LoCUS: Learning Multiscale 3D-consistent Features from Posed Images

Dominik A. Kloepfer[1], Dylan Campbell[2], João F. Henriques[1]

[1]Visual Geometry Group, University of Oxford
[2]Australian National University
{dominik, joao}@robots.ox.ac.uk, dylan.campbell@anu.edu.au

Abstract

An important challenge for autonomous agents such as robots is to maintain a spatially and temporally consistent model of the world. It must be maintained through occlusions, previously-unseen views, and long time horizons (e.g., loop closure and re-identification). It is still an open question how to train such a versatile neural representation without supervision. We start from the idea that the training objective can be framed as a patch retrieval *problem: given an image patch in one view of a scene, we would like to retrieve (with high precision and recall) all patches in other views that map to the same real-world location. One drawback is that this objective does not promote reusability of features: by being unique to a scene (achieving perfect precision/recall), a representation will not be useful in the context of other scenes. We find that it is possible to balance retrieval and reusability by constructing the retrieval set carefully, leaving out patches that map to far-away locations. Similarly, we can easily regulate the scale of the learned features (e.g., points, objects, or rooms) by adjusting the spatial tolerance for considering a retrieval to be positive. We optimize for (smooth) Average Precision (AP), in a single unified ranking-based objective. This objective also doubles as a criterion for choosing landmarks or keypoints, as patches with high AP. We show results creating sparse, multi-scale, semantic spatial maps composed of highly identifiable landmarks, with applications in landmark retrieval, localization, semantic segmentation and instance segmentation.*

1. Introduction

For an autonomous agent to be able to take useful actions, it must maintain a spatially and temporally consistent
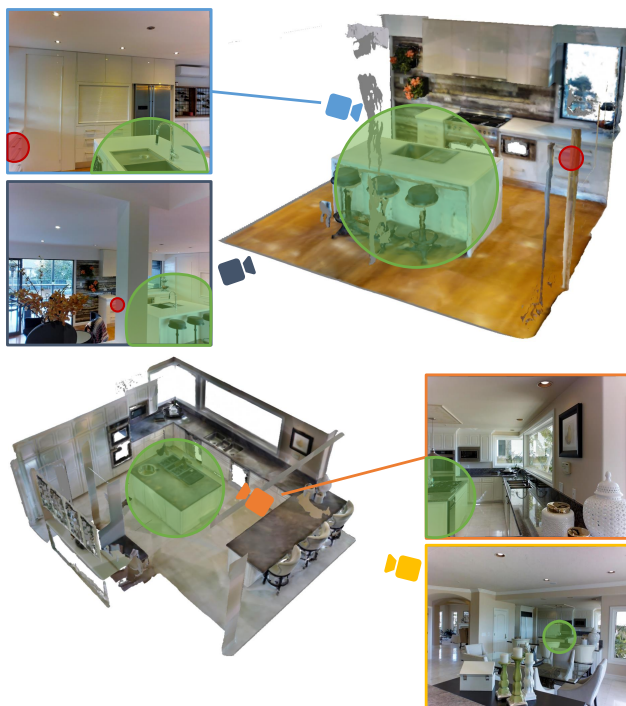

Figure 1: Problem setting. Our goal is to train a network to extract features that are identifiable and 3D-consistent, so that features at image locations corresponding to the same region in 3D space, but viewed from different positions, are similar. This can be done at multiple scales, from large (e.g., the kitchen islands in the large green circles) to small (e.g., the drawers in the small red circles). However, simply optimizing for "unique" representations at each location (e.g., via contrastive learning) runs the risk of over-fitting to the training scenes, as such objectives will discourage reuse of the same representation for different places. Instead, we encourage reusable landmark representations, such as the concept of a kitchen island, which may appear in different scenes (top and bottom panels) with appearance variations.

Code and model weights for this project can be found at https://www.robots.ox.ac.uk/~vgg/research/locus.

world model. This model may comprise the agent itself (e.g., pose estimation), the environment (e.g., mapping), and dynamic objects (e.g., instance detection), and must be maintained when confronted with previously-unseen views, occlusions, and long time horizons (e.g., loop closure and re-identification). However, visual observations used by an agent for this purpose are inherently inconsistent: the same landmark may appear significantly different from different viewpoints in space and time, due to (self) occlusion, reflections, lighting variations, and dynamic effects, among other factors. Errors in the estimation of the agent's internal state compound these problems. Therefore, it is important for any vision-based agent to convert observations into some spatio-temporally consistent form.

Existing approaches [4, 21, 41] do this at the observation synthesis (mapping) stage, by aggregating or distilling visual information in 3D. We argue that significant progress can be made before this point, at the observation processing stage, which lends itself to a more flexible image-centered representation that is useful for a range of tasks. The key is to encourage consistency between image features that unproject to the same region of 3D space, within a spatial tolerance, defining a landmark at a given scale.

This can be achieved by formulating the problem as one of patch retrieval: given an image patch from one view of a scene, retrieve all patches in other views that correspond to the same 3D location, with high precision and recall. To encourage reusability, so that the learned features are useful in new scenes, we exclude patches from the retrieval set if (when unprojected) they exceed a fixed distance from the query. Excluding such patches ensures that the representations of similar-looking landmarks in distant places are not pushed apart unnecessarily, which would promote over-fitting unique representations to the training scenes, thus making them non-reusable in new scenes. Moreover, by adjusting the spatial tolerance that defines the positive set, we can regulate the scale of the learned features. This allows us to learn features at a small scale (e.g., local textures and structures), medium scale (e.g., household objects), and large scale (e.g., whole rooms or places) in the same framework.

