2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

HD Map Change Detection with Cross-Domain Deep Metric Learning

Minhyeok Heo, Jiwon Kim and Sujung Kim*

***Abstract*— High-definition (HD) maps are emerging as an essential tool for autonomous driving since they provide high-precision semantic information about the physical environment. To function as a reliable source of map information, HD maps must be constantly updated with changes that occur to the state of the road. In this paper, we propose a novel framework for HD map change detection that can be used to maintain an up-to-date HD map. More specifically, we design our HD map change detection algorithm based on deep metric learning, providing a unified framework that directly maps an input image to estimated probabilities of HD map changes. To reduce the discrepancy between input domains, i.e., camera image and HD map, we propose an effective learning scheme for metric space based on adversarial learning. Finally, we augment our framework with a pixel-level local change detector that specifies the region of changes in the image. We verify the effectiveness of our framework by evaluating it on a city-scale urban HD map dataset. Experimental results show that our method can robustly detect changes against noises due to dynamic objects and error in vehicle poses.**

I. Introduction

High-definition (HD) maps are machine-readable maps containing high-precision semantic information about the physical driving environment, such as lane geometry, road connectivity, traffic signs and road markers. With information from HD maps, more reliable decisions can be made in various stages of autonomous driving. For instance, planning and control relies on HD maps for identifying drivable routes and road surface geometry. In perception tasks such as object detection and tracking [1], [2], drivable area and lane information is utilized to reduce false alarms. Vehicle poses can be better localized by exploiting visual landmarks such as traffic signs, road markers, and lanes [3].

Although there were previous efforts to build HD maps from high-precision aerial images [4], most latest HD maps [2], [5] are created by driving around specialized mapping vehicles equipped with high-end sensors such as LIDAR, RTK-GNSS and IMU to obtain centimeter-level accuracy. However, such an expensive sensor suite inevitably increases the cost of mapping, limiting the number of mapping vehicles that can be concurrently dispatched.

Our physical road environment constantly changes, e.g., a new crosswalk may be installed, centerline may be moved while expanding the road, or the type of arrow marker may change due to changed route regulation. To function as a reliable source of map information for autonomous vehicles, HD maps must remain up-to-date with such changes. However, it is neither efficient nor feasible to operate specialized mapping vehicles on a daily basis just to identify possible changes.

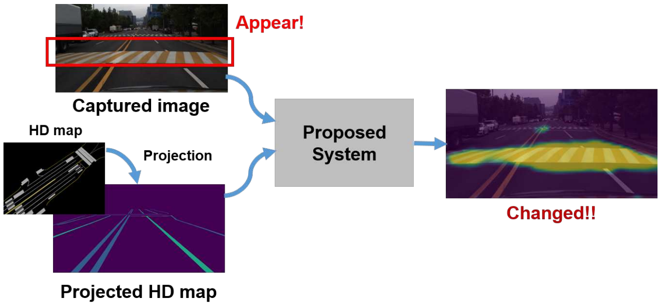


Fig. 1. Overview of the proposed system: Our HD map change detection framework estimates the pixel-level probability of changes by directly measuring the similarity between an input image and the corresponding HD map without elaborate intermediate steps typically needed by conventional methods.

In this paper, we aim to solve the HD map change detection problem with a low-cost sensor, i.e. camera. Conventional change detection methods typically take the following steps. First, predefined landmarks such as traffic signs, road markings and lanes are recognized from the input image. The algorithm is carefully designed to avoid distraction by various kinds of noise, e.g., occlusion by dynamic objects or viewpoint changes due to vehicle pose errors. Finally, classifiers are applied to determine whether or not a change has occurred.

In contrast, our approach directly maps an input image to the probability of HD map changes without such intermediate steps by exploiting deep metric learning. Deep metric learning employs neural networks to learn an embedding function that projects inputs into a feature space where the metric distance between them is an accurate measure of their semantic similarity. It has been successfully applied to various image-based applications [6]–[8]. In this work, we formulate the HD map change detection problem as a task of learning a metric space for measuring similarity between the camera image and HD map. To adjust for the domain gap between the two inputs, we utilize adversarial learning to transfer the inputs into a common feature space. We further augment our framework with a pixelwise local change detector that specifies the region of changes on the image, thereby facilitating subsequent map update. We demonstrate that our approach can successfully detect HD map changes by evaluating it on a city-scale urban dataset, and also show that the proposed method is robust against unwanted distractions such as dynamic objects and vehicle pose errors.

