CLOVER: Context-Aware Long-Term Object Viewpoint- and Environment- Invariant Representation Learning

Amanda Adkins ⓘ, *Member, IEEE*, Dongmyeong Lee, and Joydeep Biswas ⓘ, *Senior Member, IEEE*

Abstract—Mobile service robots can benefit from object-level understanding of their environments, including the ability to distinguish object instances and re-identify previously seen instances. Object re-identification is challenging across different viewpoints and in scenes with significant appearance variation arising from weather or lighting changes. Existing works on object re-identification either focus on specific classes or require foreground segmentation. Further, these methods, along with object re-identification datasets, have limited consideration of challenges such as outdoor scenes and illumination changes. To address this problem, we introduce CODa Re-ID: an in-the-wild object re-identification dataset containing 1,037,814 observations of 557 objects across 8 classes under diverse lighting conditions and viewpoints. Further, we propose CLOVER, a representation learning method for object observations that can distinguish between static object instances without requiring foreground segmentation. We also introduce MapCLOVER, a method for scalably summarizing CLOVER descriptors for use in object maps and matching new observations to summarized descriptors. Our results show that CLOVER achieves superior performance in static object re-identification under varying lighting conditions and viewpoint changes and can generalize to unseen instances and classes.

Index Terms—Deep learning for visual perception, data sets for robotic vision, semantic scene understanding.

I. INTRODUCTION

FOR robotic systems to be reliable long-term, they must understand the environment at the object level, distinguishing and re-identifying object instances despite variations in viewpoints and environmental conditions. Current vision systems often struggle with these variations. For example, object-based SLAM systems [1], [2], [3] use objects as landmarks; however, these methods primarily rely on geometric and semantic consistency, leading to object-level data association errors when there is trajectory drift and multiple instances of the same semantic class [4], [5].
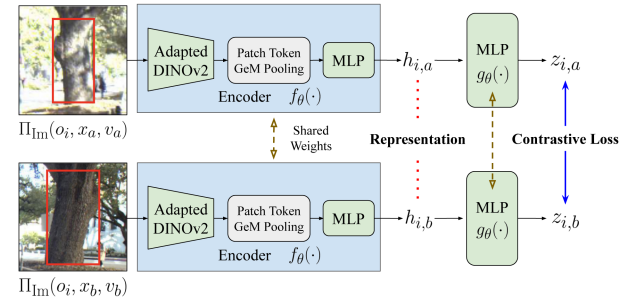
Fig. 1. Architecture of CLOVER. The tree in the input image patches is annotated with a bounding box (red) for visibility.

Object re-identification is the task of recognizing an object across multiple views. Learning-based methods address this by generating representations for images of objects and comparing a query object's representation against a database of representations for previously observed objects. However, most re-identification research is aimed at specific categories, such as humans [6], [7] and vehicles [8]. Existing approaches for general object re-identification [9], [10], [11] require foreground segmentation and focus on indoor scenes or single-object images, leaving challenges such as realistic outdoor images with significant lighting variation understudied.

In this paper, we introduce **CLOVER** (**C**ontext-aware **L**ong-term **O**bject **V**iewpoint- and **E**nvironment-Invariant **R**epresentation Learning), shown in Fig. 1, which extends the focus to *arbitrary* static objects. CLOVER leverages a foundation model backbone and the contextual information surrounding an object as an intrinsic property of the object itself and simplifies learning by integrating object context without segmentation. We further extend object re-identification by introducing a method to summarize object representations for use in object maps and enable comparison between summary representations and new observations. This method, called MapCLOVER, enables long-term scalability as well as improved robustness to viewpoint shifts.

Effective learning for object re-identification requires datasets capturing multiple instances from various viewpoints and environmental conditions. Existing general object datasets are limited to synthetic or indoor scenes or focus on the objects themselves. This limitation motivates the creation of the CODa Re-ID dataset, designed for general object re-identification research. Using the CODa Re-ID dataset, we evaluate existing general object re-identification methods [9], [10], [11] and our proposed approach.

In summary, this paper makes the following contributions: **(1)** We introduce the CODa Re-ID dataset, containing 1,037,814 observations of 557 objects across 8 semantic classes, captured under diverse viewpoints and environmental conditions in a campus setting. **(2)** We introduce CLOVER, a general object re-identification architecture that leverages foundation models and contextual information of objects using supervised contrastive learning [12]. **(3)** We introduce MapCLOVER for summarizing CLOVER descriptors for use in an object map and for matching incoming observation appearances against summary representations. **(4)** We compare our method with existing general object re-identification methods [9], [10], [11] on CODa Re-ID and ScanNet [13] and demonstrate improved performance on both seen and unseen instances during training, varying viewpoints and lighting conditions, and unseen classes.

II. RELATED WORKS

Object Re-identification and Image Retrieval: Object re-identification is the task of associating images of the same object instance captured under different viewpoints, weather, and lighting conditions. Representation learning [14], [15] has become standard for object re-identification. Viewpoint- and environmental-invariant feature representation learning remains a challenge due to the visual differences of the same instance when observed from varying perspectives. To address this challenge, several approaches have introduced viewpoint-aware training strategies [16], [17], [18]. However, these works and most re-identification methods focus on specific object classes, such as person [6], [7] and vehicle [8]. Few studies have extended re-identification to general objects. Foreground Feature Averaging [11] addresses this by averaging DINOv2 [19] features in the foreground and tested on varying viewpoints and lighting conditions. However, it is limited to single-object images and does not consider realistic outdoor environments. Re-OBJ [9] and Where Did I See It (WDISI) [10] exploit background information in addition to object appearance for re-identification. Re-OBJ jointly encodes foreground and background appearance with a CNN-based network, while WDISI uses a transformer. However, re-OBJ and WDISI only consider indoor scenes with limited lighting variation. Building on re-OBJ and WDISI, we investigate representation learning of objects by considering context as an intrinsic property of the object.