We learn this representation by optimizing a ranking-based metric, (smooth) Average Precision (AP), which doubles as a criterion for choosing distinctive landmarks (keypoints). The resulting Location-Consistent Universal Stable (LoCUS) features are semantically-meaningful, 3D-consistent at the selected scale, and balance distinctiveness with reusability, producing sparse, multi-scale, and semantic maps. We demonstrate applications in landmark retrieval, localization, semantic segmentation and instance segmentation.

To summarize, our contributions are:

1. A framework for learning 3D-consistent features from posed images via retrieval, taking into account multiple scales and how to trade off retrieval performance vs. generalization performance (reusability).

2. A unified ranking-based objective function that facilitates the selection of highly-identifiable landmarks.

3. An evaluation of the proposed features on real images of indoor environments, on the tasks of place recognition, semantic segmentation, instance segmentation and re-identification, as well as relative pose estimation.

2. Related work

The topics of keypoint detection and description [12, 14, 31], feature matching [44, 11, 16, 35, 39], structure-from-motion [36], and SLAM [15, 27, 40] have a rich history. Here, we concentrate on the most recent and related work.

Image retrieval. Learning representations for image retrieval—the task of ranking all instances in a retrieval set according to their relevance to a query image—is well-studied [2, 30, 17]. Metric learning approaches use, e.g., contrastive [10] or triplet [43] losses to encourage positive instances to be close, while negative instances are separated by a margin. Other approaches optimize (approximations) to ranking-based metrics like Average Precision (AP) directly [33, 5]. For example, Smooth-AP [5] proposes a sigmoid relaxation of the ranking function, where the tightness of the approximation is controlled by the temperature. Optimizing a ranking metric allows a model to target the correct ranking without caring about the absolute feature distances. We leverage the image retrieval literature by defining our learning task as a patch retrieval problem. By carefully defining the retrieval set, we can balance feature distinctiveness with re-usability. While image retrieval methods retrieve entire images, we retrieve 3D spherical regions projected to 2D. That is, while methods such as Brown et al. [5] compute a single feature per image that is then used to retrieve other images of the same class, we compute features for pixel patches that are then used to retrieve pixel patches that cover (parts of) the same 3D spherical region. More details can be found in Sec. 3.

Learning visual features and keypoints. Several works explore methods to learn better image features or keypoints to facilitate 2D–2D matching [44, 11, 16, 35, 39] or 2D–3D matching [11, 6] for relative/absolute pose estimation or triangulation. For example, Fathy et al. [16] use metric learning to learn 2D–2D matchable features, while Campbell et al. [6] learn geometric features that facilitate 2D–3D matching via an end-to-end trainable blind PnP solver.

Keypoint detectors, by contrast, aim to find a sparse set of repeatable points in an image [12, 14, 31]. For example, SuperPoint [12] jointly computes keypoints and descriptors using a convolutional network trained in a self-supervised framework. Similarly, D2-Net [14] obtains keypoints via non-maximum suppression on the learned feature maps. R2D2 [31] argues that repeatable regions are not necessarily discriminative, so learns to predict keypoint repeatability and reliability separately. Unlike the features learned in these works, the features that optimize our loss function do not vary as rapidly, allowing them to more closely resemble the real scene geometry and enabling segmentation.

Neural mapping and reconstruction. Deep learning approaches have gradually closed the gap on classical Structure-from-Motion [36] and SLAM [15, 27] approaches to mapping and reconstruction. For example, Neural Radiance Fields (NeRF) [25] has demonstrated photorealistic reconstruction for known cameras, and been extended to RGBD SLAM [38, 47], RGB SLAM [46], and semantic mapping [45]. Earlier, MapNet [21] investigated neural localization and mapping through convolution operators, resulting in an environment map that stores multi-task information distilled from the RGBD input, which exhibits emergent semantic meaning. Our approach produces very different kinds of maps: sparse, multi-scale, and semantic, composed of highly identifiable landmarks.

Self-supervised visual feature learning. Vision transformers (ViT) [13] have demonstrated a strong capacity for learning useful and meaningful features from large amounts of unlabelled image data [7, 18, 42]. For example, DINO [7] demonstrated that self-supervised ViT features could be used for unsupervised object segmentation. The model was trained via self-distillation between a student network and a momentum teacher network that receive two different random transformation of an image and are encouraged to encode similar features.

STEGO [18] extends DINO to unsupervised semantic segmentation via contrastive learning. It trains a shallow segmentation network appended to a fixed DINO backbone with contrastive terms that encourage the learned features to form compact clusters while preserving their global relationships. CutLER [42] extends DINO to unsupervised object detection and segmentation, achieving extremely compelling results. The model generates training data for a detector by creating foreground object masks using normalized cuts on the patch-wise similarity matrix of DINO features, with additional object masks being found through an iterative masking procedure.

N3F [41] showed that DINO image features can be distilled into a 3D feature field using the same rendering loss as NeRF [25], given camera pose supervision. They demonstrate that the resulting features are 3D-consistent, enabling 3D instance segmentation and scene editing. Our approach builds on these self-supervised methods by proposing a proxy patch retrieval task defined in 3D, unlike STEGO and CutLER, allowing us to adapt DINO features so that they learn invariances to viewing direction and instance. Like N3F, we require camera pose supervision to enable our 3D-aware loss. Unlike N3F, our features are defined in image space and can be predicted from a single image, facilitating applications like relative pose estimation.

3. Method

Our training procedure will be centered on the concept of recognizing *landmarks*: regions of space that are visually identifiable and unique within a bounded region, but reusable outside that region. We mean that landmark embeddings (representations) are "reusable" in the sense that the same embedding may be shared by more than one landmark, as long as they are far away in the spatial domain.

