base.

b) Per-frame Change Map

Given a normal frame I^{before} and a manipulated frame I^{after}, let

$$M^{\text{before}}(x, y), \quad M^{\text{after}}(x, y) \in \{0, 1, \ldots, K\}$$

denote the semantic mask label at pixel (x, y), where K is the number of monitored classes. We derive a binary change map

$$C(x, y) = \begin{cases} 1, & M^{\text{before}}(x, y) \neq M^{\text{after}}(x, y) \\ 0, & \text{otherwise.} \end{cases}$$

The map is smoothed with a 5×5 Gaussian kernel (σ=1.2) to damp isolated label noise. Finally, C is normalized so that $\sum_{x,y} C(x, y) = 1$. This normalization enables direct comparison using divergence-based metrics.

2. LiDAR Module

To systematically evaluate the impact of missing key scene elements—such as guardrails, traffic signals, traffic signs, and poles—on the performance of point cloud classification and anomaly detection, we created a series of targeted dataset variants. These variants are derived from the original dataset and simulate infrastructure deficiency scenarios by programmatically removing specific objects: guardrails (`noGuardRail`), traffic lights (`noTrafficLights`), traffic signs (`noTrafficSigns`),

[1]Empirically, 0.25 limits false detections without missing the target classes.

and poles (`noPoles`).

Each scene variant preserves the same viewpoints, trajectories, lighting, and weather conditions as the original dataset, ensuring that all variables remain constant except for the objects that were removed. This controlled setup allows for a rigorous and interpretable analysis of how the absence of key infrastructure components affects model performance. The full pipeline is illustrated in Fig. 6.

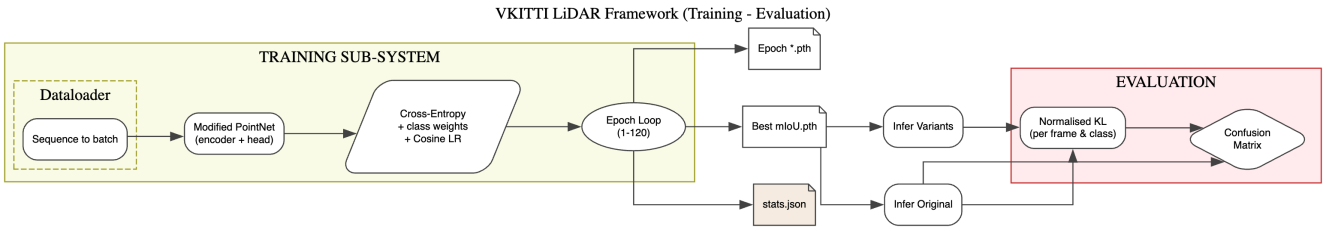


Figure 4: LiDAR Pipeline.

a) Preprocessing of Depth Image to Pseudo LiDAR Scans

We preprocess the point cloud data using the corresponding depth images and their associated semantic segmentation annotations. First, we convert the depth image to initial 3D point cloud coordinates by back-projecting using known camera internal parameters (focal length $f_x = f_y = 725.0$, principal point position $c_x = 620.5, c_y = 187.0$). This step ensures an accurate conversion from 2D image to 3D space. Subsequently, using the camera external reference information provided by the dataset, the point cloud is converted from the camera coordinate system to a unified world coordinate system to eliminate coordinate shifts due to changes in the camera position between different frames, thus providing a stable reference system for subsequent analysis.

In the back-projection process, we limit the effective range of depth values, i.e., only points within 630 meters from the camera are processed. This is because the world boundary of the virtual KITTI simulator is defined at $z = 655$ meters, and pixels beyond the valid range will be projected onto this distant boundary plane if no distance cropping is performed. This plane is not valid for both semantic and practical applications: semantically, pixels such as trees that should be located in the distant sky are incorrectly mapped to this distant and irrelevant black background region, as shown in Figure 5.

Mathematically, given the internal reference matrix of the camera:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 725 & 0 & 620.5 \\ 0 & 725 & 187.0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

For any pixel coordinate (u, v) on the image and its corresponding depth value $z(u, v)$, it can be mapped to a spatial position (X, Y, Z) under the 3D camera coordinate system by the following equation:

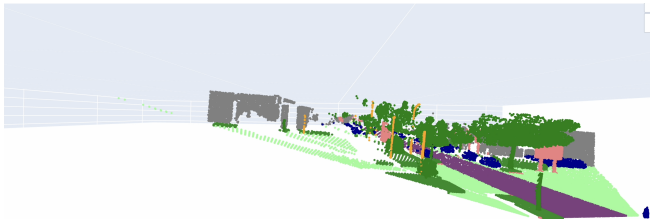
$$X = \frac{(u - c_x) \cdot z(u, v)}{f_x} \tag{2}$$

$$Y = \frac{(v - c_y) \cdot z(u, v)}{f_y} \tag{3}$$

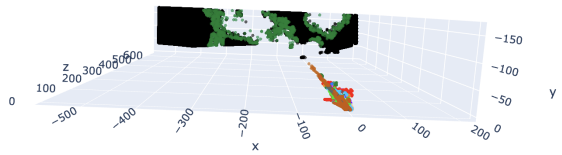
$$Z = z(u, v) \tag{4}$$

In terms of semantic information mapping, we extract the semantic segmentation annotated images corresponding to the depth map and assign each point with a corresponding semantic category identifier based on the color coding table provided in the dataset. In particular, to simplify the category analysis, we unify subcategories such as different car models (Car) to ensure the consistency of category labels.

In addition, to enhance the spatial representation of the point cloud, we use the Open3D library for normal vector estimation. Specifically, we first construct the filtered valid 3D coordinate data as Open3D point cloud objects; subsequently, we adopt an efficient KD tree for neighborhood search, set the search radius to 0.5 m, and limit the maximum number of 30 neighboring points in the neighborhood of each point, which in turn computes the local spatial structure information of each point. On this basis, the local spatial plane is fitted by the least squares method to estimate the normal vector direction of each point. This fine



(a) Projection result after applying 630 meter distance limit. Point cloud data is concentrated in the effective area, avoiding the influence of remote noise points.



(b) Wrong projection result when no distance limit ($>$ 630 meters) is imposed. Point clouds in the distance region are incorrectly mapped to the simulator boundary plane ($z = 655$ meters), causing significant semantic and structural distortions.

Figure 5: Comparative plot to illustrate he projection effect of the point cloud for the same scene before and after imposing distance limitation. The distance restriction effectively avoids the projection interference from distant irrelevant points and ensures the consistency of semantics and geometry.

processing step effectively captures the local geometric features of the point cloud, which further enhances the performance of the data in subsequent classification and anomaly detection tasks.

b) Model Architecture

The point cloud semantic segmentation model used in this paper is an enhanced implementation based on the classical PointNet framework (Qi et al., 2017). The input consists of preprocessed point cloud data, where each point is represented by its spatial coordinates (X, Y, Z). Optionally, the color information is included as additional input features. Depending on the configuration, this results in either a 3-dimensional or 6-dimensional feature vector per point.

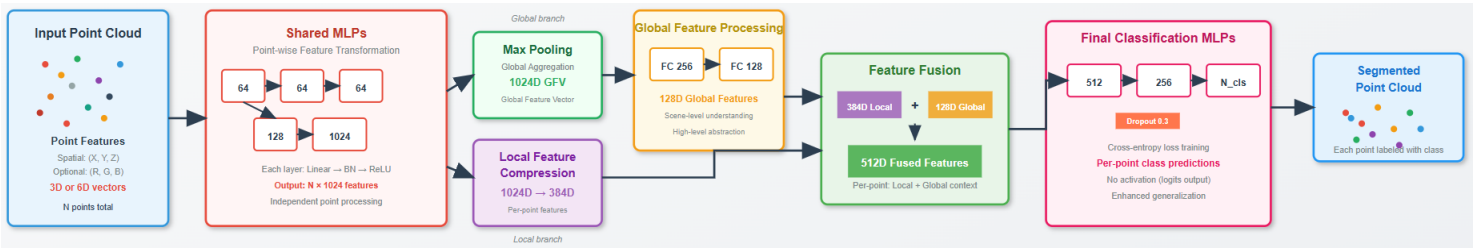


Figure 6: PointNet architecture for point cloud semantic segmentation. In the figure, MLP stands for Multi-Layer Perceptron, BN stands for Batch Normalization layer, ReLU is the activation layer, GFV stands for Global Feature Vector, and FC stands for fully connected layer.