∗ indicates corresponding author.
Authors are affiliated in Autonomous Driving Group, NAVER LABS.
{heo.minhyeok, g1.kim, sujung.susanna.kim}@naverlabs.com

978-1-7281-6211-9/20/$31.00 ©2020 IEEE

The main contributions of the paper can be summarized as follows:

- We propose a novel framework for HD map change detection based on deep metric learning which only requires image-level supervision without entailing intermediate steps typically needed by conventional methods.
- We present an effective learning scheme for metric space based on adversarial learning to deal with discrepancy between the input domains, i.e., image and HD map.
- We further augment our framework with a pixel-level local change detector that specifies the region of changes on the image.

The rest of the paper is organized as follows. After reviewing related works in Section II, we describe in detail the proposed framework based on cross-domain deep metric learning in Section III. Experimental results are presented in Section IV, and we conclude the paper with a discussion of future work in Section V.

II. Related works

A. HD map

HD maps provide high-precision semantic information about the road environment in a machine-readable form, such as lane geometry, road connectivity, traffic signs, and road markers. This semantic information is typically represented as vector maps. As HD maps are emerging as an effective solution for autonomous driving, especially for vehicles with low-cost, low-precision sensor configurations, there have been increasing efforts in industries (e.g. HERE[1], TomTom[2], and NAVER LABS[3]) to develop HD maps in recent years.

However, since most of these HD maps are still in early stages of development, each map provides its data in a form specific to its own sensor setup, and there exists no standardized format across the industry. In addition, HD maps perform extensive post-processing on the raw sensor data to fuse it across multiple sensors and guarantee high-precision. This results in domain discrepancy between the HD map data and raw sensor data such as camera images, making it difficult to detect map changes that may have occurred.

B. Image based change detection

Image-based change detection was traditionally used in limited areas such as change detection for aerial images or visual inspection for manufacturing, where it is relatively easy to obtain changed datasets under identical configurations. Recently, with an increasing number of change detection datasets becoming available, image-based change detection algorithms have been developed for more diverse applications.

In [9], an algorithm based on superpixel segmentation is proposed to detect changes before and after a tsunami on a dataset of a city captured with a 360 camera. Park et al. [10] used CLEVR dataset [11] to create a synthetic dataset rendered with predefined objects and their positions, and proposed an algorithm that detects changes and produces the results as a caption. Most similar in spirit to our work, Revaud et al. [12] proposed an algorithm based on deep metric learning that detects visual changes of stores or shops for updating maps, and also provided a large dataset containing geo-localized indoor images of shopping centers.

C. Map change detection

In robotics, there exists an extensive amount of work on SLAM systems that build and maintain a 3D indoor map by continuously updating it, but maps for SLAM are fundamentally different from HD maps for autonomous driving. Change detection for HD maps is a relatively new problem that has been studied in only a few recent papers.

In [13], a SLAM framework is proposed that simultaneously performs localization and change detection by detecting HD map features from the sensor input. However, their system requires a sensor configuration similar to the one used for mapping, and it can only detect changes to a predefined set of map features. Pannen et al. [14] proposed a crowd-based map change detection algorithm that combines particle filter and boosted classifier to determine the probability of HD map changes. While it is a general framework that can be used with other types of classifiers, it also defines a known set of map features in advance, which makes it hard to handle unexpected types of changes to the road environment.

III. Proposed Method

We propose a novel HD map change detection framework that consists of three components, as shown in Fig. 2. We measure the similarity between RGB camera image and HD map using deep metric learning. To adjust for domain difference between the two inputs, we propose a learning scheme based on adversarial learning. In addition to image-level similarity, we also estimate a pixel-level probability of changes with a local change detector. In this section, we first present a formal definition of the HD map change detection problem we aim to solve. Then we describe each component of our proposed change detection framework in detail.

A. Problem formulation

Our framework takes 2 inputs, HD map and RGB image, and produces 2 outputs, a similarity score between the inputs and a pixel-level score map of changes. Let $O_i = \{I_i, p_i\}$ denote the output from ego-vehicle's sensors at location index i, where I_i and p_i are the RGB camera image and ego-vehicle's 6 DoF pose. We assume that the HD map is a set of map objects on the road surface, i.e. lanes and road markers, represented as $\phi_i = \{(P,c), \cdots\}$ at location i, where P is a set of 3D points for each object, and c is the object's class label. Then we can obtain the HD map mask M_i by