Assume that we are given a set of training images, divided into n (potentially overlapping) rectangular patches x_i, i.e., the receptive fields of a Convolutional Neural Network (CNN) or the tokens of a Visual Transformer (ViT). Each training patch $x_i \in \mathcal{P}$ is also associated with an environment $e_i \in \mathcal{E}$ (e.g. the identity of a house in a training set composed of distinct houses) and real-world coordinates within that environment $p_i \in \mathbb{R}^3$, obtained for example by projecting the center coordinates of the patch using known camera geometry (camera pose and approximate depth) [19]. Note that this information is only needed for training – at test time no such information is necessary. The training set is then $\mathcal{X} = \{(x_1, e_1, p_1), \dots, (x_n, e_n, p_n)\}$.

Assume that we have also defined a set of *tentative landmarks* $\mathcal{L} = \{(\theta_1, \epsilon_1, \ell_1), \dots, (\theta_m, \epsilon_m, \ell_m)\}$ in 3D space: points $\ell_i \in \mathbb{R}^3$ in environments $\epsilon_j \in \mathcal{E}$ and associated embeddings $\theta_i \in \mathbb{R}^c$. These do not have to correspond to actual landmarks (or identifiable locations in 3D), and can be sampled uniformly across space.[1]

We wish to train a deep neural network $\phi : \mathcal{P} \mapsto \mathbb{R}^c$ to output embeddings that can be used to match each patch x_i to a landmark embedding θ_j, by computing pairwise scores

$$s_{ij} = \frac{\phi(x_i)^\mathsf{T}\theta_j}{\|\phi(x_i)\|\|\theta_j\|}, \qquad (1)$$

consisting of a cosine distance (inner product of normalized embeddings), where higher scores denote more likely matches. To specify whether a match is correct or not, we place a sphere of radius ρ_j around the landmark ℓ_j, and any retrievals there (and in the same environment) are considered positive:

$$y_{ij}^+ = \mathbb{1}\left(\|p_i - \ell_j\| \le \rho_j \wedge e_i = \epsilon_j\right), \qquad (2)$$

---

[1] We will discuss more efficient sampling strategies in Section 3.3.

where $\mathbb{1}(\cdot) \in \{0,1\}$ is the indicator function. We will use y_{ij}^+ as a binary mask to denote positive matches, while $y_{ij}^\Omega = 1$ is a trivial mask that denotes the union of positives and negatives. Both are used to define the Smooth Average Precision (Smooth-AP):

$$\widetilde{\mathrm{AP}}_j = \frac{1}{\sum_i^n y_{ij}^+} \sum_i^n y_{ij}^+ \frac{1 + \sum_{kl}^{nm} y_{kl}^+ \sigma_\tau(s_{kl} - s_{ij})}{1 + \sum_{kl}^{nm} y_{kl}^\Omega \sigma_\tau(s_{kl} - s_{ij})}, \quad (3)$$

with the sigmoid $\sigma_\tau(x) = \frac{1}{1+\exp(-x/\tau)}$. In the limit $\tau \to 0$, $\widetilde{\mathrm{AP}}_j$ recovers the exact AP with θ_j as the query embedding.

Discussion. Eq. 3 is similar to Smooth-AP, proposed by Brown et al. [5], with a few differences that were necessary to adapt it to patch-based landmark retrieval: 1) the retrieval set consists of rectangular image patches, so ϕ can be applied convolutionally; 2) the positive set is defined by 3D Euclidean distance (Eq. 2) with per-landmark radii ρ_j; and 3) we wrote Eq. 3 as a function of binary masks y_{ij}^+ and y_{ij}^Ω, instead of nested sets.

This objective encourages the features from two image patches to be similar if they correspond to 3D locations that are at most a distance ρ apart, since they will be in each other's positive sets. Thus the objective directly encourages 3D-location-consistent features, extracting similar features for different viewpoints of the same 3D location. Empirical support for this is given in Sec. 4.2. The objective also encourages semantic meaningfulness, extracting similar features for image patches that correspond to the same object. First, note that if two 3D locations are separated by greater than ρ but less than 2ρ, they are both within the positive set of a third landmark location, encouraging all three features to be similar. Second, note that the Smooth-AP loss does not minimise the similarity of a landmark with patches in the negative set, it only encourages the similarity with respect to the positive set to be greater than that with the negative set. Together, this results in similar features being extracted across an object, facilitating segmentation.

Multi-scale landmarks. The radius ρ_j of each landmark defines its overall scale, as any matching embeddings $\phi(x_i)$ must be invariant to different positions within this radius. Thus ϕ may learn to recognize not only small-scale keypoints, but also landmarks at the scale of household objects, whole rooms or even larger regions (place recognition), as illustrated in fig. 1.

Despite these changes, Smooth-AP still offers a few other challenges to be adapted to our setting, which we will detail in the next sections.

3.1. Landmark reusability: "don't-care" regions

Optimizing for AP has one unfortunate side effect: every high score matching a patch x_i further away from a (tenta-
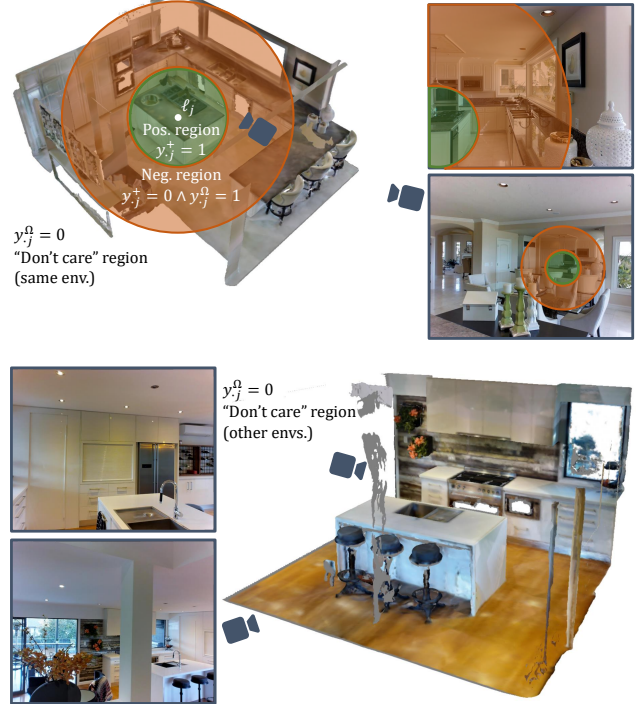