In the input stage, the features of each point are fed into a series of shared multilayer perceptrons (MLPs), which perform feature transformations independently on a point-by-point basis. The specific shared MLP structure is in order: $[64, 64, 64, 128, 1024]$, and Batch Normalization (BN) with Revised Linear Unit (ReLU) activation function is used after each linear transformation layer to enhance the training stability and feature expression ability. Then, the features of all points are aggregated into a Global Feature Vector (GFV) with a length of 1024 dimensions through a Max Pooling layer. This global feature is then subjected to feature compression and high-level abstraction through two fully connected layers (with dimensions of 256 and 128, respectively) to capture the overall semantic information of the entire scene.

Subsequently, the network re-splices the local features corresponding to each point with the above processed global feature. To control the model complexity, we compress the original 1024-dimensional local features to 384 dimensions, and then splice them with the 128-dimensional global features to form a new 512-dimensional feature representation. This strategy of fusing local and global features makes each point not only have location-related local features, but also incorporates contextual understanding of the overall scene, thus improving the semantic differentiation ability of the model.

Eventually, after a sequence of MLP layers ($[512, 256, N_{\text{cls}}]$, where N_{cls} is the number of categories), the model generates a category prediction score corresponding to each point. A Dropout mechanism with 0.3 probability is introduced in the penultimate layer of the MLP to reduce overfitting and enhance model generalization. The output layer no longer uses activation functions and directly outputs logits for each category for subsequent training optimization using the cross-entropy loss function.

3. Consistency Monitoring using Joint Divergence Scores

We quantify map–change detection using a single composite score, computed as a weighted average of the KL divergence values generated by the vision and LiDAR modules.

Let

$$D_{\mathrm{KL}}^{\mathrm{vis}} \quad \text{and} \quad D_{\mathrm{KL}}^{\mathrm{lidar}}$$

denote the KL divergence values returned by the vision and LiDAR pipelines, respectively. The combined score is defined as

$$\boxed{S \;=\; \alpha\, D_{\mathrm{KL}}^{\mathrm{vis}} \;+\; \beta\, D_{\mathrm{KL}}^{\mathrm{lidar}},} \qquad \alpha, \beta \in [0,1],\ \alpha + \beta = 1.$$

In this work the weights α and β are chosen heuristically to balance the relative confidence we place in each sensing modality. Future research will investigate adaptive weighting strategies, such as learning α and β online from cross-validated performance or Bayesian evidence, so that the score can automatically reflect changing sensor quality or environmental conditions.

II. EXPERIMENTS

1. Dataset

We use the Virtual KITTI dataset (Cabon et al., 2020) to evaluate our proposed framework due to its uniquely controlled and richly annotated synthetic environment. Unlike real-world datasets, Virtual KITTI offers pixel-level ground truth for semantic segmentation, depth, optical flow, and multi-object tracking across a variety of environmental conditions (e.g., lighting, fog, rain). This makes it an ideal testbed for benchmarking fine-grained scene understanding methods, particularly when evaluating divergence scores that depend on reliable, consistent ground truth across multiple modalities. Furthermore, the dataset allows us to isolate the effects of specific visual and structural changes in the scene, such as camera motion, object trajectory shifts, and weather perturbations, making it especially well-suited for testing our vision- and LiDAR-based modules under known map perturbations. The photorealistic rendering and exact annotations generated from 3D virtual worlds cloned from real KITTI sequences ensure that our experimental analysis remains both reproducible and representative of urban driving scenarios, while avoiding the cost and noise of manual labeling.

To systematically evaluate change detection algorithms under controlled scene alterations, we construct an artificial anomaly dataset by selectively removing specific semantic object classes from RGB images and point clouds, and their associated segmentation masks. This process simulates real-world infrastructure loss, such as missing traffic lights or signage, while preserving all other scene attributes. The classes targeted for removal in this study are: `Building`, `TrafficLight`, and `TrafficSign`.

We use the RGB and semantic mask outputs from the Virtual KITTI dataset as the basis for generating manipulated data. For each frame:

1. We identify all pixels corresponding to the selected semantic classes based on the dataset's official color-encoding table.

2. For each class found in a frame, we create a per-frame, per-class binary mask indicating the affected regions.

3. A union mask of all target-class pixels is then constructed and optionally dilated to account for class boundaries.

4. The RGB image is inpainted using the Telea inpainting algorithm (OpenCV `INPAINT TELEA`, $radius = 5$) to fill removed regions with plausible texture from nearby pixels. And the corresponding segmentation map is modified by overwriting the affected pixels with a designated void color `(0,0,0)`, rendering them semantically undefined.

2. SETUP

For the vision module, Grounding DINO and SAM are used in zero-shot mode with their official pretrained checkpoints. All images are processed on a single NVIDIA RTX 5080 GPU, achieving an average throughput of 6.8 fps, including both Grounding DINO inference and SAM-based masking. The CLIP and LoFTR baselines are evaluated using the same input resolution and hardware setup. All implementations are built using libraries such as PyTorch (Paszke et al., 2019).

For the LiDAR module, training is performed using the AdamW optimizer (Loshchilov & Hutter, 2019) with an initial learning rate of 2×10^{-4}, a weight decay of 1×10^{-4}, and a cosine annealing learning rate schedule to ensure smooth convergence in later stages. The model is trained for 120 epochs with a batch size of 16. During each epoch, the training data is randomly sampled and shuffled to promote robust learning across diverse data distributions.

To mitigate overfitting, we adopt an $80\%/20\%$ train–validation split and evaluate model performance on the validation set after each epoch. During training, each point cloud frame is randomly sampled to 4096 points. If a frame contains fewer than 4096

Figure 7: Frames from 5 real KITTI videos (left, sequences 0001, 0002, 0006, 0018, and 0020 from top to bottom) and rendered virtual clones (right) from Virtual KITTI dataset (Cabon et al., 2020).

points, random resampling with replacement is applied to match the target size. Training is executed on either CUDA-enabled GPUs or fallback CPUs, depending on hardware availability, to ensure efficient computation.

III. RESULTS

Our results are structured to answer three key questions:

- How accurately can each individual modality detect semantic changes in the map under normal and degraded conditions?
- How well do the predicted anomaly distributions align with ground-truth changes induced by simulated infrastructure removal?
- Can fusing information from vision and LiDAR improve map-change detection in diverse conditions?

To address these questions, we report detailed quantitative results under both normal (clear weather) and adverse (rainy, foggy) scenarios using a set of controlled synthetic experiments on the Virtual KITTI dataset. For each condition, we evaluate performance using KL divergence values against ground-truth pixel-level change maps. We also report True Positive Rates (TPRs) across semantic categories to assess per-class detection capability. Finally, we demonstrate that our sensor fusion strategy improves robustness and reliability, particularly in visually degraded environments.

1. Vision Module Results

a) Evaluation Method

For every RGB pair, we derive a pixel–wise change distribution $P_t(x, y)$. To gauge how well a detector $Q_t(x, y)$ reproduces the true change map[2], we compute the frame-wise KL divergence

$$D_{\mathrm{KL}}(P_t \,\|\, Q_t) = \sum_{x,y} P_t(x, y) \, \ln \frac{P_t(x, y)}{Q_t(x, y) + \varepsilon}, \quad \varepsilon = 10^{-12}.$$

A lower value indicates closer alignment with ground truth.