[1] https://www.here.com/platform/automotive-services/hd-maps

[2] https://www.tomtom.com/products/hd-map/

[3] https://hdmap.naverlabs.com/

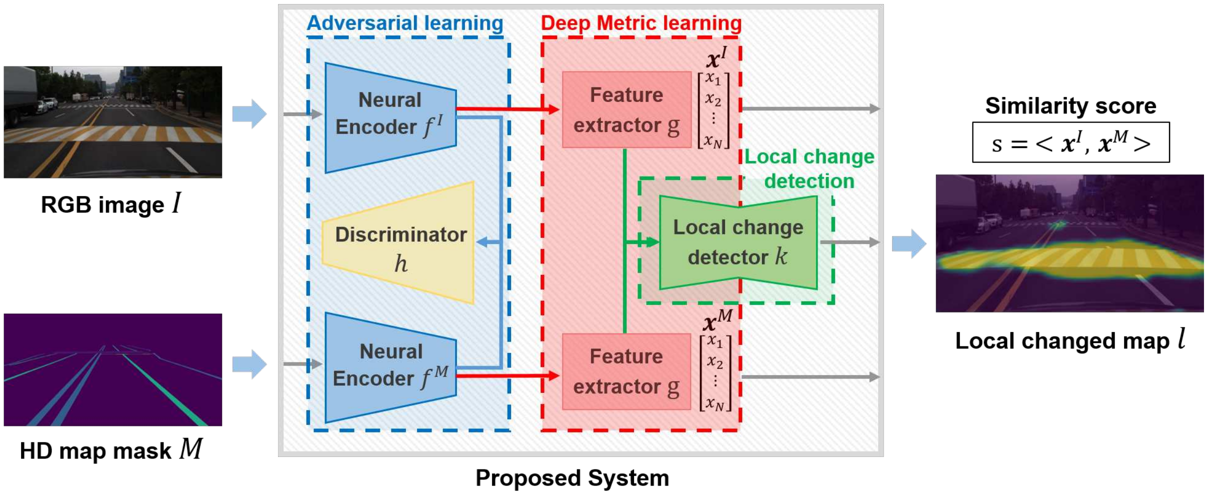


Fig. 2. Illustration of the proposed system. Our framework consists of three components as follows: i) Adversarial learning that reduces the discrepancy between input domains, i.e., image and HD map, ii) Metric learning that measures the similarity between domain-adjusted image and HD map, and iii) Local change detection that estimates a pixel-level probability of changes.

projecting all HD map objects within the camera view at current location p_i onto the camera image plane.

Our goal is to learn the similarity function $s : (I_i,\ M_i) \mapsto \mathbb{R}$ which measures the similarity between the image and the HD map mask. The output of s is high when no map change has occurred, and low otherwise. To facilitate map update by localizing the map objects where change has occurred, we also provide a pixel-level score map of change $l(x,y)$ which represents the probability of change at pixel location (x,y).

B. Deep Metric learning

Metric learning is an algorithm that learns an embedding function that projects data points into a feature space where the metric distance between them is an accurate measure of their semantic similarity. In deep metric learning, the embedding function is learned by a neural network architecture which unifies feature learning and metric learning into a joint learning framework. We design our HD map change detection algorithm based on deep metric learning, providing a unified framework that directly maps an input image to estimated probabilities of HD map changes, including those not defined in advance or unseen during training.

However, we cannot apply metric learning directly to our inputs I_i and M_i, due to domain difference between them. While I_i is an RGB image, M_i is a collection of HD map object labels for each pixel. To overcome this issue, we first apply deep neural encoders f^I, f^M to transfer I_i, M_i into the same feature space. We borrow the encoder structure in [15] which is widely used for domain adaptation. Each deep neural encoder outputs intermediate tensors lying in the same feature space, $f^I(I_i)$, and $f^M(M_i)$.

Now we can apply metric learning by extracting features from the intermediate tensors. Shared feature extractor g is a transfer function $g(I)$ that maps its input to an output tensor of channel dimension N. g consists of several convolution blocks, generalized mean pooling layer [16] and ℓ_2-normalized layer. We can define the similarity function s with f^I, f^M, and g as follows:

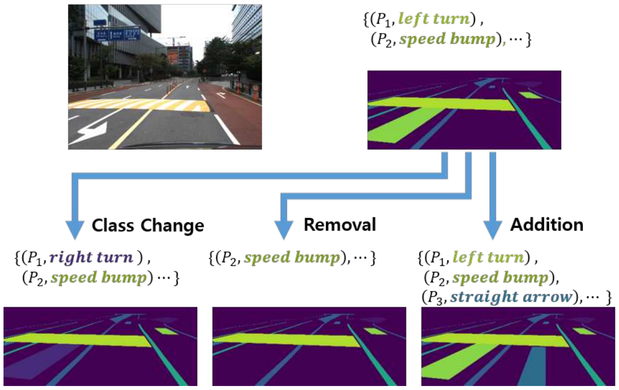

$\{(P_1, \mathit{left\ turn}),$
$(P_2, \mathit{speed\ bump}), \cdots\}$
Class Change
Removal
Addition
$\{(P_1, \mathit{right\ turn}),$
$(P_2, \mathit{speed\ bump}) \cdots\}$
$\{(P_2, \mathit{speed\ bump}), \cdots\}$
$\{(P_1, \mathit{left\ turn}),$
$(P_2, \mathit{speed\ bump}),$
$(P_3, \mathit{straight\ arrow}), \cdots\}$