Object re-identification can be seen as a special case of image retrieval, where the goal is to retrieve similar images to a query from a database. Visual place recognition (VPR) is a related retrieval task aiming to find images from the same place as a query. SelaVPR [20] addresses VPR by adding trainable adapters to a frozen pre-trained foundation model, enabling it to leverage large-scale knowledge while focusing on features most relevant to place recognition. VPR systems typically store global feature vectors of a subset of observations and may include local features for fine-grained verification or pose estimation, though recent methods such as Placeformer [21] reduce storage by focusing on the most informative features. There is often no aggregation of observations from the same scene, requiring compromises between space and retaining useful information. Another approach for image retrieval aimed to address these challenges [22] by using instance centroids instead of individual observations. MapCLOVER extends this idea, opting for a more expressive summary by clustering representations for the same object, thus finding a balance between storing all descriptors,

naively selecting a subset, and oversimplifying with a single centroid.

Deep Metric Learning: Triplet loss [23], which minimizes the distance between representations for an anchor and positive sample and maximizes the distance from a negative sample, has been used in object re-identification [9], [10], [24]. Self-supervised [25] and supervised contrastive learning [12] improve representation learning by leveraging multiple negatives per anchor, enabling tighter clustering and stronger separation than traditional contrastive losses like triplet loss.

Datasets for Object Re-identification: For effective representation learning and benchmarking in object re-identification, it is crucial to have datasets that capture various objects with instance labels in real environments from diverse viewpoints and environmental conditions. ScanNet [13], Hyperism [26], and Amazon Berkeley Objects (ABO) Dataset [27] offer annotated objects but are restricted to indoor scenes, lacking real-world weather and lighting variations. The CUTE dataset [11] includes objects captured under different lighting conditions and poses for general object re-identification tasks. However, most objects in CUTE are captured within a studio setting and lack contextual information. In contrast, our dataset includes static objects of various categories captured in outdoor scenes from multiple viewpoints, encompassing diverse weather and lighting conditions, and providing richer contextual information for real-world object re-identification.

III. CODa RE-ID DATASET

In this section, we describe how we augment the UT Campus Object Dataset (CODa) [28] to generate the CODa Re-ID dataset, which contains globally unique *object instance* labels for object observations under various lighting conditions and viewpoints. This process can be applied to other datasets, such as the NCLT dataset [29] or user-generated datasets captured under varying conditions and viewpoints.

A. UT Campus Object Dataset (CODa)

UT CODa [28] is a robot perception dataset collected over 8.5 hours across 22 sequences. It has 1.3 million 3D bounding boxes for 53 semantic classes, along with RGB images and point clouds. The dataset includes repeated traversals of the same locations in varying weather and times of day, making CODa ideal for generating object re-identification data under diverse lighting conditions and viewpoints.

B. Dataset Curation Process

We leverage the geometric location of objects to generate globally unique object instance labels, assuming objects in the same location across views are the same instance. We focus on 8 static outdoor object classes with clear spatial separability (e.g., tree, not fence) and over 500 views per instance on average; per-class details are in Table I. Our curation process has four steps: 1) global trajectory alignment, 2) instance-level 3D bounding box annotation, 3) bounding box projection, and 4) segmentation and refinement.

Global Trajectory Alignment: As our object instance labeling relies on the location of static objects, maintaining global location consistency is necessary. After obtaining a trajectory for each sequence using LiDAR-inertial odometry [30], we aligned multiple trajectories of the target environment, shown in

TABLE I
THE NUMBER OF OBJECT INSTANCES AND OBSERVATIONS IN THE CODA RE-ID DATASET

Class	# instance	# img / instance	# img / instance (Scene Variation)			
			Sunny	Cloudy	Dark	Rainy
Tree	182	1917.9	852.8	591.6	229.7	243.8
Pole	208	2141.9	964.1	667.6	266.2	244.0
Bollard	37	2960.1	1246.8	923.8	478.7	310.8
Informational Sign	21	999.1	435.2	314.1	115.8	134.0
Traffic Sign	69	1001.0	365.9	373.1	147.2	114.7
Trash Can	31	1072.0	467.9	313.1	152.6	138.4
Fire Hydrant	5	591.8	246.2	171.2	87.4	87.0
Emergency phone	4	1868.5	617.5	748.8	265.0	237.3
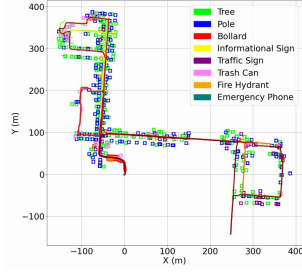
(a) Globally aligned trajectories
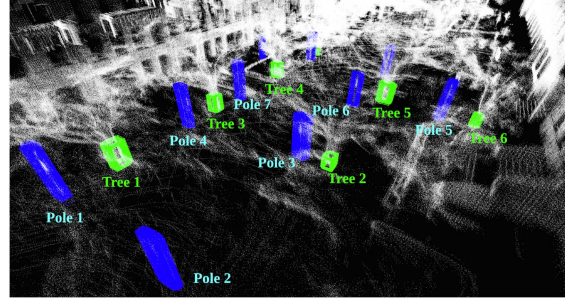
(b) 3D annotations for object instances

Fig. 2.　(a) Globally aligned trajectories. (b) Global 3D bounding boxes for tree (green) and pole (blue) from CODa.

Fig. 2(a), using interactive SLAM [31], applying manual loop closures and corrections.

Instance-level 3D Bounding Box Annotation: After obtaining globally consistent trajectories, we transformed the per-frame 3D bounding boxes from CODa into the global frame using the aligned sensor poses. These global-frame boxes were then clustered using DBSCAN [32] based on their 3D centroids, considering each cluster as a single instance. For each resulting instance, we computed a representative 3D bounding box by taking the medians of the centroids, dimensions, and rotations.[1] The processed 3D bounding boxes in the global map are shown in Fig. 2(b).