Figure 2: Illustration of the projections of the spherical regions that define the landmark retrieval objective (sec. 3.1). The small green sphere around the tentative landmark ℓ_j defines the region inside which image patches are considered positive matches with the landmark ($y_{ij}^+ = 1$). The larger orange sphere defines the region with positive and negative matches ($y_{ij}^\Omega = 1$). Importantly, outside this region matches are ignored ($y_{ij}^\Omega = 0$), as well as in other environments (bottom panel). As a result, a contrastive (or retrieval) self-supervised objective does not suppress similar embeddings for semantically-similar but spatially distant landmarks, such as the two kitchen islands in the two environments shown.

tive) landmark ℓ_j than ρ_+ will be treated as a false positive, and thus suppressed during training. Likewise, all patches in different environments $e_i \neq \epsilon_j$ will be treated the same. While this seems reasonable on the surface, at the optimum it will force all landmarks to be *unique* to a particular place in an environment and thus useless in a new environment or far away location. We would like some landmarks to be *reusable* and shared among different environments, for example for one landmark to represent a living room in different homes, as opposed to overfitting to a single living room.

In analogy with "don't-care" conditions in digital circuit design [24], which reduce circuit complexity by freeing up modeling capacity for input–output combinations that are

not important, we propose to define "don't-care" regions where the Smooth-AP objective does not constrain the deep network's output. Instead of the trivial mask $y_{ij}^\Omega = 1$ that denotes the universe of all patches as positives and negatives, we instead reduce this universe to

$$y_{ij}^\Omega = \mathbb{1}\left(\|p_i - \ell_j\| \le \kappa\rho_j \wedge e_i = \epsilon_j\right), \quad (4)$$

with $\kappa > 1$ a multiplier for the distance threshold. Together, κ and ρ_j define two concentric regions: a sphere of radius ρ_j around a landmark, where any retrievals are considered positive (Eq. 2), and a spherical shell at distance d from the landmark, with $\rho_j < d \le \kappa\rho_j$, where any retrievals are considered negative (Eq. 4). Any points outside the radius $\kappa\rho_j$ are not considered as part of the retrieval set, and are not assigned a label. The end result is that two different tentative landmarks can have very similar embeddings θ_j, as long as they are at a distance greater than $\kappa\rho_j$, and this embedding reuse will not be penalized by the Smooth-AP (Eq. 3).

3.2. Automatic landmark selection with Vectorized-Smooth-AP

So far we referred to landmarks as "tentative", so that they may not correspond to actual identifiable regions of space. However, optimizing for Eq. 3 assumes a landmark ℓ_j is fixed as a query. If we maximize Eq. 3 in expectation over j (analogously to Brown et al. [5]), we implicitly give equal importance to all tentative landmarks, even if some may correspond to places that are not easily identifiable (e.g., a wall or empty region).

Rather than devise a heuristic to identify good landmarks, we instead just let the Smooth-AP objective focus on pairs of landmarks and patches that maximize AP, by considering all pairs as if they're part of a single query

$$\overrightarrow{\text{AP}} = \frac{1}{\sum_{ij}^{nm} y_{ij}^+} \sum_{ij}^{nm} y_{ij}^+ \frac{1 + \sum_{kl}^{nm} y_{kl}^+ \sigma_\tau(s_{kl} - s_{ij})}{1 + \sum_{kl}^{nm} y_{kl}^\Omega \sigma_\tau(s_{kl} - s_{ij})}. \quad (5)$$

Eq. 5 is equivalent to *vectorizing* the matrix of masks $Y^+ \in \mathbb{R}^{n \times m}$ with elements y_{ij}^+, by stacking its elements into a single vector $\mathbf{y}^+ \in \mathbb{R}^{nm}$, and computing the Smooth-AP objective (Eq. 3) with this modified input. While subtle, this has the effect that $\overrightarrow{\text{AP}}$ will be maximized by first distinguishing the easiest landmark–patch pairs from the rest, while ignoring those that are too ambiguous. By neglecting to emphasize all tentative landmarks equally, the objective adaptively selects highly distinguishable landmarks. We can identify them by evaluating the *non-vectorized* Smooth-AP ($\widetilde{\text{AP}}_j$) on each individually, and taking the top-k landmarks:

$$\ell^* = \underset{j}{\text{top-}k} \ \widetilde{\text{AP}}_j.$$

3.3. Sampling tentative landmarks

We now turn to the definition of the tentative landmark positions ℓ_j and embeddings θ_j.

Sampling positions ℓ_j. While ideally it would be sufficient to sample the landmark positions randomly across space (either within a bounded region, or restricted to the visible hull), in a mini-batch with limited memory this is often not efficient. The reason is that y_{ij}^+ may have too few non-zero values due to non-intersecting image views, especially with a limited number of images in an environment, or in very large environments.

We found that sampling uniformly across space is very inefficient, as over 94% of the chosen tentative landmarks are not visible by more than 2 views (in the training set of Matterport3D [8]; see sec. 4 for details on the experimental setting). This creates very poor query sets for retrieval, with only one or two positive embeddings, which cause overfitting as the network easily attains 100% AP on such tentative landmarks. Instead, we need to bias the sampling more towards more visible locations. Thankfully, there is a simple way to sample spatial positions proportionally to how often they are visible in the training set of views: simply sample uniformly among all image patches across all training images. This guarantees that the sampled distribution is proportional to how often a 3D position is visible, and is easy to implement.