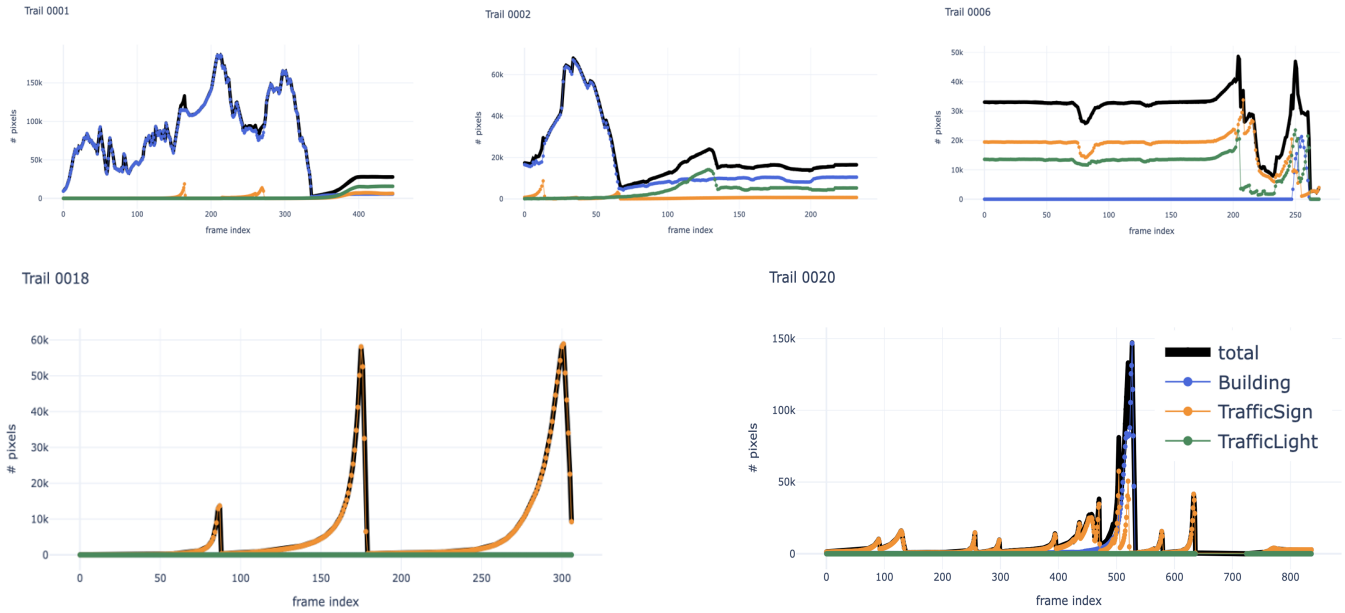


Figure 8: Temporal distribution of pixel-level changes across five annotated sequences in the dataset. The black line represents the total number of changed pixels per frame, while the colored lines break down the contributions by semantic class: `buildings (blue), traffic signs (orange), and traffic lights (green).` Peaks in the curves correspond to frames with significant structural or semantic modifications, such as the sudden appearance of signs or occlusion of buildings, which are critical for anomaly localization and map update validation.

As baselines, we use the following:

- **CLIP (ViT-B/32) with Patch Difference:** Each RGB frame is resized to 224×224, preprocessed, and passed through CLIP's visual encoder (excluding global pooling) to produce a 7×7 grid of ℓ_2-normalized 512-dimensional patch tokens. Cosine similarity is computed between corresponding tokens of before/after images, and pixel-wise change is defined as $1 - \cos\theta_{xy}$. The resulting 7×7 change map is bicubic-upsampled to the original resolution and used directly for KL divergence computation and anomaly visualization.
- **LoFTR (Detector-Free Transformer Matching) (Sun et al., 2021):** Input images are converted to grayscale, zero-centered, and resized to satisfy LoFTR's stride requirements. LoFTR generates sparse coarse correspondences with confidence scores $\alpha_i \in [0, 1]$. These are rasterized onto a $H/8 \times W/8$ grid, assigning each score to its nearest cell; empty cells are set to zero. The resulting confidence map $S(u, v)$ is bilinearly upsampled to full resolution, and anomaly scores are computed as $1 - S(x, y)$. The map is normalized for KL divergence computation and also used as a dense heatmap for visualization.

b) Average TPR under Normal Conditions

The RGB inputs have a native resolution of 1242×375 pixels at $72\,\mathrm{ppi}$. An anomaly is considered as a true positive when two conditions are met: (i) the ground-truth mask contains at least 9000 changed pixels, which corresponds to approximately 2% of

[2]Ground-truth change equals the fraction of artificially removed pixels.

the image, and (ii) the model's score exceeds a decision threshold. For our method and the CLIP-based baseline, this threshold is set to KL divergence ≥ 1.0; for the LoFTR-based baseline, an anomaly score threshold of ≥ 0.99 is used.

As shown in Table 1, our vision-only anomaly detection framework achieves an average TPR (TPR) of 95.02% across all object categories and sequences under clear weather conditions. The method detects all anomalies (100% TPR) in Sequences 0001, 0002, and 0006, demonstrating strong sensitivity to missing infrastructure when the scene is well-lit and unoccluded. While performance drops in Sequence 0020, where building detection reaches only 39.19%, we attribute this to occlusions and limited camera field of view, which make the missing structures less visually salient. Nonetheless, the model maintains high TPRs across object types: 84.80% for buildings, 83.92% for `traffic lights`, and 81.63% for `traffic signs`, underscoring its robustness even with smaller and potentially occluded features.

By comparison, the CLIP-based baseline shown in Table 2 achieves a lower overall TPR of 74.97%. Detection accuracy for buildings falls to 60.33%, and performance on smaller objects such as lights (60.40%) and signs (60.35%) lags significantly behind our method. The baseline also exhibits sequence-level gaps where no predictions are made, further limiting its utility in consistent scene monitoring.

The LoFTR-based baseline, detailed in Table 3, performs the weakest among the three approaches. Its overall TPR is only 63.84%, with `building` detection at 55.36% and `traffic lights` and signs at 50.84% and 48.13%, respectively. In more complex scenes such as Sequence 0018 and 0020, the model often fails to trigger, likely due to inadequate keypoint matching in structurally altered or low-texture areas.

These results emphasize the advantages of our approach, which explicitly incorporates spatial context and category-aware scoring to improve sensitivity to both large-scale structural changes and finer infrastructural elements.

Table 1: TPR from our approach under normal conditions.

Category	Trial ID					Avg.
	0001	0002	0006	0018	0020	
Building	100.00%	100.00%	100.00%	–	39.19%	84.80%
Traffic Light	71.71%	80.43%	99.62%	–	–	83.92%
Traffic Sign	72.06%	62.09%	100.00%	76.92%	97.10%	81.63%
Total	100.00%	100.00%	100.00%	76.92%	98.20%	95.02%

Table 2: TPR from CLIP-based baseline under normal conditions.

Category	Trial ID					Avg.
	0001	0002	0006	0018	0020	
Building	82.04%	72.12%	65.34%	–	21.83%	60.33%
Traffic Light	45.39%	62.33%	73.47%	–	–	60.40%
Traffic Sign	39.96%	45.87%	82.14%	56.48%	77.31%	60.35%
Total	82.04%	72.12%	84.81%	56.48%	79.41%	74.97%

Table 3: TPR from LoFTR-based baseline under normal conditions (Anomaly score ≥ 0.99).

Category	Trial ID					Avg.
	0001	0002	0006	0018	0020	
Building	75.66%	65.71%	59.74%	–	20.31%	55.36%
Traffic Light	38.12%	53.26%	61.13%	–	–	50.84%
Traffic Sign	31.64%	36.68%	65.82%	45.29%	61.22%	48.13%
Total	75.66%	65.71%	67.22%	45.29%	65.34%	63.84%

c) Average TPR under Foggy and Rain Conditions

Table 4 presents the average TPR of our method under adverse weather conditions, including fog and rain. These settings simulate real-world visibility challenges where vision-only systems often degrade. Despite these conditions, our method remains notably

resilient, achieving a total average TPR of 79.58%. Building detection remains relatively strong with an average of 77.22%, although Sequence 0020 sees a significant drop to 16.70%, likely due to occlusions and the compounded effects of visual degradation. Small object categories such as traffic signs show reduced performance (49.02%), which is expected given their lower contrast and greater susceptibility to noise and weather-induced blur.