Fig. 3. Examples of generated synthetic masks S. A mask was generated by changing randomly selected objects according to Removal, Addition, or Class change where an object was removed, added, or its class was changed, respectively.

$$s\left(I_i, M_i\right) = \left\langle g\left(f^I(I_i)\right), g\left(f^M(M_i)\right)\right\rangle, \tag{1}$$

where $\langle\cdot,\cdot\rangle$ denotes the inner product.

To train our similarity function s, we adopt triplet margin loss as our loss function:

$$L_{tri}(Q,P,N) = max(0, m - s(Q,P) + s(Q,N)), \tag{2}$$

where Q,P,N denote the query, positive and negative image, respectively. It enforces negative similarity $s(Q,N)$ to be smaller than positive similarity $s(Q,P)$ by a margin m.

We compose the training triplet T_i for each location i as follows:

$$T_i = (Q,\ P,\ N) = (I_i,\ M_i,\ M_i^n), \tag{3}$$

M_i^n indicates negative HD map mask:

$$M_i^n = \begin{cases} M_j & \text{if distance}\left(p_i,\ p_j\right) > 40\text{m} \\ S_i & \end{cases} \tag{4}$$

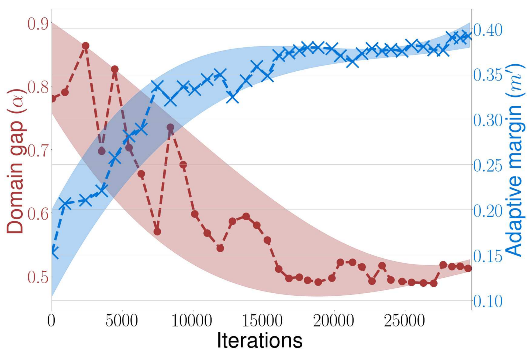


Fig. 4. Domain gap α and adaptive margin m' gradually converge as the training proceeds.

where S_i is a synthetic mask artificially generated from ϕ_i. Since changed samples are hard to collect, we generate changed data by adding, changing or removing HD map objects, as shown in Fig. 3. Manipulation of objects in HD map does not yield any image artifact that image manipulation may contain. Training proceeds by repeatedly sampling random triplets and computing the loss. If loss is non-zero, then loss gradient is computed and network weights are updated by back-propagation.

C. Adversarial learning for domain adaptation

To adjust for the domain gap between the two inputs f^I and f^M for the feature extractor g, we adopt adversarial learning [17]. We add a discriminator $h(.)$ that distinguishes between the two input domains by producing a scalar value between 0 (HD map mask) and 1 (RGB image), right after the neural encoder as shown in Fig. 2. Then we train f^I, f^M, g and h in alternating periods, using the neural encoders f^I, f^M as generators. Neural encoders are then trained to produce outputs with similar distributions that cannot be distinguished by the discriminator.

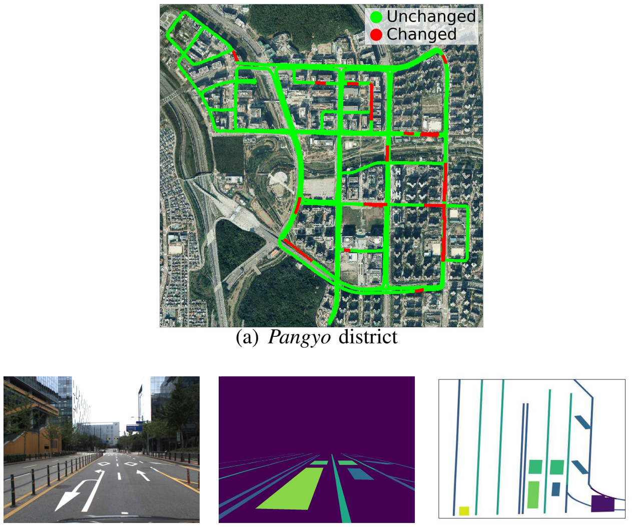
We also propose an adaptive margin for the triplet loss that adaptively controls the amount of margin depending on the current level of domain gap during training. During the initial training stage where domain gap is still large, metric training with triplet margin loss may not proceed well if the margin is too big. Therefore, we first define the amount of domain gap as follows:

$$\alpha = \Big\| \mathbb{E}[h(f^I(I_i))] - \mathbb{E}[h(f^M(M_i))] \Big\|_{\ell_1}. \tag{5}$$

As the discriminator h outputs a scalar value indicating input's closeness to a target domain, α represents the distance between the two domains with respect to the target domain.

