Adaptive-Attentive Geolocalization From Few Queries: A Hybrid Approach

Adaptive-Attentive Geolocalization from few queries: a hybrid approach

Gabriele Moreno Berton*, Valerio Paolicelli*, Carlo Masone and Barbara Caputo
Italian Institute of Technology
Turin, Italy
[gabriele.berton, valerio.paolicelli, carlo.masone]@iit.it barbara.caputo@polito.it

Abstract

We address the task of cross-domain visual place recognition, where the goal is to geolocalize a given query image against a labeled gallery, in the case where the query and the gallery belong to different visual domains. To achieve this, we focus on building a domain robust deep network by leveraging over an attention mechanism combined with few-shot unsupervised domain adaptation techniques, where we use a small number of unlabeled target domain images to learn about the target distribution. With our method, we are able to outperform the current state of the art while using two orders of magnitude less target domain images. Finally we propose a new large-scale dataset for cross-domain visual place recognition, called SVOX. The PyTorch code is available at https://github.com/valeriopaolicelli/AdAGeo .

1. INTRODUCTION

In the last decade research on visual place recognition (VPR) has experienced a steady growth, fostered by the availability of large geolocalized image datasets and of smartphones with integrated cameras that make it very easy to capture and share new data. This growth is confirmed by an increasing number of services that rely on visual place recognition systems, such as 3D reconstruction, consumer photography -"Where did I take these photos?"- and augmented reality. Moreover, the limitations of localization and orientation systems (e.g. unreliability of GPS signal in urban canyons) make visual place recognition extremely important for the success and scalability of self-driving cars and autonomous robots.

In literature, geolocalization is generally cast as an image retrieval problem. Given a query image as input, the algorithm is tasked to find images that depict the same place from a geotagged dataset, called gallery. Most of the recent studies have tried to improve upon this task by using deep convolutional neural networks to extract better representations for the retrieval. However, they typically consider the case of queries and gallery images belonging to the same domain [20, 22, 26, 43]. When the query and gallery images belong to different domains, e.g. due to changes in weather conditions, illumination or season, the performance of such place recognition approaches can be significantly degraded [31, 39]. Therefore, to make VPR approaches viable in long-term applications we need to explicitly address the cross-domain setting. In this work we tackle this challenge with a novel two-blocks architecture, called AdAGeo, that is aimed at learning robust representations for images for both source (gallery) and target (query) domains. The first block is designed to learn a mapping from the source to the target domain. This mapping is then used to transfer the style of the target domain to the labeled query images of the source domain, as an effective domain-driven data augmentation technique. The second block is tasked with producing a representation of the input data that is able to comply with different domains and is suitable for the retrieval task. This is achieved by a combination of an attention module and a domain adaptation module. Remarkably, both parts of our architecture only need few unlabeled images from the target domain to be trained. This is of paramount importance to attain a scalable VPR solution, because collecting large amounts of data every time the algorithm needs to be deployed to a different domain is impractical if not infeasible. To the best of our knowledge, this is the first architecture for few-shot domain adaptation in visual place recognition. Additionally, for training and validating our method we have built a new large-scale multi-domain dataset, called SVOX (Street View Oxford dataset), that consists of images of Oxford taken from Google Street View (gallery) and queries taken from the Oxford RobotCar dataset [23].

Contributions To summarize, the contributions of our work are:

- We present a new dataset, called SVOX, that combines Street View Images (gallery) and queries from

*The authors equally contributed

the Oxford RobotCar dataset [23], for the first city-wide multi-domain setting for visual place recognition.

- We propose a deep architecture for visual place recognition that combines two orthogonal domain adaptation modules: i) a generative approach to generate labeled data from the target domain; ii) a method to produce domain-invariant features. AdAGeo achieves a significant localization improvement using just few images from the target domain (more than 13% improvement with 5 images). To the best of our knowledge this is the first hybrid architecture for few-shot domain adaptation in geolocalization.
- We propose an attention mechanism through class-specific activation maps, which are used as score maps to weight the features during the image retrieval training and testing processes.
- We perform an extensive ablation study as well as comparisons with the current state of the art, demonstrating that our method is able to achieve better performance using just 5 target domain images, while other approaches require hundreds of images.