Sampling embedddings θ_j. A straightforward way to define the embedding for ℓ_j is to average the embeddings of all patches that map to that location in space:

$$\theta_j = \frac{1}{n} \sum_i^n y_{ij}^+ \phi(x_i).$$

In practice, we found that approximating this average by a single patch $\phi(x_i)$ such that $y_{ij}^+ > 0$ (chosen at random) is sufficient, which simplifies the implementation.

4. Experiments

In this section, we will detail our experiments, where we evaluate the ability of LoCUS features to perform place recognition and relative pose estimation, as well as evaluate its emergent semantic properties, in the form of semantic segmentation and instance segmentation with object re-identification.

4.1. Experimental setup

Datasets. Our primary dataset for training and evaluation is the Matterport3D dataset [8], which contains a wide variety of indoor environments, captured densely with RGB

Table 1: Place recognition (retrieval) results, for our Lo-CUS features and the DINO [7] baseline. We report our objective, the smooth vectorized AP ($\overrightarrow{\text{AP}}$), and the Average Precision (AP), which quantifies the retrieval performance. For the same features, AP, which corresponds to our objective in the limit of $\tau \to 0$, will always be higher than $\overrightarrow{\text{AP}}$.

Model	Objective ($\overrightarrow{\text{AP}}$)		Average Precision (AP)	
	Train	Val.	Train	Val.
ResNet50 [20]	0.11	0.11	0.11	0.12
DINO [7]	0.20	0.20	0.20	0.20
DINOv2 [29]	0.17	0.17	0.17	0.17
LoCUS (Ours)	**0.56**	**0.54**	**0.57**	**0.55**

and depth information. It also includes dense segmentations at the object level, which facilitate the evaluation of our model's semantic properties.

Training details. We train a 2-stage transformer [13] with 128-dimensional internal features, on top of a frozen DINO backbone [7]. The final features extracted from image patches have 64 dimensions, and the DINO backbone computes 768-dimensional features, so we use two linear layers to map between these feature spaces, resulting in 503,232 trainable parameters. This model is trained by implementing the Vectorized-Smooth-AP objective from Eq. 5. We maximize the objective using the Adam optimizer with an initial learning rate of 10^{-4} and mini-batches of size 16, sampled from the Matterport3D training set [8], and train for 20 epochs. For all experiments, we set the hyperparameters $\tau = 0.01$ and $\rho_j = 0.2$ (in meters). With these settings, the model can be trained on a single NVIDIA RTX 2080Ti GPU.

4.2. Place recognition and retrieval

Since our method is trained with a specific relaxation of Average Precision (AP) on retrieval-focused sets of image patches, its primary objective is most closely aligned with place recognition via retrieval. As such, we start by evaluating its AP on unseen validation environments, which contain objects and layouts that were not seen during training. This assesses the reusability of features produced by our method.

Baselines. For this experiment, we compare with pre-trained ResNet50 [20], DINO [7], and DINOv2 [29] baselines. The features of the final layer of the ViT are reduced to 64 using PCA, the same dimension as our features (similar to Tschernezki et al. [41]). Since our model shares almost all of its weights with the DINO baseline, this compar-

Figure 3: Visualization of co-segmentation results, obtained by thresholding the cosine distance (Eq. 1) of the LoCUS features of a query image patch (blue and orange, highlighted in the top left image) and LoCUS features of patches in other views (remaining images). The thresholded regions are indicated in matching colours.

ison well-illustrates the effect and advantages of our training method.

Results. The results for this experiment are reported in Table 1. In addition to the AP on the validation set, for both our method and the DINO [7] baseline, we also report the AP on the training set. As expected, our LoCUS features significantly outperform the DINO baseline, despite sharing almost all weights. While the retrieval performance decreases on the validation set, this decline is minimal compared to the effect of the training, demonstrating the reusability of the features in unseen environments.

4.3. Semantic and instance segmentation

We now turn to scene-level object segmentation. There are two broad categories of segmentation classes: 1) amorphous geometry ("stuff") such as walls, floor and ceiling; and 2) distinct objects ("things") such as furniture or appliances. The former are useful for evaluating semantic segmentation at the texture level, while the latter requires distinguishing individual objects, and thus allows us to evaluate instance segmentation. This setting is slightly more broad than instance segmentation: an object must not merely be segmented distinctly from other objects in a given image, but it must be also *re-identified* in different images from varied points of view, so it also encompasses the task of object re-identification.

Table 2: Semantic and instance segmentation (respectively "stuff" and "things") results, with object re-identification. Both models extract 64-dimensional feature vectors for 8×8 pixel patches, which are then classified into the relevant classes using a linear probe. Semantic classes contain "stuff" pixels grouped into their semantic categories, while instance classes contain pixels belonging to individual objects. *Uses ground-truth instance labels. †Released after submission deadline.

Model	Semantic			Instance			Overall		
	mAP	mIoU	Jac	mAP	mIoU	Jac	mAP	mIoU	Jac
ResNet50	0.39	0.26	0.55	0.18	0.11	0.12	0.19	0.12	0.41
DINO [7]	0.49	0.34	0.63	0.40	0.28	0.28	0.40	0.29	0.52
DINOv2†	**0.55**	0.38	0.67	0.49	0.35	0.39	0.49	0.35	0.58
Mask2Former	-	0.03	0.07	-	0.00	0.00	-	0.00	0.06
+ Oracle*	-	**0.41**	**0.71**	-	0.39	0.53	-	0.39	0.64
MaskDINO	-	0.05	0.15	-	0.00	0.00	-	0.00	0.12
+ Oracle*	-	**0.41**	**0.71**	-	0.40	0.54	-	**0.40**	**0.65**
LoCUS (Ours)	0.53	0.37	0.67	**0.54**	**0.40**	0.42	**0.54**	0.39	0.59

Qualitative results on co-segmentation. We start by exploring a single co-segmentation task, highlighting a patch in one image and then finding all matching patches in other views, by simply thresholding the similarity metric (Eq. 1). The results can be seen in Section 4.2. We can observe that, despite dramatic changes in viewpoint, the learned LoCUS features are very stable over 3D space, successfully matching over very significant changes in distance, rotation, partial occlusion and out-of-view regions.