While domain gap is initially being adjusted, we allow the algorithm to learn a relaxed metric space, and once domain gap has been sufficiently reduced, we want to finetune the metric space with a tighter margin. Therefore, we define the adaptive margin m' attenuated by the change in domain gap, as follows:

$$m' = m \cdot \exp\Big(- |\nabla \alpha| / \beta_{att} \Big), \tag{6}$$



(a) *Pangyo* district

(b) RGB image I (c) HD map mask M (d) HD map objects ϕ

Fig. 5. Example of NAVER LABS dataset. (a) shows the district *Pangyo*. Green and red lines denote the unchanged and changed regions. (b-d) show the captured image, projected HD map mask, HD map objects, respectively.

where β_{att} denotes the level of attenuation. In early stages of training, the domain gap will decrease quickly, resulting in a small margin m'. As the training proceeds, α will converge to a certain value and the margin will also converge to m, as shown in Fig. 4.

D. Local change detection

We further augment our framework with a local change detector k to learn a pixel-level score map of changes, $l(x,y)$. To provide output in the form of an image, we adopt an encoder-decoder structure for k. Features extracted from the second to the last layer of feature extractor g for an image I_i and a negative mask M_i^n are each normalized across the channel dimension, concatenated and used as input to k. Output of k is the estimated change score map l with the same spatial resolution as the input image, i.e., $l \in \mathbb{R}^{W \times H}$.

Normally, it is difficult to acquire ground truth data for change detection because changes are scarce and hard to annotate. However, since we synthetically generated changed samples, we can easily obtain the binary ground truth change map l_i^{gt} from the negative masks:

$$l_i^{gt}(x,y) = \mathbb{I}\left(M_i(x,y) \neq S_i(x,y)\right), \tag{7}$$

where $\mathbb{I}$ denotes the indicator function, $\mathbb{I}: X \mapsto \{True, False\}$.

The local change detector k is trained separately after we finish training the networks for image-level similarity s and freeze their weights. We use l_i^{gt} as ground truth to learn the change score map l by optimizing a binary cross-entropy loss.

IV. Experiments

A. Datasets

To train and validate our change detection framework, we use NAVER LABS HD map dataset which is publicly

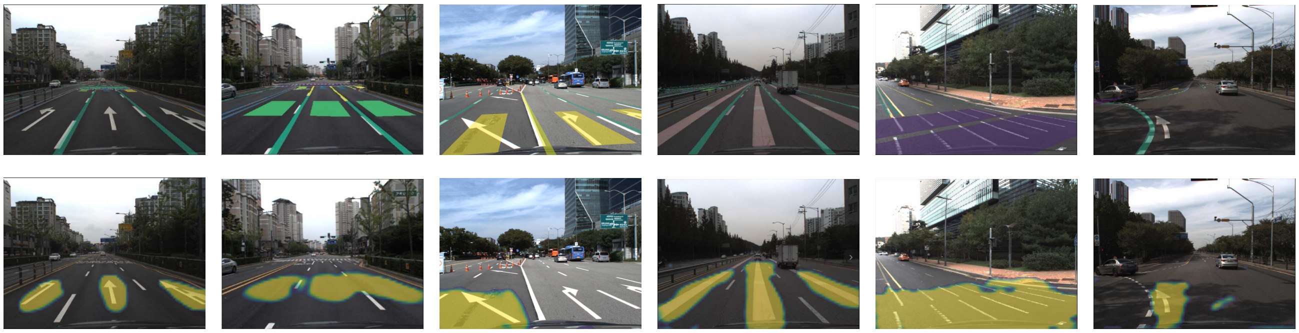

(a) Appearance (b) Disappearance (c) Class change (d) Undefined class (e) Undefined class (f) Undefined class

Fig. 6. The qualitative results of proposed algorithm. Top row shows HD map mask projected onto camera image, and bottom row shows results of local change detector. (a-b) Our algorithm can detect multiple changes even though we generated each synthetic sample with a single change. (c) Change of class label is correctly detected. (arrow type changed due to construction) Our algorithm can also detect changes that were not included in the training data: (d) colored guide line in the middle of lane, (e) unfinished painting of crosswalk, and (f) curved arrow.