2. Related works

In the following we review previous work on VPR and domain adaptation, the two fields closer to our contribution.

Visual Place Recognition. Most VPR approaches cast the problem as an image retrieval task [2, 14, 20, 22, 24, 26, 36]. This is mostly due to the fact that recent years have seen a huge increase in large scale datasets that cover entire cities or countries, both for research [6, 23, 37] and for commercial use (such as Google Street View, Bing Streetside and Apple Maps). The increasing availability of datasets has also allowed end-to-end deep learning methods to become dominant in this field, combining deep feature extraction backbones with trainable aggregation modules [2, 12] or pooling layers [29]. Between the feature extractor and the head, recent architectures for VPR have introduced other modules to improve the retrieval performance. In particular, following the success of class activation maps (CAM) [40], several architectures have implemented some sort of attention module [20, 22, 24, 26, 43] to make the models more robust. For what concerns the problem of cross-domain VPR, it has mostly been addressed indirectly and with a limited scope, with approaches that are based on heuristics (e.g. selecting features corresponding to man-made structures [25]), on regions of interest [7], or tailored for a specific domain shift (e.g. day/night [35, 11]). However, none of these methods allows for generalization. Only few previous works have explicitly tackled the cross-domain problem. In particular, [28, 1] both use GANs to replace the query with a synthetic image that depicts the same scene but with the appearance of the source domain. The authors of [36] instead use MK-MMD [13] for domain adaptation and allow the localization of old grayscale photos against a gallery of present-day images. Both source and target datasets are not available at the time of this writing. While these prior works use either a generative approach or a domain adaptation method, in AdAGeo we combine both solutions and show that there is a benefit to this. Moreover, AdAGeo is truly a few-shot domain adaptation solution that requires as little as 5 unlabeled and not aligned images from the target domain to produce convincing results, whereas [28, 1, 36] need several orders of magnitude more images from the target domain.

Domain adaptation. Unsupervised Domain Adaptation attempts to reduce the shift between the source and target distribution of the data by relying only on labelled source data and unlabeled target data. There are typically two approaches that are used for unsupervised domain adaptation. The first approach is based on learning a style-transfer transformation to map images from one domain to the other. The cross-domain mapping is usually learned through GANs, as in [17, 18], or autoencoders [32]. The authors of [42] propose to use a cycle-consistency constraint to learn a meaningful translation, which has since been used in a number of tasks [3, 9, 16, 30]. The second approach is based on learning domain-invariant features from the data, building on the idea that a good cross-domain representation contains no discriminative information about the origin (i.e. domain) of the input. This approach was introduced by [10], where a domain discriminator network and the gradient reversal layer (GRL) forces the feature extractor to produce domain-invariant representations. This method found successful applications in many tasks, such as object detection [10], semantic segmentation [4] and video classification [5]. As an alternative, [38] shows that features with larger norms are more transferable across domains, and proposes to increasingly enlarge the norms of the embedding during training. In this work we integrate approaches from both kinds in a unique pipeline that only needs few samples from the target domain. We demonstrate via an ablation study that the improvements provided by the two methods are complementary, thus they can be advantageously combined.

3. Dataset

In order to address the cross-domain VPR problem we need a dataset that supports different domains between gallery (source) and queries (target). In recent years there have been few VPR datasets that include multiple ambient conditions (weather, seasons, ligthing) [6, 23, 37, 2, 34, 31], however they do not fit our use case due to a limited number of domains [2], a limited geographical coverage [23, 31], a

	SVOX		RobotCar				
	Gallery	Queries	Snow	Rain	Sun	Night	Overcast
Train	22232	11294	750	714	712	702	705
Val	17226	14698	-	-	-	-	-
Test	17166	14278	937	870	854	823	872

Table 1: Sizes of SVOX dataset and Oxford RobotCar [23] from 5 different scenarios

non dense collection of images [6] or a non urban setting [34]. For this reason we built a new dataset specific for cross-domain VPR in urban setting, called Street View Oxford (SVOX).