Projection of 3D Annotations: To generate 2D bounding box annotations with object instance labels for re-identification, we project the 3D annotations onto images. First, we query 3D annotations within a specific range around the camera pose. Next, we filter these annotations based on the camera's field of view and if point clouds are detected within the 3D bounding box at the image's timestamp. After filtering, we inscribe ellipsoids into the 3D bounding boxes and then project the ellipsoids onto the images. This reduces overestimation compared to 3D bounding boxes. For further details on ellipsoid projection, refer to [1] and [2].

Instance Segmentation and Refinement: Object instance foreground segmentation is required for some re-identification methods [9], [10], [11], so we augment CODa Re-ID with segmentation masks for greater utility. To generate masks, we used the Segment Anything Model (SAM) [33]. We queried the projected 2D bounding box to obtain a segmentation mask, then adjusted the bounding box to fit the mask contour, addressing projection misalignment.

---

[1] Global bounding box annotations could alternatively be annotated directly in the 3D global map.

IV. CLOVER

We introduce CLOVER, an object representation learning approach that includes context without requiring foreground segmentation. We discuss the problem formulation, image processing and loss function, and architecture for CLOVER.

A. Problem Formulation

We are interested in learning an object representation function f_θ with neural network weights θ such that the learned representations are unique to each object instance and invariant to environmental conditions and viewpoints. Let $o_i \in O$ be the ith object instance, $v_a \in V$ be the ath environmental condition, encompassing illumination and weather, and $x_a \in X, X \subseteq \text{SE}(3)$ be the viewpoint of the object, derived from the camera pose relative to the object. For each unique viewpoint and environmental condition, an observation of object instance o_i yields an image patch $I_{i,a} \in \mathcal{I}$. We model this relationship with the image patch projection operator $\Pi_{\text{Im}} : O \times V \times X \to \mathcal{I}$, which is treated as a closed-box system that produces observations $\mathcal{I}$ for objects based on their appearance, environmental condition, and observing viewpoint. The returned patch includes relevant context around the object. $f_\theta : \mathcal{I} \to H$ is an object representation function that takes an observation $I_{i,a}$ and generates a representation $h_{i,a} \in H$ for that observation. Formally, we aim for f_θ to satisfy the following conditions which encode these desired uniqueness and invariance properties:

$$\forall v_a, v_b \in V, x_a, x_b \in X : \tag{1}$$
$$f_\theta(\Pi_{\text{Im}}(o, x_a, v_a)) = f_\theta(\Pi_{\text{Im}}(o, x_b, v_b)),$$
$$\forall o_i, o_j \in O, i \neq j :$$
$$f_\theta(\Pi_{\text{Im}}(o_i, \cdot, \cdot)) \neq f_\theta(\Pi_{\text{Im}}(o_j, \cdot, \cdot)). \tag{2}$$

B. Context-Aware Object Representation Learning

Contextual Object Image Patches: To train f_θ, we need a dataset $\mathcal{D} = \{I_{j,a}\}$ consisting of observations with varying environmental conditions and viewpoints of different object instances, where j is an index for instance o_j and $I_{j,a}$ is an observed image patch of object o_j with viewpoint and environmental condition v_a, x_a. We divide this dataset into instance-specific partitions $\mathcal{D}_i = \{I_{j,a} \mid j = i\}$, each with only observations of object i. As we hypothesize that limited context can help distinguish static object instances, the input $I_{i,a}$ should include context around the object. Therefore, we obtain $\{I_{i,a}\}$ by first generating object detections as 2D bounding boxes with center $c_{i,a}$ and dimensions $(h_{i,a}, w_{i,a})$ for a full image I_{full} from a camera. For each bounding box, we add a fixed margin m to the larger of its dimensions to get $d_{i,a} = m + \max(w_{i,a}, h_{i,a})$. Each $I_{i,a}$ is then obtained by cropping image I_{full} into a $d_{i,a} \times d_{i,a}$ square centered at $c_{i,a}$. This enables inclusion of context in the image $I_{i,a}$ for which we are generating representation $h_{i,a}$.

To improve generalization, we supplement our training procedure for f_θ by applying several augmentations [12], [25], [34] to the input images $\{I_{i,a}\}$: color jittering, bounding box adjustment, crop size variation, image rotation, and random erasing. Color jittering alters color to improve lighting invariance. Bounding box position adjustment and crop size variation change the center and size of the image patch to simulate noisy bounding box detections. Image rotation simulates camera orientation variation with small random rotations, and random erasing emulates occlusions. We denote this augmented training set $\tilde{D}$.

Supervised Contrastive Learning: We cannot learn a function that exactly satisfies (1) and (2), so we instead aim to learn parameters θ such that

$$\forall v_a, v_b \in V, x_a, x_b \in X, o_i, o_j \in O, i \neq j:$$
$$s(f_\theta(\Pi_{\text{Im}}(o_i, x_a, v_a)), f_\theta(\Pi_{\text{Im}}(o_i, x_b, v_b)))$$
$$\gg s(f_\theta(\Pi_{\text{Im}}(o_i, \cdot, \cdot)), f_\theta(\Pi_{\text{Im}}(o_j, \cdot, \cdot))), \quad (3)$$

where $s(\cdot, \cdot)$ computes the representation similarity, with higher outputs indicating greater similarity. To encourage similarity between representations for the same object and uniqueness across objects, we train f_θ with the augmented dataset $\tilde{\mathcal{D}}$ using Supervised Contrastive Loss [12], which enables negative information from the full batch, as opposed to triplet loss [23], which has only one negative sample per positive. f_θ is a neural network encoder. Following [12], [25], [35], [36], we append a projection head g_θ to the encoder f_θ to generate the embeddings $z_{i,a} = g(h_{i,a})$ used in our loss. This projection head g_θ maps representations to the smaller latent space where contrastive loss is applied. Our loss with temperature τ is thus given by

$$\mathcal{L} = \sum_{I_{i,a} \in \mathcal{I}} \frac{-1}{|\tilde{\mathcal{D}}_i|} \sum_{I_{i,b} \in \tilde{\mathcal{D}}_i, a \neq b} \log \frac{\exp\left(z_{i,a}^\top z_{i,b}/\tau\right)}{\sum_{I_{j,c} \in \tilde{\mathcal{D}}} \exp\left(z_{i,a}^\top z_{j,c}/\tau\right)}. \quad (4)$$