In contrast, the CLIP-based baseline (Table 5) shows a lower average TPR of 58.61%. While Sequence 0001 retains high building detection (78.83%), the average building TPR drops to 50.44%, and the performance on lights (39.32%) and signs (40.84%) remains modest. The model's reliance on global image embeddings without sufficient spatial grounding may explain this lack of robustness under degraded conditions.

The LoFTR-based baseline, shown in Table 6, performs the weakest, with a total average TPR of only 43.35%. Detection of buildings is particularly affected, falling to 37.43% on average. For `traffic lights` and `traffic signs`, the TPRs are 34.25% and 32.59%, respectively. The degradation is likely due to poor keypoint matching reliability in visually noisy scenes, which prevents consistent localization of changed regions.

Together, these results highlight the superior robustness of our approach in inclement weather. While all vision-based models suffer from degraded inputs, our method maintains higher recall across all categories and scenarios, emphasizing the value of contextual awareness and anomaly scoring grounded in localized visual features.

Table 4: TPR under rain and foggy conditions for our approach (%).

Category	0001	0002	0006	0018	0020	Avg.
Building	99.33	100.00	92.86	–	16.70	77.22
Traffic Light	55.26	63.91	95.44	–	–	71.54
Traffic Sign	46.15	4.27	97.78	41.92	54.97	49.02
Total	99.33	100.00	100.00	41.92	56.63	79.58

Table 5: TPR for CLIP-based baseline (**KLD** ≥ 1) under rain and foggy conditions.

Category	Trial ID					Avg.
	0001	0002	0006	0018	0020	
Building	78.83%	63.35%	46.86%	–	12.73%	50.44%
Traffic Light	29.10%	39.81%	48.72%	–	–	39.32%
Traffic Sign	27.31%	30.73%	55.67%	38.41%	52.09%	40.84%
Total	78.83%	63.35%	58.12%	38.41%	54.33%	58.61%

Table 6: TPR for LoFTR-based baseline under rain and foggy conditions.

Category	Trial ID					Avg.
	0001	0002	0006	0018	0020	
Building	51.31%	44.68%	39.86%	–	13.87%	37.43%
Traffic Light	25.94%	35.72%	41.08%	–	–	34.25%
Traffic Sign	21.45%	25.07%	44.38%	30.78%	41.25%	32.59%
Total	51.31%	44.68%	46.23%	30.78%	43.75%	43.35%

d) Qualitative Results

To localize the spatial regions associated with anomalies, we compare patch-level visual representations from normal and anomalous frames. These representations are extracted using a self-supervised ViT model (DINOv2-ViT-B/14) (Dosovitskiy et al., 2020), which is also used in our detection pipeline. Each image is resized and center-cropped to 224×224, normalized, and then passed through the DINOv2 backbone. From the intermediate output, we extract normalized patch tokens corresponding to a 14×14 grid, resulting in 196 spatial tokens where each token represents a 16×16 region of the original image.

To quantify localized changes, we compute the Euclidean distance between the corresponding patch tokens from the normal and anomalous images. The resulting distance map is visualized as a heatmap, providing an interpretable overlay that highlights

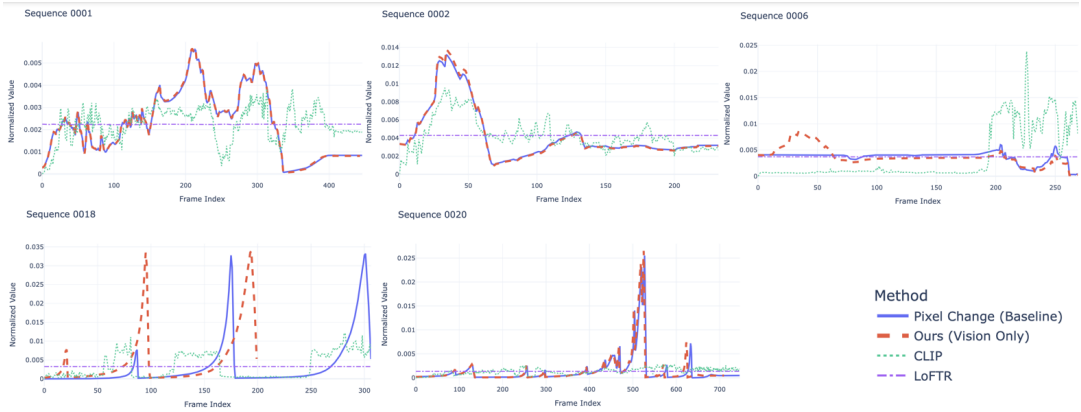


Figure 9: KL Divergence using our method, CLIP-based method, and LoFTR v.s. pixel change over frame

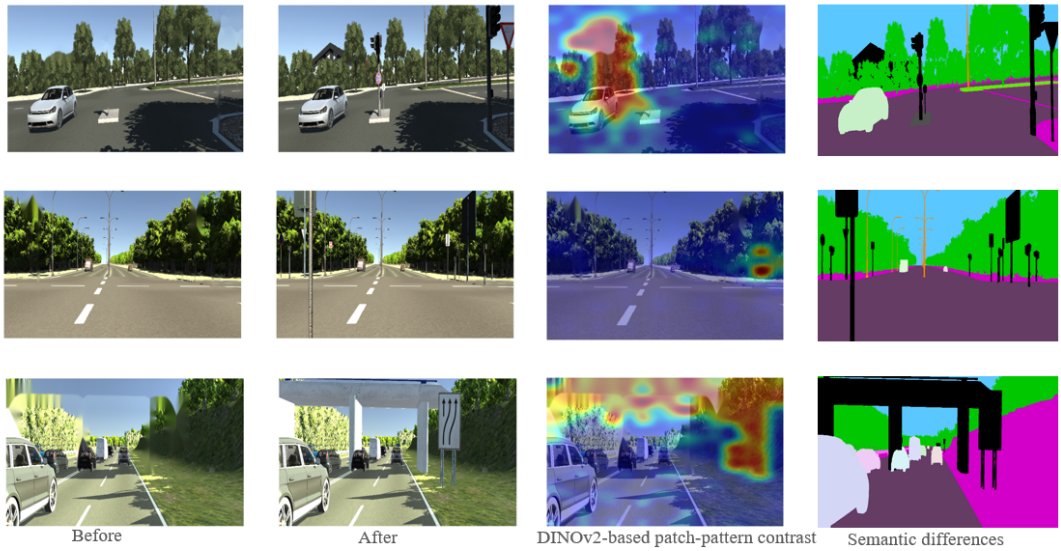


Figure 10: Qualitative results showing semantic anomaly detection across three representative scenes. From left to right: original (nominal) image, modified (anomalous) image with structural changes, heatmap of DINOv2-based patch-pattern contrast highlighting regions of detected change, and corresponding semantic differences using ground-truth labels. The heatmaps accurately localize modifications such as added traffic signs, altered light poles, and occluded structures, aligning well with semantic-level ground truth.

regions of potential anomalies. For frames where an anomaly is both present and correctly detected, we evaluate localization quality using the Intersection over Union (IoU) metric between the predicted anomaly region and the ground-truth change mask.

Table 7 reports the mean IoU between the predicted anomaly heatmaps that are generated from the patch-token distance and the ground-truth change masks. Overall, our method achieves a strong average mIoU of 0.634 across all sequences and classes, indicating reliable spatial alignment between predicted and true anomaly regions. Buildings are the most consistently localized class (avg. 0.646), due to their large size and structural regularity, while traffic signs show more variation depending on occlusion and lighting conditions. Sequence-wise, performance is highest in 0006 (0.752) and lowest in 0002 (0.535), reflecting the influence of scene complexity and contrast. These results validate that our method not only detects the presence of map anomalies but also accurately localizes them at the pixel level.