To build SVOX, we used Google Street View to extract images covering a wide area in the city of Oxford. In particular, we took images from 2012 for the gallery, and images from 2014 as training queries (see Tab. 1), making sure that for each query there is at least one positive sample in the gallery from previous years. We split the dataset in three geographically disjoint subsets, for training, validation and testing. The images collected from Google Street View provide the single source domain. Then, to provide different target domains we used samples from the Oxford RobotCar dataset [23] in which images are conveniently tagged according to their weather or lighting conditions. For all our experiments we use the 5 domains of Snow, Rain, Sun, Night and Overcast, as defined in the RobotCar dataset. Figures 1b-1g show the differences between the 6 domains. Notice that besides weather, season, and lighting conditions, the RobotCar domains also differ from the source domain for the viewpoint (the hood of the car is visible in the foreground). Similarly to [27], we take one image every 5 meters, in order to avoid using highly redundant data, for example collected when the car was stuck with a red traffic light. This procedure results in roughly 1500 images per domain. The images collected from the RobotCar dataset are used for domain adaptation and as queries to test the models on different target domains. As shown in Fig. 1a, we ensure that for each target query (RobotCar [23]) there is at least one positive sample in the source test gallery (SVOX). Moreover the split is such that the SVOX training data (gallery and query sets) does not overlap RobotCar places (target sets), to avoid possible overfitting.

Further details about the procedure implemented to collect SVOX are provided in the supplementary material.

4. Method

In this section we present AdAGeo, our method for domain adaptive and attentive visual place recognition. The architecture is composed of two parts (Fig. 2), which are trained separately. The first one is a few-shot domain-driven data augmentation (DDDA) module (Sec. 4.1). By using just few images from the target domain, this module is able to effectively transfer their style to the source domain images. In this way we can use these labeled augmented images to make the VPR model robust to the target domain. The second block is made of a CNN encoder, which extracts features for the domain adaptation (DA) module (Sec. 4.3), and for the attention (Att) module (Sec. 4.2) followed by a descriptors aggregator (Sec. 4.4), which builds robust attentive embeddings for each image.

As shown in Fig. 2, during the phase 2, the network receives the SVOX gallery set as retrieval gallery, the SVOX query set and the related pseudo-target images as queries to perform the main task, while the unsupervised domain adaptation (DA) task is computed over SVOX, pseudo-target and just a few target images.

4.1. Few-shot domain-driven data augmentation

In unsupervised domain adaptation we have a labeled source dataset $X^s = \{(x_i^s, y_i^s)\}_{i=1}^{n^s}$ made of n^s samples (comprising gallery and queries) from source domain D_s, and an unlabeled target dataset $X^t = \{(x_j^t)\}_{j=1}^{n^t}$ made of n^t samples from target domain D_t. The goal of our few-shot domain-driven data augmentation is to learn a mapping from D_s to D_t, in the case where n^t is small (results of of experiments with different values of n^t are later shown in Fig. 5b). This mapping is used as data augmentation for the training queries, to generate labeled target domain queries, and to ultimately make the image retrieval model more robust to the domain shift. We take inspiration from [8], which proposes an approach for the related problem of learning a bi-directional mapping between two domains, for which they only have one sample belonging to D_t. The idea is to use an architecture made of two parallel autoencoders, one for each domain. Let us call Ae_S and Ae_T the two autoencoders, where $Ae_S(x) = Dec_S(Enc_S(x))$ and $Ae_T(x) = Dec_T(Enc_T(x))$, with Enc_S and Enc_T denoting encoders and Dec_S and Dec_T decoders. The goal is to minimize the distance between the distributions of the latent spaces of the two autoencoders, forcing the encoders to produce domain-invariant embeddings, while at the same time each decoder should be able to translate the embeddings to an image in its own domain. This is achieved by minimizing a reconstruction loss on both autoencoders:

$$L_{REC} = \sum_{s \in S} \|Ae_S(s)) - s\|_1 + \sum_{t \in T} \|Ae_T(t)) - t\|_1 \quad (1)$$

as well as cycle-consistency losses:

$$L_{sts-cycle} = \sum_{s \in S} \|Dec_S(\overline{Enc_T}(\overline{Dec_T}(Enc_S(s)))) - s\|_1$$
$$L_{tst-cycle} = \sum_{t \in T} \|Dec_T(\overline{Enc_S}(\overline{Dec_S}(Enc_T(t)))) - t\|_1 \quad (2)$$