C. Network Architecture

We model f_θ as a neural network encoder, and define a separate projection head for contrastive learning. The encoder is derived from SelaVPR [20], which augments a frozen DINOv2 [19] backbone with learnable adapter blocks and applies GeM pooling to the generated patch tokens. To enhance encoder

expressivity, we extend the adapted DINOv2 backbone with a multi-layer perceptron (MLP), which forms the final stage of f_θ. This design is motivated by prior work [37], [38] showing that additional layers can improve feature quality and task performance. This dual adaptation—adapter blocks for task-specific tuning and an MLP for feature refinement—allows CLOVER to leverage DINOv2's large-scale pre-training while balancing adaptability with catastrophic forgetting risk. Empirically, this combination outperforms either component used in isolation. The projection head is an MLP with a single hidden layer to obtain $z_{i,a} = g(h_{i,a})$. After training, we discard the projection head g_θ and use the encoder f_θ for object re-identification. The network architecture is shown in Fig. 1.

V. MAPCLOVER

A. Problem Formulation

To enable long-term scalable matching using CLOVER descriptors, we cannot simply maintain representations for all previous object observations $\{h_{i,\cdot}\}$ for each object instance i. Therefore, we must develop a representation summary function f_{sum} and a summary similarity function s_{sum}.

The goal of the representation summary function f_{sum} is to produce a summary representation y_i that is bounded in size and concisely summarizes previous observations $\{h_{i,a_k}\}$ of an object instance i in various viewing conditions $\{a_k\}$. This has the form $y_i = f_{\text{sum}}(\{h_{i,a_k}\})$. The summary similarity function s_{sum} should produce a similarity score that is higher when evaluating a representation $h_{i,\cdot}$ and summary representation y_i corresponding to the same object instance i than when computed on a representation $h_{j,\cdot}$ and summary representation y_i corresponding to different instances i and j. Formally, the summary similarity function $s_{\text{sum}}(h, y)$ should satisfy the following property:

$$s_{\text{sum}}(h_{i,\cdot}, y_i) \gg s_{\text{sum}}(h_{j,\cdot}, y_i) \forall i, j, i \neq j. \quad (5)$$

B. Representation Summarization and Similarity Evaluation

Representation Summary Function: The representation summary function should enable conciseness as well as tolerance of remaining variance resulting from differing viewing conditions not alleviated by CLOVER descriptor training. Averaging descriptors is not suitable in cases of notable variance, such as opposing viewpoints, as an averaged descriptor could fail to match either viewpoint well.

To account for remaining variation in descriptors $h_{i,\cdot}$, we define the summary representation y_i as a representative set of vectors $\{r_i \in H\}$ in the same space as CLOVER descriptors. Using this representative set avoids the unimodal assumptions of an averaged representation, enables intra-object variation to be captured, and prevents having to reconcile drastically different viewing conditions. Our summarization function $f_{\text{sum}}(\{h_{i,a_k}\})$ performs k-means clustering on all observations $\{h_{i,a_k}\}$ for an object instance i. Cluster centers are used as the representative set $\{r_i\} \in y_i$.

Summary Similarity Function: We define s_{sum} as

$$s_{\text{sum}}(h, y_i) = \max_{r \in y_i} s(h, r), \quad (6)$$

where, as noted earlier, $s(\cdot, \cdot)$ computes the representation similarity. We use cosine similarity to align with the CLOVER training objective. This summary similarity function computes the similarity of the query representation h with each entry r

in the representative set and uses the highest similarity as the summary similarity.

VI. EXPERIMENTS

We evaluate our method in real-world campus scenes with static objects. We conduct experiments on both observation-observation based matching using individual CLOVER representations and observation-map matching using summarized representations with MapCLOVER. Our CLOVER experiments address four key questions:

1) How invariant are representations generated by CLOVER to variations in viewpoint and lighting/weather?
2) How well does CLOVER generalize to unseen instances?
3) How well does CLOVER generalize to unseen classes?
4) What is the impact of each component of the formulation for CLOVER on its performance?

Additionally, we evaluate CLOVER on ScanNet [13]. Finally, we assess how reliably incoming observations are matched to summarized representations using MapCLOVER. CODa Re-ID, parameters, code, and extended results can be found at https://github.com/ut-amrl/CLOVER.

A. CLOVER Evaluation Setup

Datasets: To evaluate the performance of our proposed method, we trained and tested CLOVER with our CODa Re-ID dataset. The weather conditions for each object observation are defined by the sequence in which the observation was captured, categorized as sunny, dark, cloudy, or rainy. To test invariance, we created two groups of variations: illumination and viewpoint. For illumination, we generated two subsets based on the weather condition: similar illumination and different illumination. For viewpoint, for each query, we separated comparison observations into three grades of viewpoint difference difficulty based on the differences in the distance d of objects from the respective camera and the angle α between the object-to-camera rays. We denote the subsets as easy $(d \leq 10\,\mathrm{m}, \alpha \leq 15°)$, medium $(d \leq 30\,\mathrm{m}, \alpha \leq 90°)$, and hard $(d > 30\,\mathrm{m}, \alpha > 90°)$. We used three classes (tree, pole, and bollard) for training across all experiments. The dataset split varied depending on the experiment, which will be explained in the following sections.

Evaluation Metrics: We quantified the performance of object re-identification using mean average precision (mAP) and top-1/top-5 accuracy, which measure the percentage of queries with a correct match in the top-1 or top-5 retrieved items. We also display the average number of matches for a query and the average retrieval set size to convey task difficulty. Object re-identification is performed among observations of objects of the same class as the query object observation, based on the assumption that an object detector can provide its class. We exclude observations of the same object from the retrieval set if they are from the same sequence as the query or if the environmental conditions or viewpoints between the query and the observations of the same object do not meet the illumination/viewpoint test case criteria.