Table 7: Segmentation quality (mIoU) per class across sequences.

Category	Seq_0001	Seq_0002	Seq_0006	Seq_0018	Seq_0020	Avg.
Building	0.723	0.769	0.770	–	0.322	0.646
Traffic Light	0.651	0.540	0.734	–	–	0.642
Traffic Sign	0.639	0.297	0.751	0.604	0.781	0.614
mIoU	0.671	0.535	0.752	0.604	0.552	**0.634**

2. Lidar Module Results

a) Evaluation Method

To systematically analyze changes in point-cloud data across frames and establish a clear baseline for evaluation, we adopt two complementary metrics: Change Ratio (CR) and Jaccard Distance (JD) (Jaccard, 1912). CR captures point-level geometric differences, making it effective for detecting local structural changes. JD compares voxel-level occupancy states, offering robustness across varying periods, sensors, and resolutions. Together, these metrics provide a reliable foundation for assessing sensitivity and stability in point-cloud change detection.

Change Ratio (CR): The change ratio is used to quantify the geometric difference between the current frame and the reference frame at the point cloud level and is defined as follows:

$$\mathrm{CR} = \frac{|P_{\text{changed}}|}{|P_{\text{cur}}|}, \tag{5}$$

where P_{cur} denotes the entire set of points in the current frame and P_{changed} denotes the set of points that fall within the voxels that were previously empty in the reference frame. The rate of change takes values between 0 and 1: values close to 0 indicate that the current scan is highly consistent with the reference map, and values close to 1 indicate a large number of structural differences. As an early warning indicator for rapid change detection, CR is suitable for building deformation monitoring, temporary obstacle detection, and other scenarios that require timely detection of local changes. However, it should be used in conjunction with density compensation measures when comparing across devices or resolutions.

Jaccard Distance (JD): The Jaccard distance is a normalized global metric based on ensemble intersection and merging operations, which is used to evaluate the similarity of two frames of point clouds at the voxel level. Let V_{ref} and V_{cur} be the sets of voxels occupied by the reference frame and the current frame, respectively; then JD is defined as:

$$\mathrm{JD} = 1 - \frac{|V_{\text{ref}} \cap V_{\text{cur}}|}{|V_{\text{ref}} \cup V_{\text{cur}}|}, \tag{6}$$

This formula is essentially the complementary form of the Intersection over Union (IoU) ratio, which again takes values between 0 and 1: when the set of voxels in the two frames is identical, JD equals 0; When there are no overlapping voxels, JD equals 1. Unlike CR, JD focuses only on whether a voxel is 'occupied' or not, regardless of the number of points within a single voxel, and is therefore more robust to point cloud density, sampling noise, and single point errors. This feature makes it particularly suitable for long-term sequential map update tasks in time, resolution, and devices.

For point clouds we compare the class-probability distributions extracted from the original scan and its object-removed variant. Let $p_{\text{orig}}(c)$ and $p_{\text{var}}(c)$ be the class histograms, and $N_{\text{orig}}(c), N_{\text{var}}(c)$ their raw point counts. A naïve, class-wise KL term

$p_{\text{orig}}(c)\ln\big[p_{\text{orig}}(c)/p_{\text{var}}(c)\big]$ tends to explode in sparse classes. We therefore introduce a *weighted, normalised* KL scatter

$$\frac{1}{\ln K}\,\frac{\min\big(N_{\text{orig}}(c),N_{\text{var}}(c)\big)}{N_{\text{orig}}}\;p_{\text{orig}}(c)\;\ln\frac{p_{\text{orig}}(c)}{p_{\text{var}}(c)},$$

which bounds each class to $[0,1]$ and the global sum to $[0,K]$. The weight down-scales highly volatile but tiny classes (e.g. guard-rails, poles) so they no longer dominate the frame score, and optional truncation $\max(0,\cdot)$ guarantees non-negativity.

By analyzing the overall normalized KL scatter (see Figure 12), we found that scenes with different sequences exhibited significantly different model sensitivities after key semantic elements were removed. In the initial frames of both sequences 0001 and 0002, the KL scatter shows a significant peak, followed by a rapid fall back to a relatively stable state. This indicates that the model is more sensitive to changes in the semantic environment at the beginning of these scenes, and as the scenes continue to unfold and the feature environment stabilizes, the model is able to adapt gradually, and the KL scatter tends to stabilize, showing that the model possesses a certain degree of scene adaptability.

In contrast, sequences 0006 and 0020 show a more complex trend, especially sequence 0006 shows a continuous upward trend of KL scatter in the middle and late stages, suggesting that the scene features of this sequence gradually show more perceptual differences triggered by the absence of key semantic elements in the subsequent frames. In particular, sequence 0006 has much higher KL scatter values than other sequences in subsequent frame times, which may imply that certain infrastructure elements in the scene occupy a more important position within the model's viewpoint, which, once missing, significantly interferes with the semantic understanding and classification stability of the overall scene.

The fluctuation of the overall KL dispersion of sequence 0018 is significantly smaller, with a smoother trend and consistently lower values, which contrasts with the significant fluctuations of the other sequences. This implies that the removal of key semantic elements has a limited impact on the overall perceptual ability of the model under such scene-specific conditions and indirectly suggests that the model may be more robust in dealing with semantic deficits in some specific scenes.

b) Semantic Categorization Performance

We next conduct an in-depth analysis of the semantic categorization performance. A quantitative analysis of the confusion matrix, as shown in Figure. 14, reveals significant variation in classification performance across categories. Notably, `Terrain`, `Building`, and `Car` consistently achieve high accuracy, indicating the model's stability and reliability in recognizing categories with well-defined structural features.

However, the model showed significant misclassification between categories with slender structures and some overlap in spatial distribution and morphological characteristics, such as `TrafficLight`, `Traffic Sign`, and `Pole`. Among these three categories, the model shows some instances of misclassification. For example, sequences 0002 and 0006 have significantly higher misclassification ratios between these categories, and the normalized accuracies of the rows are even lower than 50% at some intersections of categories, revealing that the model has greater challenges in capturing the spatial morphology differences of elongated objects, which needs to be strengthened and optimized in future model design.

In particular, sequences 0006 and 0020 show more misclassifications between the categories of `Car` and `Misc`, which may be related to the atypical appearance of the vehicle in the scene or the semantic ambiguities caused by the lighting conditions of the scene. This may stem from atypical vehicle appearance or scene-induced semantic ambiguity due to lighting conditions. Such phenomena suggest that more atypical sample features may need to be considered in future data enhancement strategies or model design to improve the model's classification accuracy for complex and non-standard vehicles.

c) KL Divergence by Category

Further, from the category-by-category normalized KL scatter results (see Figure 13), the absence of a specific category affects the overall perceptual ability of the model to different degrees. Across all test sequences, the key infrastructure categories of `Building`, `Traffic Light`, and `Traffic Sign`, and other key infrastructure classes all show a clear upward trend in KL scatter values, and this change is more significant and stable across multiple sequences (especially sequences 0001 and 0002). This indicates that the model exhibits significant sensitivity to the absence of these infrastructure classes, which happen to be critical for semantic understanding and driving decisions in realistic autonomous driving scenarios.

In sequence 0006, the KL dispersion of the categories `GuardRail` and `Pole` varies significantly stronger than the other sequences, suggesting that guardrails and poles may play a key role in the model's spatial perception and semantics. This suggests that guardrails and poles play a critical role in the model's spatial and semantic perception. Once these objects are removed, the overall structure of the scene and the semantic consistency perceived by the model will be negatively affected.