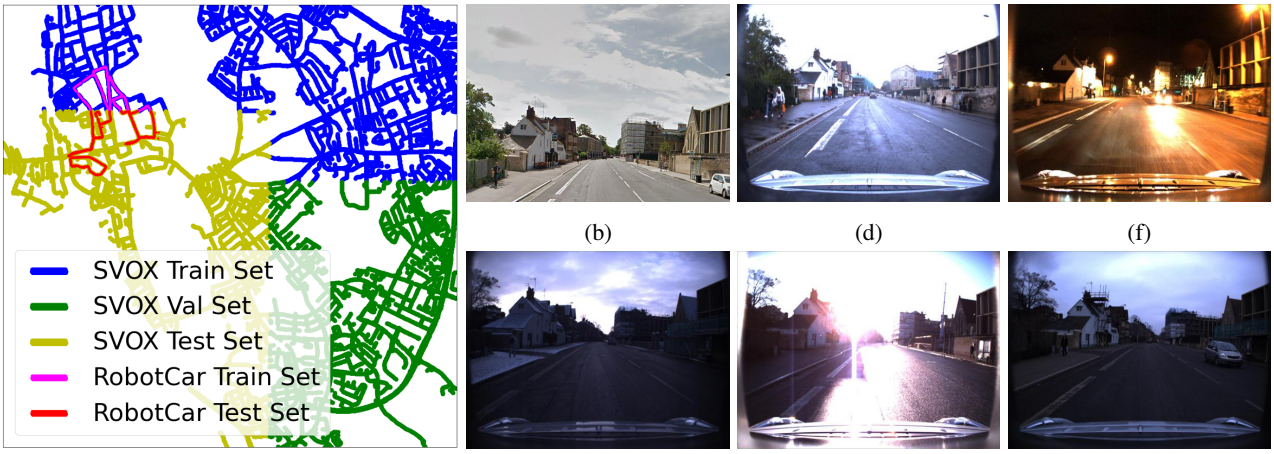


Figure 1: Combining SVOX and RobotCar datasets: a) shows the areas covered by SVOX and RobotCar [23] on the Oxford city map. b) an example of image from SVOX; c-g) examples from the RobotCar scenarios: respectively Snow, Rain, Sun, Night and Overcast, depicting the same location as image b.

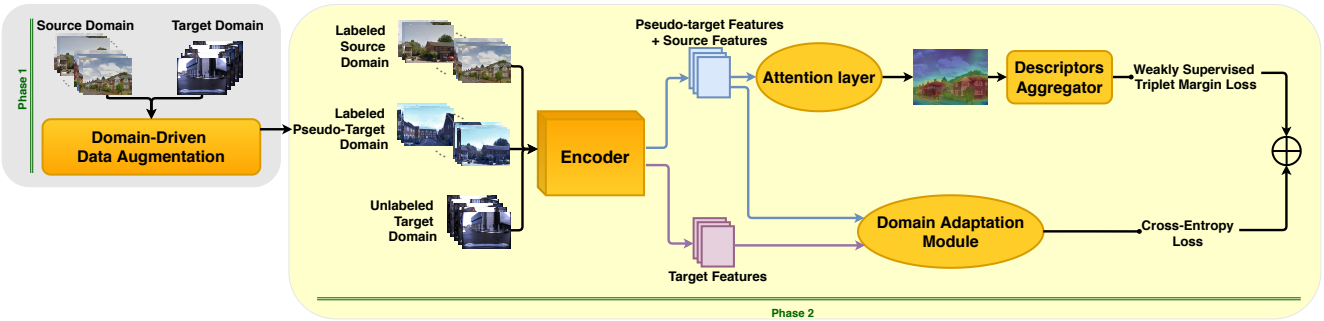


Figure 2: The proposed AdAGeo architecture: training is performed in two phases: during phase 1 the domain-driven data augmentation learns a transformation from source to target domain from just 5 target images. The transformation is then used to generate labeled pseudo-target images. Phase 2 is tasked with the actual geolocalization task, leveraging the generated pseudo-target images, an attention layer and a domain adaptation module.