Implementation Details: For image patches, we use a 10-pixel margin and resize the cropped images to 224×224 pixels. Following SelaVPR [20], we use the DINOv2 [19] model based on VIT-L14. Our encoder MLP matches the DINOv2 output dimension of 1024 and uses one hidden layer, and our projection head MLP has an output dimension of 128 and also uses one

hidden layer. We use 0.07 for our loss temperature and employ the SGD optimizer with a learning rate of 0.001. The learning rate is scheduled using CosineAnnealingLR [39]. We use early stopping, halting training if there is no validation mAP improvement for 10 consecutive epochs. All experiments were conducted on an Nvidia RTX A6000 GPU. CLOVER has 361 million parameters and achieves an average inference time of 15 ms per 32-image batch. In our experiments, unless otherwise noted, bounding boxes are from CODa Re-ID annotations to enable instance labeling, though CLOVER works with bounding boxes from any off-the-shelf object detector.

Baselines: We compare our method with re-OBJ [9], WDISI [10], and FFA [11]. These all require segmentation, for which we used the masks provided in our CODa Re-ID dataset. We compare representations with cosine similarity for CLOVER and FFA, and L_2 distance for re-OBJ and WDISI, as their losses use this metric. We implemented re-OBJ and WDISI ourselves, as code was not available.

B. CLOVER Experimental Results

Lighting, Weather, and Viewpoint Invariance: To evaluate how invariant the representations are to viewpoint and lighting/weather variations, we split CODa Re-ID into a training set and a test set, with 7 and 13 sequences respectively. This is similar to a real-world deployment procedure: initial data is collected from the target environment and used to train the model for subsequent deployments. We narrow the retrieval set for each observation to restrict the viewpoint similarity and environmental similarity as described above to reflect robustness to the described variation.

Table II shows the results for CLOVER and the comparison algorithms when the retrieval set for each observation contains all lighting conditions, only similar illumination, only different illumination, and varying viewpoint difficulties. Cases with easy viewpoint differences have the highest performance, with performance degrading slightly as the degree of viewpoint change increases. This indicates that viewpoint shift is more challenging for all algorithms than lighting variation, but that CLOVER is still quite robust to significant viewpoint shifts. We additionally test using bounding boxes generated by YOLOv11 [40], obtaining instance labels from overlapping ground truth bounding boxes. The mAP of all approaches using the detections is shown in Table V; CLOVER again has the best performance.[2]

Unseen Instance Generalization: To test CLOVER's generalization to unseen instances, we split the CODa Re-ID dataset by region such that test set instances are distinct from the training set. Table III shows the results when evaluated on unseen instances in varying lighting conditions and viewpoint differences. Though the results are somewhat less accurate than for the sequence-based split in Table II, particularly as the viewpoint difference increases, our method outperforms all baselines. This is notable as FFA [11] is not learned, so it does not benefit from having seen instances in training.

Unseen Class Generalization: We test CLOVER's and the comparison algorithms' ability to generalize to unseen classes by evaluating on the three most common classes not used in training (informational sign, traffic sign, and trash can). We use weights from training using the region-based split. Table IV shows results for all methods. Despite using classes not seen

---

[2] Full results can be found on our website.

TABLE II
EFFECT OF ILLUMINATION AND VIEWPOINT CHANGE ON OBJECT INSTANCE RETRIEVAL FOR OBJECT INSTANCES SEEN IN TRAINING

| | All | | | Similar Illumination | | | Different Illumination | | | Viewpoint Change | | | | | | | | |
| | | | | | | | | | | Easy | | | Medium | | | Hard | | |
	mAP	top-1	top-5	mAP	top-1	top-5	mAP	top-1	top-5	mAP	top-1	top-5	mAP	top-1	top-5	mAP	top-1	top-5
re-OBJ	0.371	0.563	0.712	0.352	0.479	0.642	0.314	0.445	0.609	0.477	0.576	0.708	0.204	0.230	0.284	0.194	0.207	0.419
WDISI	0.495	0.690	0.821	0.480	0.620	0.747	0.437	0.584	0.730	0.631	0.718	0.831	0.317	0.356	0.533	0.303	0.360	0.573
FFA	0.236	0.506	0.649	0.215	0.391	0.516	0.204	0.385	0.513	0.393	0.537	0.666	0.100	0.114	0.201	0.083	0.081	0.166
CLOVER (ours)	**0.811**	**0.876**	**0.939**	**0.802**	**0.838**	**0.915**	**0.765**	**0.828**	**0.903**	**0.873**	**0.890**	**0.943**	**0.712**	**0.734**	**0.843**	**0.731**	**0.777**	**0.892**
Avg Matches/Ref Set	15.98 / 1221			8.71 / 1205			9.97 / 1213			6.27 / 1204			5.67 / 1226			7.62 / 1193		

TABLE III
EFFECT OF ILLUMINATION AND VIEWPOINT CHANGE ON OBJECT INSTANCE RETRIEVAL FOR OBJECT INSTANCES NOT SEEN IN TRAINING

| | All | | | Similar Illumination | | | Different Illumination | | | Viewpoint Change | | | | | | | | |
| | | | | | | | | | | Easy | | | Medium | | | Hard | | |
	mAP	top-1	top-5	mAP	top-1	top-5	mAP	top-1	top-5	mAP	top-1	top-5	mAP	top-1	top-5	mAP	top-1	top-5
re-OBJ	0.210	0.453	0.617	0.247	0.402	0.520	0.183	0.381	0.552	0.347	0.482	0.629	0.070	0.067	0.178	0.056	0.052	0.127
WDISI	0.288	0.580	0.706	0.364	0.540	0.658	0.248	0.493	0.635	0.480	0.618	0.731	0.130	0.143	0.255	0.083	0.089	0.224
FFA	0.294	0.567	0.707	0.339	0.492	0.617	0.259	0.497	0.650	0.446	0.591	0.724	0.126	0.144	0.239	0.121	0.136	0.257
CLOVER (ours)	**0.519**	**0.800**	**0.882**	**0.523**	**0.682**	**0.766**	**0.485**	**0.758**	**0.855**	**0.750**	**0.846**	**0.914**	**0.376**	**0.433**	**0.608**	**0.238**	**0.261**	**0.448**
Avg Matches/Ref Set	13.91 / 618			4.35 / 616			11.52 / 617			5.24 / 613			4.15 / 612			7.13 / 620		