Implementation. For the quantitative evaluation, we use linear probes to assess the learned features' correlation with respect to the semantic classes, as is common in self-supervised learning [7]. To do this, we extract the LoCUS features $\phi(x_i)$ over all training images (considered frozen) and train a patch-wise linear classifier with a cross-entropy loss and the ground-truth segmentation labels. We use the same optimizer settings as for the main objective until convergence, for all methods.

Evaluation setting and metrics. We use an evaluation set of *unseen scenes*, which are not part of the training set, and thus test the generalization ability of the methods. We report segmentation metrics on "stuff" pixels only (semantic segmentation), on "things" pixels only (instance segmentation and object re-identification), and on all pixels taken together. For each case, we compute three metrics:

1. mAP: For each object instance (in the case of instance segmentation) or for each class (in the case of semantic segmentation), we calculate the average precision (AP) of the linear classifier, in a one vs. all mode (i.e., considering all other pixels as negative labels). We then average

across all instances or classes to obtain a mAP score.

2. mIoU: We calculate the Intersection-over-Union (IoU) [34] between the predictions and ground-truth binary masks for each object (or class) separately, and then report the average.

3. Jaccard (Jac): Similarly to the mIoU, we compute the Jaccard index separately for each object (or class), and report the average. The Jaccard index is given by TP / (FP + FN), given the counts of binary True Positives (TP), False Positives (FP) and False Negatives (FN).

Baselines. To provide a point of comparison to the semantic segmentation capabilities of the proposed features, we also report results for a number of segmentation baselines. We evaluate pretrained ResNet50 [20], DINO [7], and DINOv2 [29] feature extractors, first reducing the computed features to the same number of dimensions as ours (64) using PCA computed over the full training set. This is the same strategy employed to evaluate Neural Feature Fusion Fields [41]. Similar to our method, we then use a linear probe to produce the segmentations.

We also evaluate two recent segmentation-specific models, Mask2Former [9] and MaskDINO [23] in their default setting, and a setting where we relabel each predicted segmentation mask with the ground-truth scene-consistent instance ID ("Oracle"). The former performs poorly because it does not maintain consistent instance identities across frames (no object re-identification), as is required by this task.

Results. The results are shown in Table 2. Our proposed LoCUS features are better able to discriminate both semantic classes, such as undifferentiated ceiling or wall regions, as well as to re-identify particular object classes. Figure 6 visualizes the qualitative results. The proposed method outperforms the baseline feature extractor methods, especially on the instance segmentation and re-identification task, showing that the object identity predictions are stable under viewpoint changes. DINO features are trained to be invariant to image-space augmentations [7], and so understandably do not enjoy the same stability across viewpoints, especially when they change dramatically.

Our method performs comparably with the Oracle methods despite not receiving the ground-truth labels.

4.4. Relative pose estimation

Since our LoCUS features are trained to be stable across 3D viewpoints, within a specified scale, they should be helpful for tasks that require spatial reasoning. Furthermore, the fact that we can train features for landmarks at different scales should help with coarse-to-fine strategies. For this reason, we focus on relative pose estimation. Note

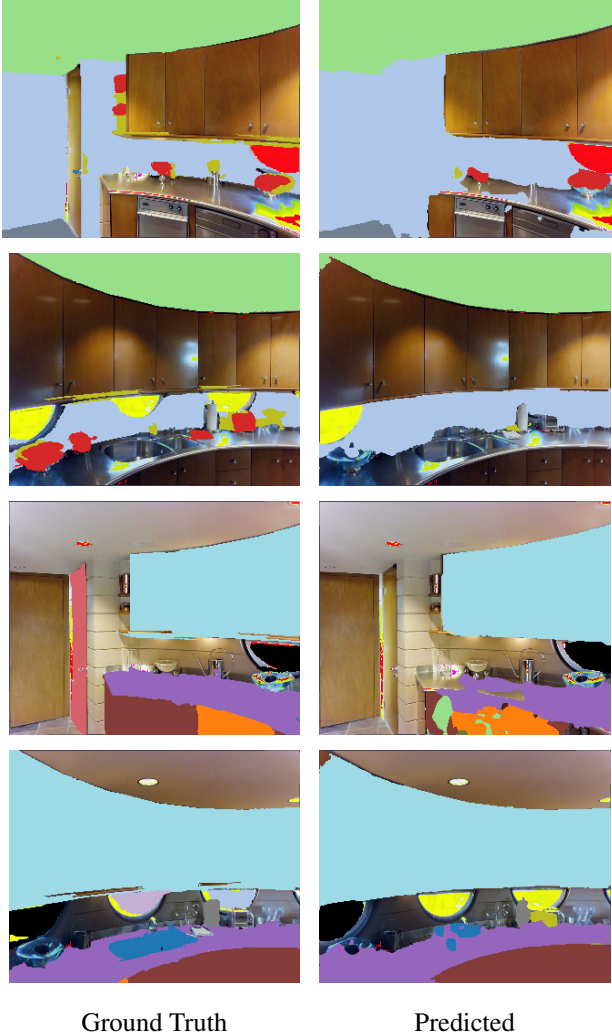|	
Ground Truth	Predicted

Figure 6: Qualitative results of semantic segmentation (wall, ceiling, floor classes) in the first two rows and of instance segmentation (household objects) in the third and fourth row. Note that object instance identities are stable across viewpoints, thus also performing object re-identification.

that this is different from other settings like Simultaneous Localization And Mapping (SLAM), since those assume temporal continuity over a video stream. In contrast, we perform relative pose estimation between single pairs of images, without any extra context, which severely limits the information available.