TABLE I
SUMMARY OF HD MAP OBJECTS

Category	Marker type	Class index
Others	Others	0
Lane	White dotted line	1
	White solid line	2
	Yellow line	3
	Blue line	4
	Stop line	5
Arrow marker	Straight	6
	Left turn	7
	Right turn	8
	U-turn	9
	Prohibition	10
Information	No waiting zone	11
	Crosswalk	12
	Speed bump	13
	Text	14
	Speed limit	15
	Yield	16

available for research purposes. It provides HD semantic information about the road environment in cities of South Korea, including lane geometry, road connectivity and road markers. As it contains a large number of annotated map objects such as lanes and road markers where changes occur frequently, it is a reasonable choice for our experiments. Each lane and road marker in this dataset is georeferenced using high-precision RTK GPS, and annotated with polygon coordinates and class labels. We organize these objects into 17 classes as shown in Table I.

B. Implementation details and metrics

Network architecture: We borrow the encoder model and discriminator from [15] for the neural encoders f^I, f^M and discriminator h. We use one-hot encoding for the HD map mask M to represent the presence or absence of each class. Therefore, the dimension of the input channel for f^M is the number of HD map object classes $C = 17$. The shared feature extractor g consists of several residual blocks [18], and generalized mean pooling layer [16]. We set the output feature dimension of g as $N = 512$. The output feature of g is then ℓ_2-normalized for inner product calculation in Euclidean space. For the local change detector k, we adopt U-Net architecture [19] and perform bilinear upsampling to obtain change score map with the same spatial resolution as the input.

Training detail: We collected about 20K images with 6 DoF poses using a vehicle equipped with a low-cost camera and an RTK GNSS sensor, to acquire a training dataset with precise ground truth poses. We use images from the unchanged regions in *Pangyo* district, as shown in Fig. 5 (a). As described in Sec. III-B, negative samples are synthetically generated for metric learning.

We train f^I, f^M, g, h for 30 epochs using Adam solver with an initial learning rate of 0.001 and a batch size of 4. Then we train the local change detector k for 150 epochs with an initial learning rate of 0.0001. The rest of the hyperparameters are the same. The triplet loss margin m and attenuation level parameter β_{att} are set to 0.4 and 0.1, respectively. We apply horizontal flipping and color jittering for data augmentation. To simulate error in vehicle poses, we add a random noise $(< 1\text{m}, 5^\circ)$ during training.

Evaluation metrics: We evaluate our framework with two metrics, mAP$_r$ and mAP$_s$. mAP$_r$ is the mean AP score computed for each mask M_i among all HD map masks by querying I_i. This metric measures the effectiveness of our similarity function s at estimating the similarity between the input image and corresponding HD map mask. The second metric mAP$_s$ is the mean AP score computed for each mask M_i against all of its synthetic negative masks S_i by querying I_i. This metric measures the capability of our framework for detecting changes.

C. Quantitative results

For quantitative evaluation, we sample approximately 4K images of unchanged regions in *Sangam* district, and generate on average 40 synthetic changed masks for each image.

Ablation studies: As shown in Table II, adding adversarial learning and attenuated margin improves both metrics over the baseline, verifying their effectiveness. There is a

TABLE II

ABLATION STUDIES

Methods	mAP$_r$	mAP$_s$
Baseline	0.37	0.60
+ Adversarial learning	0.43	0.59
+ Attenuated margin	**0.51**	**0.71**

TABLE III

mAP$_s$ SCORES BY EACH TYPE OF CHANGE

Category	Type of change		
	Addition	*Removal*	*Class change*
Lane	-	0.86	0.63
Arrow marker	0.87	0.83	0.50
Information	0.91	0.90	0.57

clear advantage for using an adaptive margin for metric learning attenuated during early training phase over using a fixed margin. The benefit of adding adversarial learning is mainly exhibited by one of the two metrics mAP$_r$ because it is more closely related to domain adaptation, while mAP$_s$ is designed to reflect our framework's ability to detect changes. In our experiments, the corresponding HD map masks are retrieved at approximately 1.5–2 ranking on average. Note that AP scores can be interpreted as the inverse of the retrieval ranking among possible candidates.

Performances for each change type: Table III shows our framework's performance for each type of changes measured by mAP$_s$, the AP score for synthetic masks. Note that we did not create *Addition* changes for lanes. The mAP$_s$ scores show that our framework detects *Addition* and *Removal* with high accuracy, while also handling *Class change* reasonably well.