where the bar above a module means that its weights are frozen during backpropagation of this loss. Moreover, it is important that the embeddings approximate a Gaussian distribution, which helps the two domains to better align, and can be achieved through a variational loss on both encoders:

$$\begin{aligned} L_{VEnc_S} &= \sum_{s \in S} KL(\{Enc_S(s)|s \in S\} \| \mathcal{N}(0, I)) \\ L_{VEnc_T} &= \sum_{t \in T} KL(\{Enc_T(t)|t \in T\} \| \mathcal{N}(0, I)) \end{aligned} \tag{3}$$

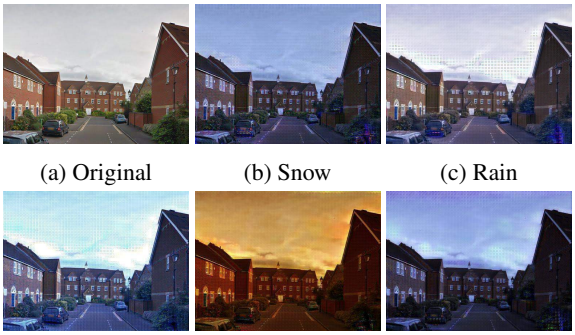
We can then compute the final loss as:

$$\begin{aligned} L_{final} = L_{REC} + L_{sts-cycle} + L_{tst-cycle} + \\ 0.001 L_{VEnc_S} + 0.001 L_{VEnc_T} \end{aligned} \tag{4}$$

Once the training process is finished, it is possible to generate new images from the source domain dataset X^s, by translating them into the target domain D_t. We therefore generate a new pseudo-target dataset $X^{pt} = \{(x_i^{pt}, y_i^{pt})\}_{i=1}^{n^{pt}}$ where $n^{pt} = n^s$, $x_i^{pt} = Dec_T(Enc_S(x_i^s))$ and $y_i^{pt} = y_i^s$ for all $i \in \{1, 2...n^{pt}\}$. We call this pseudo-target because its domain $D_{pt} \approx D_t$. The creation of the pseudo-target dataset is a data augmentation technique performed only once, offline, in order to speed up the training of the second part of the architecture.

4.2. Attention mechanism

In order to highlight the most important features' areas for the retrieval task, we introduced an attention layer after the encoder. To this purpose, we took inspiration from the class activation map (CAM) paper [40] which tries to focus on discriminative image areas that are the most useful to produce the class output in the image classification task, ex-



(a) Original (b) Snow (c) Rain

(d) Sun (e) Night (f) Overcast

Figure 3: An image from the SVOX dataset (a) and the 5 generated pseudo-target images over the 5 domains of RobotCar. Although the generated images present visible artifacts, this step is essential for cross-domain robust geolocalization (see Tab. 3).

ploiting the final average pooling layer present in recent networks such as the ResNet [15]. Let us consider for a given image of dimension $3 \times H \times W$ the extracted feature representation f of shape $D \times H_1 \times W_1$ where D is the number of kernels from the last convolutional layer in the encoder. Furthermore, consider also the backbone classifier block, which contains a fully connected layer with $D \times C$ weights w_{cd} with d values respectively for each class c. The attention map AM_c for a given class c is obtained by the following linear combination:

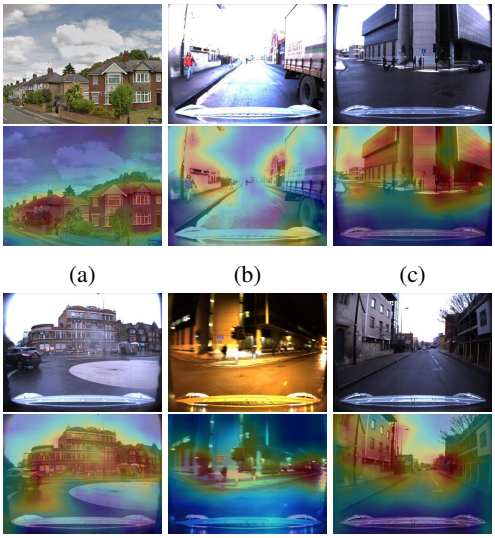
$$AM_c = \sigma(\sum_d f_d \cdot w_{cd}) \tag{5}$$

where σ is the softmax function and whose result has dimension $H_1 \times W_1$.
Finally, AM_c is upsampled to $H \times W$ and is applied over the input image, to visualize the most relevant regions for that class c.