However, it should be noted that some non-target categories such as `Terrain` and `Miscellaneous Objects (Misc)`, show significant fluctuations in KL scatter in some frames of sequences 0018 and 0020. This indicates that the absence of key

categories can lead to misclassification of secondary categories, highlighting the need to improve the model's robustness to background class shifts in complex scenes to preserve overall perceptual performance.

d) Semantic categorization and benchmark analysis

We combine voxel-based benchmark metrics (Jaccard index and change ratio) with a distributional measure of normalized KL scatter to provide a comprehensive description of the semantic performance and robustness of the PointNet model.

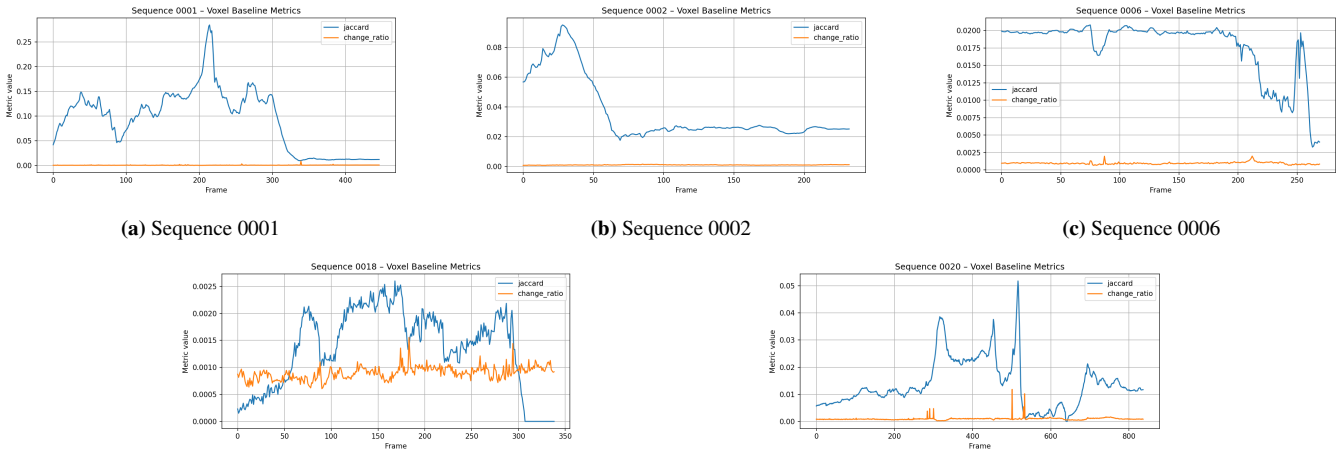
The voxel-based Jaccard index and change ratio metrics (Figure 11) provide a different perspective on the stability of the model's performance.The Jaccard index primarily captures the structural completeness and volumetric overlap of the semantic fragments, whereas the change ratio across sequences remains static, indicating little sensitivity to fine-grained semantic changes. Specifically, sequences 0001 and 0002 show significant peaks and valleys in the Jaccard index, indicating significant changes in scene content, while sequence 0018 shows only slight oscillations, which is consistent with its more homogeneous semantic composition and indicates limited sensitivity to minor semantic perturbations. Sequences 0006 and 0020, on the other hand, show a more complex pattern, with Jaccard values showing a significant plunge when key semantic elements are removed, highlighting the critical role of certain infrastructure elements in model perception.

The normalized KL scatter results (Figure 11) complement the voxel-level metrics by revealing subtle changes in the semantic confidence distribution in the model. Although voxel-level metrics primarily highlight volumetric and geometric differences, KL scatter provides insights into how semantic priors migrate and redistribute after scene element removal. Sequences 0001 and 0002 have significant spikes in KL scatter values in the initial frames, indicating that scene changes initially perturbed the prior severely. These early perturbations subsequently stabilize, showing that the model is able to effectively adapt or recalibrate the semantic prior. In contrast, Sequence 0006 shows a sharp increase in KL scatter in later frames, indicating a gradual accumulation of perceptual uncertainty or semantic redistribution due to the removal of key structural components.

The persistent slight negative KL scatter observed in sequence 0002 and part 0018 implies a counterintuitive decrease in scatter, possibly indicating an abnormal increase in model prediction confidence in the absence of key semantic inputs. This pattern reveals a potential vulnerability of semantic inference mechanisms, whereas voxel-level analysis struggles to recognize such phenomena.

Differences between the behavior of the Jaccard index and the KL scatter reveal discrepancies between geometric stability and semantic confidence distributions. For example, while sequences 0018 and 0020 have moderate or low Jaccard fluctuations, significant KL scatter variations reveal the model's vulnerability to subtle but far-reaching semantic perturbations. In contrast, the increase in KL scatter in sequence 0006 at a later stage is closely correlated with a sudden drop in the Jaccard index, together emphasizing the importance of infrastructural elements such as guardrails for semantic coherence.

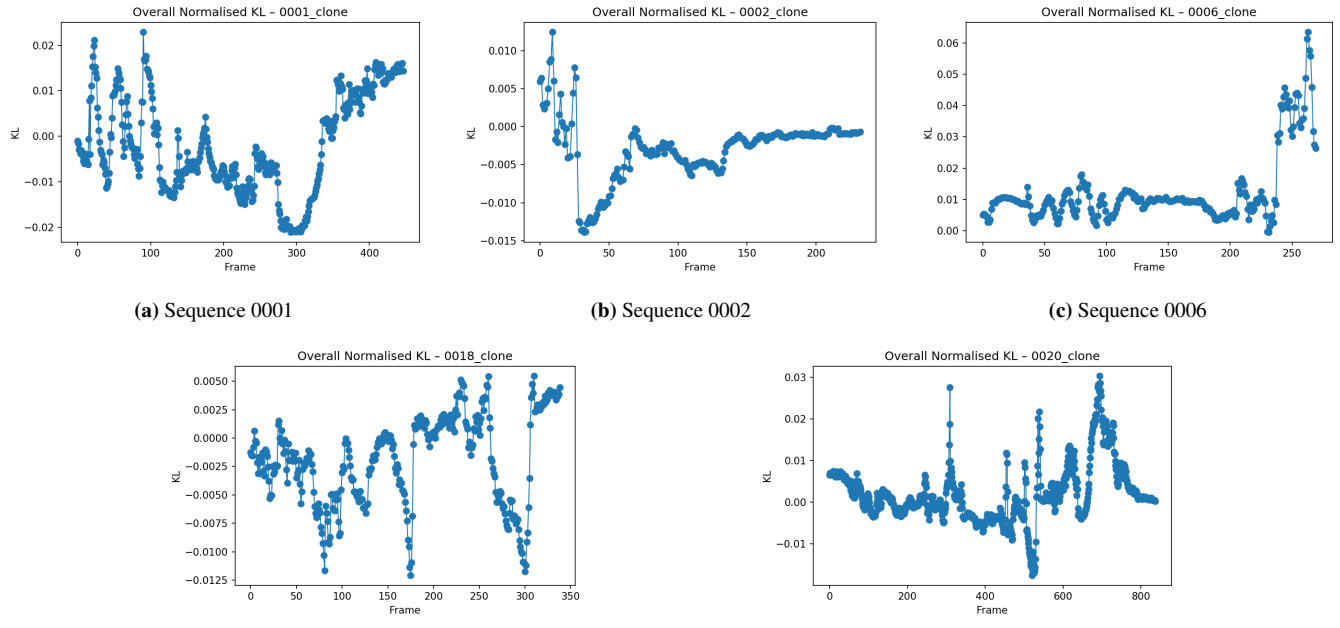
The observed differences between the elemental and distributional metrics emphasize two key aspects of semantic segmentation model design: first, high semantic importance is likely to be concentrated in the smaller but significant infrastructure categories, such as traffic signs, poles, and guardrails, whose absence has an impact on semantic coherence that far exceeds their geometrically present proportions; second, distributional metrics such as normalized KL scatter can serve as early indicators of model vulnerability, capturing subtle changes before semantic instability manifests itself visibly in voxel metrics. Therefore, a comprehensive assessment approach that incorporates both voxel overlap and semantic confidence distributions is essential for comprehensive model evaluation and optimization.