TABLE IV
PERFORMANCE OF CLOVER AND COMPARISON METHODS ON CLASSES NOT SEEN IN TRAINING

| | Informational Sign | | | Traffic Sign | | | Trash Can | | |
	mAP	top-1	top-5	mAP	top-1	top-5	mAP	top-1	top-5
re-OBJ	0.602	0.817	0.924	0.343	0.607	0.708	0.389	0.678	0.822
WDISI	0.747	0.942	0.969	0.433	0.719	0.815	0.510	0.778	0.875
FFA	0.735	0.888	0.929	0.391	0.701	0.765	0.505	0.788	0.859
CLOVER (ours)	**0.897**	**0.964**	**0.978**	**0.606**	**0.840**	**0.880**	**0.703**	**0.850**	**0.922**
Avg Matches/Ref Set	22.95 / 226			23.75 / 689			22.80 / 320		

TABLE V
RE-IDENTIFICATION PERFORMANCE (mAP) ON SEEN INSTANCES USING
YOLOV11 GENERATED BOUNDING BOXES

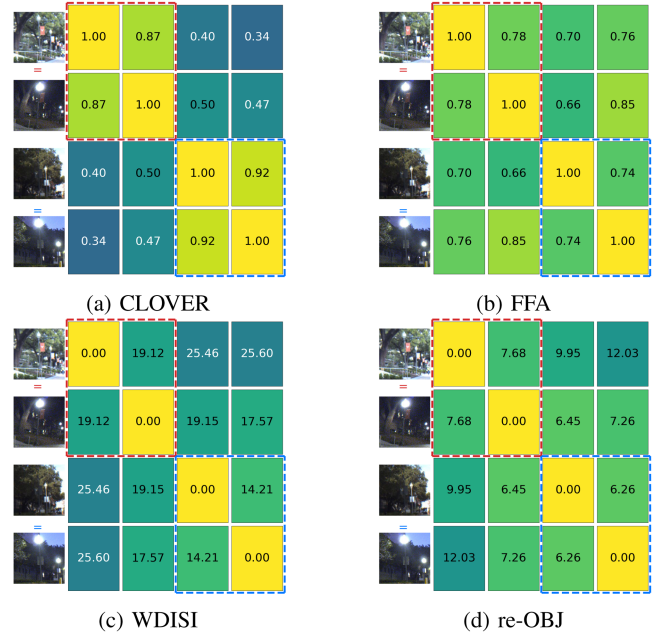|	All	Sim. Illum.	Diff. Illum.
re-OBJ	0.452	0.434	0.381
WDISI	0.585	0.576	0.516
FFA	0.274	0.258	0.230
CLOVER (ours)	**0.882**	**0.886**	**0.846**
Avg Matches/Ref Set	14.76 / 956	8.32 / 942	9.16 / 949

Fig. 3. Qualitative performance on two pairs of images, each from the same object. Values give similarity between images: higher cosine similarity (CLOVER, FFA) and lower L_2 distance (WDISI, re-OBJ) indicate higher matching confidence.

in training, performance is still good, with CLOVER outperforming all methods, demonstrating generalization to unseen classes. In some cases, CLOVER and other methods have better results for unseen classes than for seen classes (Table II). We hypothesize that this is due to a greater number of matches and more visual distinctiveness.

Qualitative Results: Fig. 3 shows the similarity between observations of two instances in the first two and last two rows, respectively, using cosine similarity for CLOVER and FFA [11] and L_2 distance for WDISI [10] and re-OBJ [9]. An approach correctly discriminating between object instances would have better similarity (higher for cosine, lower for distance) in the 2x2 blocks along the diagonal and worse similarity in the off-diagonal blocks. For these observations, CLOVER is the only approach that has strictly better similarity for the same instance than different instances.

Ablations: We conduct ablations with the split used in Table II to understand the impact of components and parameters on CLOVER's performance. *No Aug.* tests CLOVER with no training set augmentations. We examine the impact of context by varying the margin. To understand the role of the DINOv2 backbone, we test a frozen version without adapters and an unfrozen version where the backbone weights are updated

during training. We also explore architectural modifications by removing the MLP applied to aggregated DINOv2 features. The final variant uses triplet loss [23]. Table VI shows the performance of CLOVER and these variations, with CLOVER outperforming all others except sometimes the 75 pixel margin

TABLE VI
RESULTS FOR CLOVER AND ABLATED VERSIONS

	All			Similar Illumination			Different Illumination		
	mAP	top-1	top-5	mAP	top-1	top-5	mAP	top-1	top-5
CLOVER (ours)	**0.811**	**0.876**	0.939	**0.802**	**0.838**	**0.915**	**0.765**	**0.828**	0.903
No Aug.	0.787	0.872	0.937	0.768	0.829	0.910	0.751	0.826	0.902
Fixed Margin=0	0.766	0.853	0.920	0.757	0.814	0.896	0.731	0.806	0.881
Fixed Margin=75	0.803	0.842	**0.947**	0.783	0.798	**0.915**	0.756	0.793	**0.922**
Frozen DINOv2 (no adapters)	0.410	0.668	0.818	0.364	0.538	0.693	0.358	0.547	0.699
Unfrozen DINOv2	0.496	0.625	0.737	0.508	0.605	0.707	0.445	0.539	0.651
No Encoder MLP	0.765	0.862	0.928	0.746	0.810	0.894	0.720	0.805	0.885
Triplet Loss	0.563	0.701	0.832	0.541	0.630	0.771	0.498	0.597	0.738
Avg Matches/Ref Set	15.98 / 1221			8.71 / 1205			9.97 / 1213		