In our architecture we used the fully connected layer of a CNN pretrained on Places365 [41], which contains $C = 365$ classes, to produce the AM_c. The idea stems from the fact that the classes in Places365 [41] (such as house, building, market) are inherently relevant to our task. The images are passed to the whole backbone extracting the local features representation f from the last convolutional layer and producing the $AM_{c_{max}}$ for the category c_{max} with the highest probability $\mathbb{P}$, predicted by the fully connected layer. Then, the features are spatially weighted with the scores calculated before:

$$\begin{gathered} f^w = f \cdot AM_{c_{max}} \\ c_{max} = c_i \mid i = \arg\max_j [\mathbb{P}(c_j)], \forall j \in C \end{gathered} \tag{6}$$

producing new weighted features f^w with the same dimensions as f.



(a) (b) (c)

(d) (e) (f)

Figure 4: Visualization of attention score maps on source domain (a) and target domains (b-f) unseen by the attention module at training time.

We demonstrate that the attention mechanism is useful also for the target images, since the salience regions can help to distinguish also the elements across different domains. Fig. 4 shows the results obtained applying the attention mechanism over all domains at test-time, which shows significant visual results also over target domains unseen by the attention module.

4.3. Domain adaptation module

In order for the retrieval to work well across domains it is important that the embeddings produced by the attention module are domain agnostic, i.e. they do not encode domain-specific information. We achieve this by using a domain discriminator which receives embeddings from the three domains D_s, D_{pt} and D_t. The discriminator is composed of two fully connected layers, and its goal is to classify the domain to which the embeddings belong. Just before the discriminator there is a gradient reversal layer (GRL) [10], that in the forward pass acts as an identity transform, while in the backward pass multiplies the gradient by -λ, where $\lambda > 0$. The use of this layer effectively sets up a minimax game strategy, where the discriminator tries to minimize the domain classification loss, that is a cross-entropy loss $\mathcal{L}_{CE}$, while the feature extractor learns to produce domain-invariant embeddings, acting as an adversary to the discriminator.

4.4. Weakly supervised descriptors aggregation

In order to transform the attentive embeddings into vectorized representations of each image we use a NetVLAD [2]

layer, arguably the most common descriptor aggregator for VPR [2, 20, 36]. To use NetVLAD, we first perform K-means clustering over 500 randomly sampled embeddings of images from all 3 domains to compute K centroids. Then, given the embeddings f^w of dimension $D \times H_1 \times W_1$, reshaped with dimensions $D \times R$ where $R = H_1 \times W_1$, the (j, k) element of the VLAD representation V [19] is computed as

$$V(j,k) = \sum_{i=1}^{R} \frac{e^{-\|f_i^w - c_k\|^2}}{\sum_{k'} e^{-\|f_i^w - c_{k'}\|^2}} \cdot (f_i^w(j) - c_k(j)) \quad (7)$$

where $f_i^w(j)$ and $c_k(j)$ are the j-th dimensions of the i-th embedding and k-th centroids, respectively; while the fraction is the soft-assignment of descriptor f_i^w to centroid k-th. Given the intrinsic nature of VPR data, where the label for each image is represented solely by its position, it is not possible to use standard supervised losses to drive the training, because two photos taken in the same position (therefore with the same label) but opposite directions would depict different locations. To overcome this, we use a weakly supervised triplet margin loss [2], which for each query q is defined as

$$\mathcal{L}_{triplet} = \sum_{y}^{Y} h(\min_i d^2(F(q), F(p_i^q)) + m - d^2(F(q), F(n_y^q))) \quad (8)$$

where $d(\cdot, \cdot)$ represents the Euclidean distance, $F(x)$ is the features representation for image x, $\{p_i^q\}$ is the set of potential hard positives (images within 10 meters from the query q), $\{n_y^q\}$ is the set of Y negatives (further than 25 meters), h is the hinge loss and m is a constant parameter chosen as margin.