(a) Sequence 0001 **(b)** Sequence 0002 **(c)** Sequence 0006

(d) Sequence 0018 **(e)** Sequence 0020

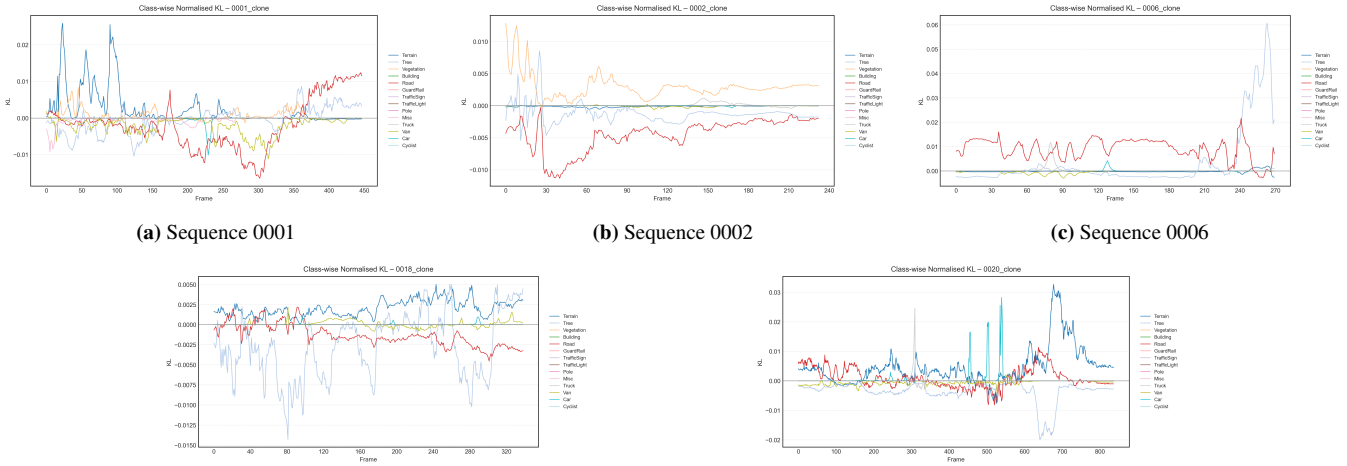
Figure 11: LiDAR baseline-metric trends across representative sequences.



(a) Sequence 0001 **(b)** Sequence 0002 **(c)** Sequence 0006

(d) Sequence 0018 **(e)** Sequence 0020

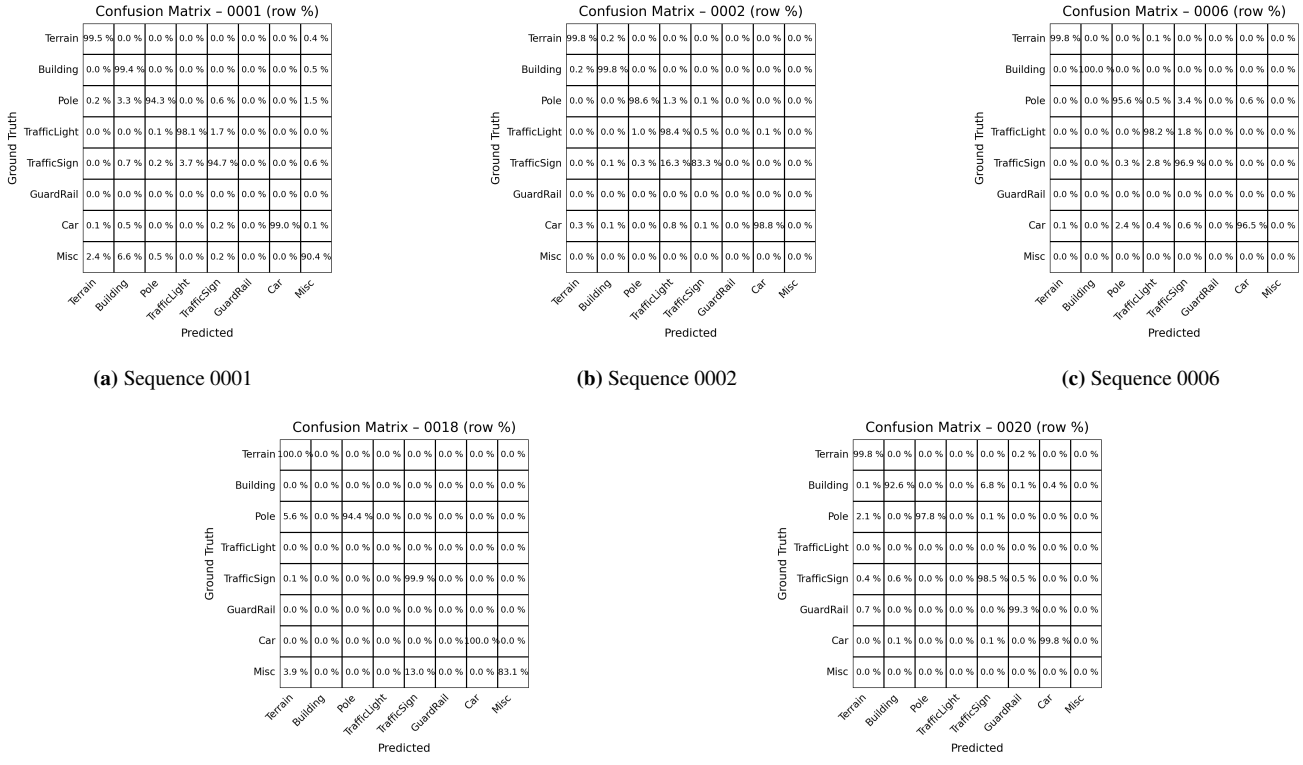
Figure 12: Overall normalized KL divergence trends across sequences. Notable differences and variability indicate sensitivity of the PointNet model to semantic omissions.



(a) Sequence 0001 **(b)** Sequence 0002 **(c)** Sequence 0006

(d) Sequence 0018 **(e)** Sequence 0020

Figure 13: Class-wise normalized KL divergence illustrating sensitivity for individual semantic classes across different sequences.



(a) Sequence 0001 **(b)** Sequence 0002 **(c)** Sequence 0006

(d) Sequence 0018 **(e)** Sequence 0020

Figure 14: Confusion matrices for semantic classification illustrating model accuracy and misclassification tendencies across key categories.

3. Sensor Fusion Results

a) Evaluation Method

After benchmarking the *vision* and *LiDAR* branches independently, we assess the fused output. This balanced weighting leverages complementary strengths, semantic richness from RGB, and spatial consistency from LiDAR, to improve map change detection in visually degraded scenes. Two different weighting configurations were tested under distinct environmental conditions:

- **Normal Conditions (Clear Weather):** Using $\alpha = 0.8$ and $\beta = 0.2$, we prioritize RGB-based semantic inferences. In this setting, no major improvements were observed, as RGB alone was already highly accurate.
- **Raining and Foggy Conditions:** We set $\alpha = 0.5$ and $\beta = 0.5$ to equally trust both RGB and LiDAR features. This configuration led to major improvements, especially for the `Traffic Sign` and `Traffic Light` classes. The average total accuracy improved to 86.19% under degraded visual conditions.

We assess how closely each method approximates true pixel-level changes under normal conditions using two metrics: KL Divergence, which quantifies probabilistic mismatch (lower is better), and Pearson Correlation, which measures spatial alignment between predicted and ground-truth change maps (higher is better).

We compare our method against two baselines. The first combines a CLIP-based vision module (ViT-B/32, patch size 32) with a LiDAR-based Jaccard distance algorithm for map-change detection. The second uses LoFTR—a detector-free local feature matching transformer—for vision, also paired with the same LiDAR-based Jaccard metric.