TABLE VII
RE-IDENTIFICATION PERFORMANCE ON SCANNET

	mAP	top-1	top-5
re-OBJ	0.466	0.723	0.853
WDISI	0.533	0.770	0.891
FFA	0.432	0.662	0.819
CLOVER (ours)	**0.658**	**0.815**	**0.931**
Avg Matches/Ref Set	7.70 / 7273		

TABLE VIII
PERFORMANCE OF MAPCLOVER AND BASELINES. THE AVERAGE NUMBER OF
OBSERVATIONS PER OBJECT THAT FORM THE MAP IS 8.99, AND THE AVERAGE
RETRIEVAL SET SIZE IS 66.7.

Summary/Sim Method	Size	Top-1	Top-5	Top-10
Average	-	0.764	0.908	0.942
Random Set, Max Sim	5	0.738	0.899	0.938
Clustering, Avg Sim	5	0.644	0.856	0.915
MapCLOVER (Clust, Max)	5	**0.803**	**0.917**	**0.948**
Random Set, Max Sim	10	0.796	0.919	0.949
Clustering, Avg Sim	10	0.651	0.864	0.920
MapCLOVER (Clust, Max)	10	**0.811**	**0.921**	**0.950**

variant, highlighting the importance of the proposed components and parameters. The triplet loss results may also partly explain the worse performance of WDISI [10] and re-OBJ [9], which use triplet loss.

ScanNet: To demonstrate the broader efficacy of our method, we also test on the ScanNet dataset [13] using 240 scenes. Scan-Net does not have cross-trajectory instance labels, so we train and test on disjoint subsets of the scenes, with one trajectory per included scene. All observations for each object come from the same trajectory and thus have the same lighting conditions. The test set contains 1,140 instances from 60 scenes. Table VII shows the mAP and top-1/top-5 accuracy of CLOVER and comparison approaches. CLOVER again outperforms all other approaches, though the improvement relative to other approaches is reduced due to the lack of challenging lighting variations. These results demonstrate that our training procedure generalizes.

C. MapCLOVER Evaluation Setup

To evaluate the reliability of MapCLOVER in matching incoming object observation appearance representations to summarized object appearance representations, we divide the test set of the region-based split used in Table III into map sequences and query sequences, averaging across 30 different train/test sequence splits. We create summary descriptors for each object instance using CLOVER descriptors for observations in the map sequences and use descriptors for query sequence observations as queries. We compare MapCLOVER against two baselines: (1) averaging all map sequence CLOVER descriptors to get a summarized representation, similar to [22], and (2) randomly selecting descriptors to form a representative set. We also evaluate using the average similarity as our summary similarity function, instead of the maximum. Additionally, we test representative set sizes 5 and 10 to understand the tradeoff between accuracy and scalability. Performance is assessed using top-1, top-5, and top-10 accuracy.

D. MapCLOVER Experimental Results

Table VIII shows the top-k metrics for random representative set, observation averaging, and MapCLOVER, using both average and maximum similarity for comparison. MapCLOVER has notably improved accuracy compared to randomly choosing a representative set or averaging. Further, using the maximum similarity as proposed yields better accuracy than average similarity. Additionally, accuracy improves as the representative set size increases, but MapCLOVER's advantage over other approaches is greatest at smaller set sizes, demonstrating Map-CLOVER's ability to enable concise representations with minimal accuracy impact.

VII. CONCLUSION

In this letter, we propose CLOVER to generate representations for object observations, enabling discrimination between object instances with improved environment condition and viewpoint invariance compared to existing methods. MapCLOVER builds upon CLOVER by providing methods to concisely augment an object map with summary appearance data and match object summaries with incoming observations. Our CODa Re-ID dataset includes observations in varying lighting conditions and viewpoints to enable further research into general object re-identification.

CLOVER, which leverages limited context to re-identify static objects, is not suitable for dynamic objects like humans and vehicles, as their backgrounds can change unpredictably. Future work could include utilizing CLOVER-generated representations to improve object data association in object-based SLAM for both short- and long-term changes. Finally, a joint matching framework to consider compatibility of both geometry and appearance representations for multiple object observations simultaneously could enable more robust association using CLOVER and MapCLOVER.

ACKNOWLEDGMENT

REFERENCES

[1] L. Nicholson, M. Milford, and N. Sünderhauf, "QuadricSLAM: Dual quadrics from object detections as landmarks in object-oriented SLAM," *IEEE Robot. Automat. Lett.*, vol. 4, no. 1, pp. 1–8, Jan. 2019.

[2] K. Ok, K. Liu, K. Frey, J. P. How, and N. Roy, "Robust object-based SLAM for high-speed autonomous navigation," in *Proc. 2019 Int. Conf. Robot. Automat.*, 2019, pp. 669–675.

[3] A. Adkins, T. Chen, and J. Biswas, "Obvi-SLAM: Long-term object-visual SLAM," *IEEE Robot. Automat. Lett.*, vol. 9, no. 3, pp. 2909–2916, Mar. 2024.

[4] K. J. Doherty, D. P. Baxter, E. Schneeweiss, and J. J. Leonard, "Probabilistic data association via mixture models for robust semantic SLAM," in *Proc. 2020 IEEE Int. Conf. Robot. Automat.*, 2020, pp. 1098–1104.

[5] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, "Probabilistic data association for semantic SLAM," in *Proc. 2017 IEEE Int. Conf. Robot. Automat.*, 2017, pp. 1722–1729.

[6] L. Zheng, Y. Yang, and A. G. Hauptmann, "Person re-identification: Past, present and future," 2016, *arXiv:1610.02984*.