Dataset. We use the image pairs generated from Matterport3D for relative pose estimation that were introduced in SparsePlanes [22]. We remark that there is very limited overlap between the views of each pair, which makes this an extremely challenging task.

Metrics. We report standard metrics for translation and rotation error. For translation, we report median and average errors, as well as the fraction of pairs that have an error smaller than 1 meter. For rotation, we also report median and average errors, and the fraction of pairs with error smaller than 30 degrees.

Method. Given a pair of images, we extract the LoCUS features of each patch $\phi(x_i)$. We then calculate pairwise scores (Eq. 1), and for each patch, filter out all scores that are smaller than a threshold of 0.7. We then use two conditions on the continuity of the mapping from patches in image A to patches in image B to remove outliers from the set of patch pairs. Details on this process can be found in the supplementary material. Taking the top-100 pairs by score, we use a standard robust 5-point RANSAC algorithm [28] to calculate the essential matrix with the smallest error, and then find the corresponding relative pose (up to unknown scale) using a RANSAC chirality check [3]. The unknown scale can in practice be recovered using for example very coarse depth measurements; here we simply scale the translation vector by its ground truth length.

Baselines. We compare with several baselines from the literature. Most of these were specifically engineered for geometric matching tasks, while ours focuses on coarser (multi-scale) landmark retrieval. As such, we expect ours to be more robust at matching in the large scale, while other methods to do better at very fine-grained geometric matching. We report results for SuperPoint [12] (pre-trained and with its feature dimension reduced to 64 using PCA) with nearest neighbours (NN) search and SuperPoint with FGINN for outlier removal. Given the pixel matches extracted in this way, we compute the in the same way as our method (5-point RANSAC [28]). We also report results for a number of methods that do not extract features, but are specialised to estimate relative poses more directly: Sparse-Planes [22], 8-Point-Supervision [32], and PlaneFormers [1].

Results. The results are presented in Table 3. We can see that, despite not being trained specifically for camera localization, the spatial stability of the trained features does help localize the camera correctly in most instances. Nevertheless, we would expect that with greater overlap between views, methods that are more geared towards fine-grained keypoint matching would do better than coarse matching methods such as ours, which are more concerned with coarse place (landmark) recognition. The most comparable method are the two relative pose estimation algorithms using SuperPoint keypoints, which our method outperforms.

Table 3: Relative pose estimation results. We report translation errors in meters and rotation errors in degrees.

Model	Translation			Rotation		
	Med.	Avg.	≤ 1m	Med.	Avg.	≤30°
SparsePlanes [22]	0.63	1.25	0.67	7.33	22.78	0.83
8-Point-Sup [32]	0.64	1.01	0.67	8.01	19.1	0.85
PlaneFormers [1]	0.66	1.19	0.67	5.96	22.2	0.84
SuperPoint [12]						
+ NN	1.08	1.84	0.48	34.4	47.8	0.47
+ FGINN [26]	1.02	1.87	0.49	29.9	45.2	0.50
LoCUS (Ours)	0.92	1.69	0.53	22.1	34.5	0.58

4.5. Ablation study

We also evaluated the relative impact of different design decisions in our method, and assessed its robustness to different hyper-parameter choices. The results from the preceding sections used the optimal combination under the constraint of similar memory consumption found in this study. We refer the interested reader to the supplemental material for detailed results.

5. Conclusion

We have proposed a method for learning multi-scale view-invariant features from posed images by optimizing a novel retrieval-based objective: Vectorized-Smooth-AP. This objective modulates the DINO [7] ViT features towards 3D-consistency and adaptively selects highly-distinguishable landmarks. Moreover, we select the retrieval set in such a way to encourage the model to balance retrieval (distinctiveness) with reusability (generalisability), through the introduction of a "don't-care" region beyond a certain spatial extent.

We demonstrate compelling performance when using these features for several downstream tasks, including place recognition and retrieval, semantic and instance segmentation with re-identification, and relative pose estimation, demonstrating the utility of our learned features. This result reinforces the strong semantic properties of self-supervised image features and shows how aggregating information in 3D, via the ranking loss function and camera pose supervision, can improve their effectiveness, especially for 3D-aware tasks. Nonetheless, strategies for removing the weak camera pose supervision warrant investigation, since a fully self-supervised approach would facilitate access to greater quantities of data. Depending on the environment, Structure-from-Motion [36] or SparsePose [37] may be able to alleviate this requirement, making it possible to train on larger-scale video data.