D. Qualitative results

Fig. 6 shows qualitative results of our change detection on real-world changed samples. Although we augmented each synthetically generated training sample with a single change, our framework is able to simultaneously detect multiple changes, as shown in Fig. 6 (a-b). More importantly, Fig. 6 (d)-(f) show that our algorithm can detect changes to HD map objects that are not defined in the HD map or not observed in the training data, demonstrating the generalization capability of our proposed framework.

We also demonstrate how our framework performs on a video sequence in Fig. 7 where (b) is a graph showing the similarity scores along the driving route from (c) to (h) in Fig. 7 (a). The similarity score drops in the region where the change actually occurs, visualized in pink. Also note that the similarity score changes continuously although our framework does not enforce temporal smoothness.

E. Discussions

Robustness against localization error: We verify the robustness of our proposed framework against the localization error by adding random noise. Our framework maintains approximately 80% of its accuracy when noise in the range of $(< 1m, 5°)$ is added, as shown in Fig. 8 (a). Furthermore, the local change detector k correctly localizes the changed region even when M_i is obtained from noisy pose, as shown in Fig. 8 (b). We believe that adding noise during training enabled our local change detector to learn the high-level structure of the HD map mask, thereby enhancing its robustness to error.

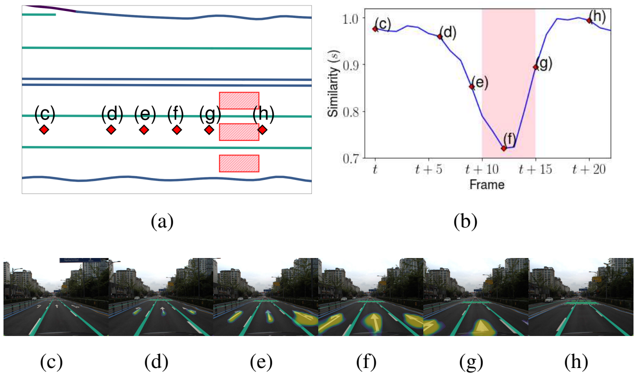


Fig. 7. Results on a video sequence from frame t to $t+x$. (a) visualizes the road layout along the driving route, (b) is a plot of the responses of similarity function s over time, (c-h) are the local change detection results.

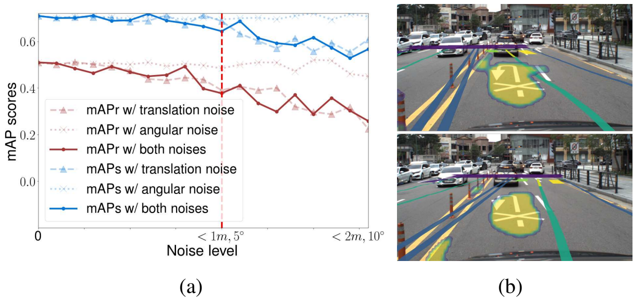


Fig. 8. The robustness of proposed framework to localization error. (a) Plot showing degradation of performance against increasing noise. (b) Regardless of localization error, local change detector can correctly localize the changed region.

Moving objects: In real-world driving scenarios, lanes and markers are frequently occluded by moving objects, i.e., vehicles and pedestrians. However, such occlusions should not be falsely detected as actual HD map changes. During generation of synthetic changed samples for training our framework, we discarded samples where the synthetic mask overlaps with moving objects, assisted by a semantic segmentation network trained on Cityscapes [20]. As shown in Fig. 9, our framework successfully ignores changes due to occlusion by moving objects, without directly handling moving objects at inference time.

Failure cases: In Fig. 10 (a), only part of the lanes are visible as the road is being repaved. Although this is a temporary state that does not correspond to meaningful HD map change, our algorithm still detects it as a change. Also, Fig. 10 (b) shows an example where only part of the road marker is visible within the image frame. Our algorithm fails to recognize the marker and detects it as a change. We believe this can be handled by extending our framework to use

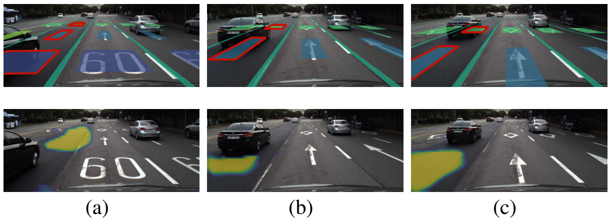

(a) (b) (c)

Fig. 9. Occlusion by moving objects. Changed region (top row, outlined in red) in the left lane is (a-b) partially occluded by moving object and (c) reappears. Our framework does not detect occlusion by moving object as a valid HD map change.