5. Experiments

In this section we explain the experimental protocol, focusing on the methods considered for comparisons, the training details for AdAGeo, the experimental results and an ablation study.

5.1. Comparisons with other methods

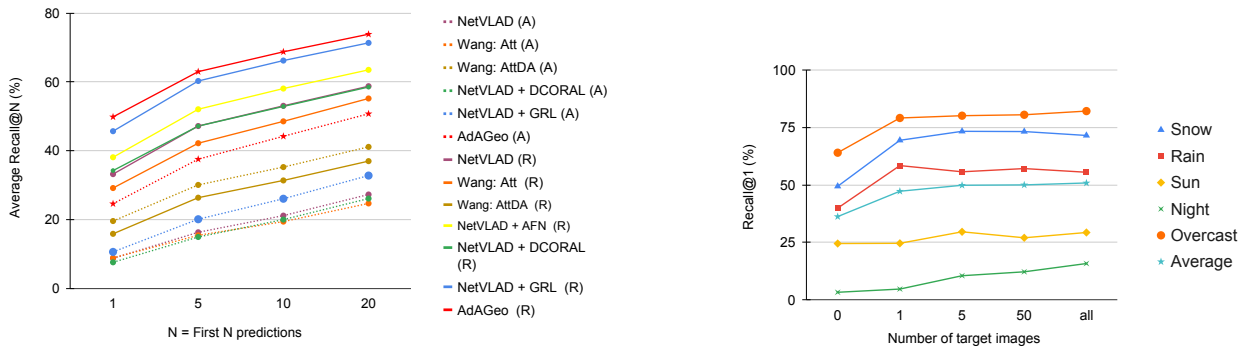
To compare AdAGeo with other methods, we first consider NetVLAD [2], arguably the most used and well-established method for visual place recognition. We also compute results with the only other method built for VPR with domain adaptation, by Wang et al. [36], which uses an attention module and MK-MMD [13], with the code provided by the authors. We used the two variants proposed by the authors, the first one with just the attention mechanism (Wang: Att) and the second one with also the DA branch (Wang: Att+DA). Given the lack of other methods for the task, we implement NetVLAD [2] with a GRL [10] branch, as well as NetVLAD [2] with a DeepCORAL [33] branch and NetVLAD [2] with an SAFN [38] branch, as SAFN is chosen as the current state of the art for domain adaptation. For SAFN, we compute the features norm from the embeddings produced by the last convolutional layer of the backbone, using the code provided by the authors. For fairness of comparisons, we compare the methods using as backbones AlexNet [21] and ResNet18 [15], both cropped at the last convolutional layer, pretrained on Places365 [41].

5.2. Training details

The training process is split in two distinct phases, as shown in Fig. 2. The first phase is tasked with building the pseudo-target dataset using $n^t = 5$ target domain images (Sec . 4.1). We adopt the successful architecture of [8] consisting in two encoders made of two convolutional layers and four residual blocks, and two symmetric decoders made of four residual blocks and two deconvolutional layers. In this phase we use the Adam optimizer with learning rate 0.0002 and batch size 1. The second phase is tasked with building the embedding for each image, and the domain adaptation task is performed using the same target domain images as in the first phase. The backbone pretrained on Places365 [41] is finetuned from the last two convolutional blocks to the end (both for AlexNet [21] and ResNet18 [15]), while the features are extracted at the last convolutional layer, before ReLU, to be passed to the attention and the domain adaptation modules. As optimizer we use Adam with learning rate 0.00001, and for each iteration we use 4 tuples, each consisting of 1 query image, the best positive, and 10 negative samples. The negative samples are chosen following the standard described in [2], in order to increase the likelihood that $\mathcal{L}_{triplet} > 0$, by making sure that each negative is similar enough to the positive. The two losses are combined as $\mathcal{L}_{triplet} + \alpha \cdot \mathcal{L}_{CE}$ where $\alpha = 0.1$. Finally, unlike most domain adaptation methods which train the network for a constant number of epochs, or perform validation and early stopping on the source validation set, we perform validation and early stopping on the generated pseudo-target validation set which, having a similar distribution to the target set, helps to stop the training in an optimal position.