References

[1] Samir Agarwala, Linyi Jin, Chris Rockwell, and David F Fouhey. Planeformers: From sparse view planes to 3d reconstruction. In *Proc. ECCV*, pages 192–209. Springer, 2022. 8, 9

[2] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proc. CVPR*, pages 5297–5307, 2016. 2

[3] Gary Bradski. The opencv library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, 25(11):120–123, 2000. 8

[4] Samarth Brahmbhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. Geometry-aware learning of maps for camera localization. In *Proc. CVPR*, pages 2616–2625, 2018. 2

[5] Andrew Brown, Weidi Xie, Vicky Kalogeiton, and Andrew Zisserman. Smooth-ap: Smoothing the path towards large-scale image retrieval. In *Proc. ECCV*, 2020. 2, 4, 5

[6] Dylan Campbell, Liu Liu, and Stephen Gould. Solving the blind perspective-n-point problem end-to-end with robust differentiable geometric optimization. In *Proc. ECCV*, pages 244–261. Springer, 2020. 2

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proc. ICCV*, 2021. 3, 6, 7, 9

[8] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017. 5, 6, 9

[9] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation, 2022. 7

[10] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proc. CVPR*, volume 1, pages 539–546. IEEE, 2005. 2

[11] Zheng Dang, Kwang Moo Yi, Yinlin Hu, Fei Wang, Pascal Fua, and Mathieu Salzmann. Eigendecomposition-free training of deep networks with zero eigenvalue-based losses. In *Proc. ECCV*, pages 768–783, 2018. 2

[12] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proc. CVPR*, pages 224–236, 2018. 2, 3, 8, 9

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. ICLR*, 2021. 3, 6

[14] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint description and detection of local features. In *Proc. CVPR*, pages 8092–8101, 2019. 2, 3

[15] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *Proc. ECCV*, pages 834–849. Springer, 2014. 2, 3

[16] Mohammed E Fathy, Quoc-Huy Tran, M Zeeshan Zia, Paul Vernaza, and Manmohan Chandraker. Hierarchical metric learning and matching for 2D and 3D geometric correspondences. In *Proc. ECCV*, pages 803–819, 2018. 2

[17] Albert Gordo, Jon Almazán, Jerome Revaud, and Diane Larlus. Deep image retrieval: Learning global representations for image search. In *Proc. ECCV*, pages 241–257. Springer, 2016. 2

[18] Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T. Freeman. Unsupervised semantic segmentation by distilling feature correspondences. In *Proc. ICLR*, 2022. 3

[19] Richard I. Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 3

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 6, 7

[21] Joao F. Henriques and Andrea Vedaldi. Mapnet: An allocentric spatial memory for mapping environments. In *Proc. CVPR*, 2018. 2, 3

[22] Linyi Jin, Shengyi Qian, Andrew Owens, and David F Fouhey. Planar surface reconstruction from sparse views. In *Proc. ICCV*, pages 12991–13000, 2021. 8, 9

[23] Feng Li, Hao Zhang, Huaizhe Xu, Shilong Liu, Lei Zhang, Lionel M Ni, and Heung-Yeung Shum. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3041–3050, 2023. 7

[24] Giovanni De Micheli. *Synthesis and optimization of digital circuits*. McGraw-Hill Higher Education, 1994. 4

[25] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: representing scenes as neural radiance fields for view synthesis. In *Proc. ECCV*, pages 405–421. Springer, 2020. 3

[26] Dmytro Mishkin, Jiri Matas, and Michal Perdoch. Mods: Fast and robust method for two-view matching. *Computer vision and image understanding*, 141:81–93, 2015. 9

[27] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015. 2, 3

[28] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE transactions on pattern analysis and machine intelligence*, 26(6):756–770, 2004. 8

[29] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 6, 7

[30] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In *Proc. ECCV*, pages 3–20. Springer, 2016. 2

[31] Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. R2d2: Reliable and repeatable detector and descriptor. In *NeurIPS*, volume 32, 2019. 2, 3

[32] Chris Rockwell, Justin Johnson, and David F. Fouhey. The 8-point algorithm as an inductive bias for relative pose prediction by vits. In *3*. 8, 9

[33] Michal Rolínek, Vít Musil, Anselm Paulus, Marin Vlastelica, Claudio Michaelis, and Georg Martius. Optimizing rank-based metrics with blackbox differentiation. In *Proc. CVPR*, pages 7620–7630, 2020. 2

[34] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proc. MICCAI*, pages 234–241. Springer, 2015. 7

[35] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proc. CVPR*, pages 4938–4947, 2020. 2

[36] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proc. CVPR*, pages 4104–4113, 2016. 2, 3, 9

[37] Samarth Sinha, Jason Y Zhang, Andrea Tagliasacchi, Igor Gilitschenski, and David B Lindell. Sparsepose: Sparse-view camera pose regression and refinement. *arXiv preprint arXiv:2211.16991*, 2022. 9

[38] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison. imap: Implicit mapping and positioning in real-time. In *Proc. CVPR*, pages 6229–6238, 2021. 3

[39] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proc. CVPR*, pages 8922–8931, 2021. 2

[40] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. In *NeurIPS*, volume 34, pages 16558–16569, 2021. 2

[41] Vadim Tschernezki, Iro Laina, Diane Larlus, and Andrea Vedaldi. Neural feature fusion fields: 3D distillation of self-supervised 2D image representations. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2022. 2, 3, 6, 7

[42] Xudong Wang, Rohit Girdhar, Stella X Yu, and Ishan Misra. Cut and learn for unsupervised object detection and instance segmentation. *arXiv preprint arXiv:2301.11320*, 2023. 3

[43] Kilian Q Weinberger, John Blitzer, and Lawrence Saul. Distance metric learning for large margin nearest neighbor classification. In *NeurIPS*, volume 18, 2006. 2

[44] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *Proc. CVPR*, pages 2666–2674, 2018. 2

[45] Shuaifeng Zhi, Edgar Sucar, Andre Mouton, Iain Haughton, Tristan Laidlow, and Andrew J Davison. ilabel: Revealing objects in neural fields. *IEEE Robotics and Automation Letters*, 2022. 3

[46] Zihan Zhu, Songyou Peng, Viktor Larsson, Zhaopeng Cui, Martin R Oswald, Andreas Geiger, and Marc Pollefeys. Nicer-slam: Neural implicit scene encoding for rgb slam. *arXiv preprint arXiv:2302.03594*, 2023. 3

[47] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proc. CVPR*, pages 12786–12796, 2022. 3