[7] M. Ye, J. Shen, G. Lin, T. Xiang, L. Shao, and S. C. H. Hoi, "Deep learning for person re-identification: A survey and outlook," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 6, pp. 2872–2893, Jun. 2022.

[8] X. Liu, W. Liu, H. Ma, and H. Fu, "Large-scale vehicle re-identification in urban surveillance videos," in *Proc. 2016 IEEE Int. Conf. Multimedia Expo*, 2016, pp. 1–6.

[9] V. Bansal, S. James, and A. Del Bue, "re-OBJ: Jointly learning the foreground and background for object instance re-identification," in *Proc. 2019 Int. Conf. Image Anal. Process.*, 2019, pp. 402–413.

[10] V. Bansal, G. L. Foresti, and N. Martinel, "Where did i see it? Object instance re-identification with attention," in *Proc. 2021 IEEE/CVF Int. Conf. Comput. Vis. Workshops*, 2021, pp. 298–306.

[11] K. Kotar, S. Tian, H. -X. Yu, D. Yamins, and J. Wu, "Are these the same apple? Comparing images based on object intrinsics," in *Proc. Adv. Neural Inf. Process. Syst.*, 2024, vol. 36. [Online]. Available: https://openreview.net/forum?id=pu3sNlrgQr

[12] P. Khosla et al., "Supervised contrastive learning," in *Proc. Adv. Neural Inf. Process. Syst.*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020, vol. 33, pp. 18661–18673.

[13] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *Proc. 2017 IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5828–5839.

[14] F. Xiong, M. Gou, O. Camps, and M. Sznaier, "Person re-identification using kernel-based metric learning methods," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 1–16.

[15] M. Hirzer, P. M. Roth, M. Köstinger, and H. Bischof, "Relaxed pairwise learned metric for person re-identification," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 780–793.

[16] X. Zhou, Y. Zhong, Z. Cheng, F. Liang, and L. Ma, "Adaptive sparse pairwise loss for object re-identification," in *Proc. 2023 IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 19691–19701.

[17] X. Shi, H. Liu, W. Shi, Z. Zhou, and Y. Li, "Boosting person re-identification with viewpoint contrastive learning and adversarial training," in *Proc. 2023 IEEE Int. Conf. Acoust., Speech Signal Process.*, 2023, pp. 1–5.

[18] Y. Xiahou, N. Li, and X. Li, "Identity consistency multi-viewpoint generative aggregation for person re-identification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 3, pp. 1441–1455, Mar. 2024.

[19] M. Oquab et al., "DINOv2: Learning robust visual features without supervision," *Trans. Mach. Learn. Res.*, pp. 1–31, 2024. [Online]. Available: https://openreview.net/forum?id=a68SUt6zFt

[20] F. Lu, L. Zhang, X. Lan, S. Dong, Y. Wang, and C. Yuan, "Towards seamless adaptation of pre-trained models for visual place recognition," in *Proc. 12th Int. Conf. Learn. Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=TVg6hlfsKa

[21] S. S. Kannan and B. -C. Min, "PlaceFormer: Transformer-based visual place recognition using multi-scale patch selection and fusion," *IEEE Robot. Automat. Lett.*, vol. 9, no. 7, pp. 6552–6559, Jul. 2024.

[22] M. Wieczorek, B. Rychalska, and J. Dabrowski, "On the unreasonable effectiveness of centroids in image retrieval," in *Proc. Neural Inf. Process., 28th Int. Conf.*, 2021, pp. 212–223.

[23] J. Wang et al., "Learning fine-grained image similarity with deep ranking," in *Proc. 2014 IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1386–1393.

[24] E. Bayraktar, Y. Wang, and A. DelBue, "Fast re-OBJ: Real-time object re-identification in rigid scenes," *Mach. Vis. Appl.*, vol. 33, no. 6, 2022, Art. no. 97.

[25] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[26] M. Roberts et al., "Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding," in *Proc. 2021 IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10912–10922.

[27] J. Collins et al., "ABO: Dataset and benchmarks for real-world 3D object understanding," in *Proc. 2022 IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 21126–21136.

[28] A. Zhang et al., "Towards robust robot 3-D perception in urban environments: The UT campus object dataset," *IEEE Trans. Robot.*, vol. 40, pp. 3322–3340, 2024.

[29] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice, "University of Michigan North Campus long-term vision and lidar dataset," *Int. J. Robot. Res.*, vol. 35, no. 9, pp. 1023–1035, 2016.

[30] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "FAST-LIO2: Fast direct LiDAR-inertial odometry," *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2053–2073, Aug. 2022.

[31] K. Koide, J. Miura, M. Yokozuka, S. Oishi, and A. Banno, "Interactive 3D graph SLAM for map correction," *IEEE Robot. Automat. Lett.*, vol. 6, no. 1, pp. 40–47, Jan. 2021.

[32] M. Ester, H. -P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discov. Data Mining*, 1996, pp. 226–231.

[33] A. Kirillov et al., "Segment anything," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 4015–4026.

[34] T. Xiao, X. Wang, A. A. Efros, and T. Darrell, "What should not be contrastive in contrastive learning," in *Proc. Int. Conf. Learn. Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=CZ8Y3NzuVzO

[35] J. -B. Grill et al., "Bootstrap your own latent - a new approach to self-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 21271–21284.

[36] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 9912–9924.

[37] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. E. Hinton, "Big self-supervised models are strong semi-supervised learners," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 22243–22255.

[38] M. Caron et al., "Emerging properties in self-supervised vision transformers," in *Proc. 2021 IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9650–9660.

[39] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in *Proc. Int. Conf. Learn. Representations*, 2017. [Online]. Available: https://openreview.net/forum?id=Skq89Scxx

[40] G. Jocher, J. Qiu, and A. Chaurasia, "Ultralytics YOLO," Jan. 2023. [Online]. Available: https://github.com/ultralytics/ultralytics