Note that each per-frame sequence is treated as a discrete distribution over time. To allow valid comparison, the sequences are normalized to sum to 1, and then the KL-Divergence is calculated. For fair comparison, each sequence is normalized using z-score scaling before computing the Pearson correlation.

b) Kl Divergence Score and Pearson Coefficient Evaluation

As seen in Table 8, our fusion-based method achieves an exceptionally low average KL divergence of 0.1064, indicating strong agreement with the ground-truth change maps. Most sequences (e.g., Sequence 0001 and 0002) show near-zero divergence, while even the more complex Sequence 0020 remains under 0.27. Pearson correlation values in Table 9 further affirm this alignment, with an average of 0.7180, indicating high spatial similarity between the predicted and actual anomaly regions.

In contrast, the CLIP-based baseline exhibits substantial KL divergence (avg. 0.6331) and a lower average Pearson correlation of 0.3808. Although it captures some structural anomalies, it lacks spatial reasoning due to its reliance on global embeddings rather than localized semantic context. The LoFTR-based method performs even worse in terms of correlation (0.1500), reflecting noisy or misplaced anomaly predictions. Despite a low KLD in Sequence 0006, LoFTR's predictions remain spatially inconsistent. This disparity highlights that low divergence alone does not imply good spatial agreement if the prediction lacks consistency.

Table 8: KL divergence from true pixel change distribution (lower is better) under normal conditions.

Method	seq_0001	seq_0002	seq_0006	seq_0018	seq_0020	Avg.
Ours	0.0021	0.0005	0.0493	0.2100	0.2702	**0.1064**
CLIP-based + Jaccard	0.1974	0.1345	0.8734	1.0733	0.8867	0.6331
LoFTR-based + Jaccard	0.2037	0.1300	0.0401	1.0783	1.1894	0.5283

Table 9: Pearson correlation with true pixel change distribution (higher is better) under normal conditions.

Method	seq_0001	seq_0002	seq_0006	seq_0018	seq_0020	Avg.
Ours	0.9973	0.9997	0.5933	0.0873	0.9121	**0.7180**
CLIP + Jaccard	0.5538	0.7854	0.0928	0.2616	0.2102	0.3808
LoFTR + Jaccard	0.2985	0.3065	0.0129	0.1030	0.0291	0.1500

Table 10 shows that our model continues to outperform baselines with an average KL divergence of 0.1285. Even in the most visually degraded scenarios, such as those shown in Sequence 0018 and Sequence 0020, our method maintains stability with divergence values well below those of the CLIP and LoFTR-based baselines, which both exceed 1.2. Table 11 also reinforces this finding since our model maintains an average Pearson correlation of 0.682, again doubling the performance of the best-performing baseline (CLIP-based + Jaccard: 0.374). In sequences where lighting and visibility are heavily impaired, the

baselines fail to produce accurate spatial distributions, resulting in highly noisy predictions. The LoFTR method, which depends on keypoint correspondences, particularly struggles in degraded conditions and has the lowest average correlation (0.258).

Table 10: KL divergence between predicted anomaly scores and ground-truth pixel-change maps under rain and fog ($\alpha = \beta = 0.5$).

Method	0001	0002	0006	0018	0020	Avg.
Ours	0.0026	0.0006	0.0591	0.2532	0.3272	**0.1285**
CLIP-based + Jaccard	0.2781	0.1662	1.3382	1.4395	1.2087	0.8861
LoFTR-based + Jaccard	0.3987	0.3555	0.0614	1.4780	1.3566	0.7300

Table 11: Pearson correlation between predicted anomaly maps and ground truth under rain and fog ($\alpha = \beta = 0.5$).

Method	0001	0002	0006	0018	0020	Avg.
Ours	0.830	0.780	0.471	0.646	0.685	**0.682**
CLIP-based + Jaccard	0.460	0.628	0.288	0.215	0.279	0.374
LoFTR-based + Jaccard	0.278	0.279	0.202	0.214	0.319	0.258

IV. CONCLUSIONS

We present a unified vision-LiDAR framework for real-time semantic anomaly detection in urban maps that achieves state-of-the-art performance under challenging environmental conditions. Our key contributions include: (1) multi-modal sensor fusion with KL divergence-based consistency monitoring, (2) zero-shot semantic interpretation using vision-language models, and (3) robust performance in adverse weather conditions through adaptive sensor weighting. Our approach achieves state-of-the-art performance in all metrics, with high accuracy in map anomaly detection, spatial consistency, and resilience under degraded conditions. Moreover, our method exhibited high-quality spatial localization and strong temporal alignment across sequences. The added sensor fusion strategy further enhanced performance in degraded settings, improving map change detection rates. These results collectively highlight the effectiveness of our approach in capturing both spatial and temporal aspects of semantic drift, making it a promising solution for scalable and real-time map integrity monitoring in autonomous navigation systems. Future directions include integrating temporal cues and extending the approach to real-world datasets beyond simulation.

REFERENCES

Blanch, J., Walter, T., Enge, P., Pervan, B., Gratton, L., Fernandez, F., Dellago, R., & Rippl, M. (2015). Advanced raim user algorithm description: Integrity support message processing, fault detection, exclusion, and protection level calculation. *NAVIGATION, Journal of the Institute of Navigation*, *62*(4), 283–312. https://doi.org/10.1002/navi.97

Brown, R. G. (1992). A baseline raim scheme and a note on the equivalence of three raim methods. *NAVIGATION, Journal of the Institute of Navigation*, *39*(3), 301–316. https://doi.org/10.1002/j.2161-4296.1992.tb02278.x

Cabon, Y., de Charette, R., Perrotton, X., & Hesch, J. (2020). Virtual kitti 2. *arXiv preprint arXiv:2001.10773*.

Diehl, M., & Bertram, R. (2023). Navigation with uncertain map data for automated vehicles. *IEEE International Conference on Robotics and Automation (ICRA)*. https://link.springer.com/chapter/10.1007/978-3-658-34754-3_11

Ding, X., Zhu, J., Li, J., Li, S., Han, N., Chen, B., Cheng, W., Yao, H., Zhou, G., Veitch, A., et al. (2024). Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *European Conference on Computer Vision (ECCV)*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Houlsby, N., Konhauser, A., Kuenen, L., Wujek, M., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Gu, X., Ivanovic, B., & Pavone, M. (2023). Producing and leveraging online map uncertainty in trajectory prediction. *IEEE International Conference on Intelligent Vehicles*. https://arxiv.org/abs/2305.01545

Harithas, S. S., & Krishna, M. (2023). Urbanfly: Uncertainty-aware planning for navigation amongst high-rises with monocular visual-inertial slam maps. *ArXiv preprint*. https://arxiv.org/abs/2204.00865

Jaccard, P. (1912). *The distribution of the flora in the alpine zone* (Vol. 11).

Katyal, K., & Hager, G. D. (2023). Uncertainty-aware occupancy map prediction using generative networks for robot navigation. *IEEE Transactions on Robotics*. https://ieeexplore.ieee.org/document/9610137

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., & Girshick, R. (2023). Segment anything. *arXiv preprint*, *arXiv:2304.02643*. https://arxiv.org/abs/2304.02643

Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, *22*(1), 79–86. https://doi.org/10.1214/aoms/1177729694

Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. *International Conference on Learning Representations (ICLR)*.

Pairet, O., & Lahjanian, R. (2023). Online mapping and motion planning under uncertainty for safe navigation in unknown environments. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. https://ieeexplore.ieee.org/document/9610137

Paszke, A., Gross, S., Massa, F., Lerer, V., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, *32*.

Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 652–660. https://doi.org/10.1109/CVPR.2017.16

Sun, J., Shen, Z., Jiang, Y., Sarlin, T., Moorthy, D., & Quan, Y. (2021). Loftr: Detector-free local feature matching with transformers. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11892–11901.

Zou, L., & Sester, M. (2023). 3d uncertain implicit surface mapping using gaussian mixture models and gaussian processes. *ArXiv preprint*. https://arxiv.org/html/2403.07223v3