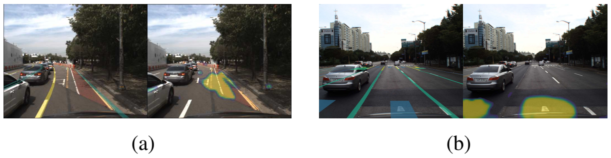

(a) (b)

Fig. 10. Failure cases where partially visible objects are falsely detected as changes. (a) Only part of the lanes are visible on a partially paved road. (b) Arrow truncated at the edge of the image frame.

multiple input frames to incorporate temporal information.

V. Conclusion and Future works

In this paper, we proposed a novel framework for detecting HD map changes with a typically affordable low-cost camera. By exploiting deep metric learning, our framework directly maps an input image to the estimated probability of HD map change, without elaborately designed intermediate steps for defining individual map objects and examining for possible changes. To adjust for the domain gap between the input camera image and HD map, we propose an effective learning scheme based on adversarial learning. Furthermore, we augment our framework with a local change detector that estimates a pixel-level probability of change on the image. We verify the effectiveness of our framework by evaluating it on a city-scale urban HD map dataset.

There are a few interesting directions for further research to extend our work. First of all, as the ultimate goal of change detection is to update the HD map with the recognized changes, we need to devise a way to identify the type of change and the map object where the change occurred. We would like to include vertical objects such as traffic signs and traffic lights as map objects. The scope of the objects is easily extendable with our framework, which is another advantage. Finally, we believe the performance of our map change algorithm can be further enhanced by aggregating information from multiple input frames over time as well as multiple vehicles for crowd-sourced map updates.

References

[1] B. Yang, M. Liang, and R. Urtasun, "Hdnet: Exploiting hd maps for 3d object detection," in *Conference on Robot Learning (CoRL)*, 2018, pp. 146–155.

[2] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, *et al.*, "Argoverse: 3d tracking and forecasting with rich maps," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 8748–8757.

[3] W.-C. Ma, I. Tartavull, I. A. Bârsan, S. Wang, M. Bai, G. Mattyus, N. Homayounfar, S. K. Lakshmikanth, A. Pokrovsky, and R. Urtasun, "Exploiting sparse semantic hd maps for self-driving vehicle localization," in *2019 International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[4] G. Máttyus, S. Wang, S. Fidler, and R. Urtasun, "Hd maps: Fine-grained road segmentation by parsing ground and aerial images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3611–3619.

[5] R. Kesten, M. Usman, J. Houston, T. Pandya, K. Nadhamuni, A. Ferreira, M. Yuan, B. Low, A. Jain, P. Ondruska, S. Omari, S. Shah, A. Kulkarni, A. Kazakova, C. Tao, L. Platinsky, W. Jiang, and V. Shet, "Lyft level 5 av dataset 2019," 2019. [Online]. Available: https://level5.lyft.com/dataset/

[6] A. Gordo, J. Almazán, J. Revaud, and D. Larlus, "Deep image retrieval: Learning global representations for image search," in *European conference on computer vision*. Springer, 2016, pp. 241–257.

[7] F. Wang, W. Zuo, L. Lin, D. Zhang, and L. Zhang, "Joint learning of single-image and cross-image representations for person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1288–1296.

[8] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5297–5307.

[9] K. Sakurada and T. Okatani, "Change detection from a street image pair using cnn features and superpixel segmentation." in *BMVC*, 2015, pp. 61–1.

[10] D. H. Park, T. Darrell, and A. Rohrbach, "Robust change captioning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 4624–4633.

[11] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. Lawrence Zitnick, and R. Girshick, "Clevr: A diagnostic dataset for compositional language and elementary visual reasoning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2901–2910.

[12] J. Revaud, M. Heo, R. S. Rezende, C. You, and S.-G. Jeong, "Did it change? learning to detect point-of-interest changes for proactive map updates," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4086–4095.

[13] K. Jo, C. Kim, and M. Sunwoo, "Simultaneous localization and map change update for the high definition map-based autonomous driving car," *Sensors*, vol. 18, no. 9, p. 3145, 2018.

[14] D. Pannen, M. Liebner, and W. Burgard, "Hd map change detection with a boosted particle filter," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 2561–2567.

[15] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[16] F. Radenović, G. Tolias, and O. Chum, "Fine-tuning cnn image retrieval with no human annotation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 7, pp. 1655–1668, 2018.

[17] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[19] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[20] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.