5.3. Results

All methods are trained on SVOX dataset (Tab. 1). For methods which use domain adaptation (DA), we used the whole unlabeled target train set from Oxford RobotCar [23] (around 800 images, depending on the domain, see Tab. 1) for the DA task. For our architecture, we only used 5 images from the unlabeled target set for DA, simulating a five-shots scenario. Testing is then performed using the test gallery from SVOX and the test queries from Oxford RobotCar [23]. For methods with DA, trainings are per-



(a) (b)

Figure 5: **a)** Comparison between all methods, shown as recall@N averaged over the 5 target domains (Average Recall@N). The base CNN encoder is denoted in brackets: (A)lexNet and (R)esNet18. **b)** Results of experiments with AdAGeo with 0-shots, 1-shot, 5-shots, 50-shots and all-shots. With easier domains (Snow, Rain, Overcast) AdAGeo shows a good improvement in accuracy with just 1 target domain image, while with more challenging domains (Sun, Night) AdAGeo requires a higher amount of images to perform significant improvements.

Method	EN	#T	Snow	Rain	Sun	Night	Overcast	Avg
			R@1	R@1	R@1	R@1	R@1	
NetVLAD[2]	A	0	9.8 ± 2.6	9.2 ± 1.5	2.0 ± 0.2	0.0 ± 0.0	22.5 ± 4.5	8.7
Wang: Att[36]	A	0	11.7 ± 1.5	9.8 ± 0.6	2.9 ± 0.6	0.2 ± 0.1	19.6 ± 2.0	8.8
Wang: Att+DA[36]	A	all	28.7 ± 1.5	20.6 ± 2.5	6.4 ± 0.2	0.8 ± 0.2	41.3 ± 1.0	19.6
NetVLAD[2]+DCORAL[33]	A	all	9.6 ± 1.1	8.7 ± 1.3	2.2 ± 0.3	0.1 ± 0.1	17.2 ± 2.0	7.6
NetVLAD[2]+GRL[10]	A	all	12.4 ± 3.4	9.2 ± 1.8	3.2 ± 0.3	0.1 ± 0.2	28.0 ± 2.4	10.6
AdAGeo (ours)	A	5	**34.9** ± 2.2	**26.4** ± 3.3	**10.0** ± 0.1	**1.7** ± 0.4	**49.9** ± 1.9	**24.6**
NetVLAD[2]	R	0	50.1 ± 1.3	36.5 ± 0.6	17.7 ± 0.9	1.6 ± 0.4	60.0 ± 0.7	33.2
Wang: Att[36]	R	0	47.2 ± 5.0	28.1 ± 3.3	13.5 ± 1.9	1.3 ± 1.4	55.7 ± 4.6	29.2
Wang: Att+DA[36]	R	all	23.8 ± 6.2	11.2 ± 1.4	5.7 ± 0.5	0.9 ± 0.5	37.6 ± 8.2	15.8
NetVLAD[2]+SAFN[38]	R	all	57.3 ± 2.5	43.6 ± 0.4	19.1 ± 2.0	2.2 ± 0.7	68.3 ± 1.2	38.1
NetVLAD[2]+DCORAL[33]	R	all	60.2 ± 2.0	33.5 ± 1.1	14.1 ± 0.6	2.1 ± 0.8	61.2 ± 3.6	34.2
NetVLAD[2]+GRL[10]	R	all	68.9 ± 2.5	50.9 ± 2.0	27.1 ± 4.8	4.6 ± 1.2	76.9 ± 0.7	45.7
AdAGeo (ours)	R	5	**73.3** ± 2.2	**55.7** ± 1.8	**29.6** ± 1.0	**10.5** ± 1.9	**80.1** ± 1.5	**49.8**

Table 2: Comparison between all methods, shown as recall@1 (R@1) on each target domain. Column EN stands for the encoder used: AlexNet (A) or ResNet18 (R). #T shows the number of target images used at training time. Snow, Rain, Sun, Night and Overcast are the 5 target domains of the SVOX+RobotCar dataset. The last column shows the average recall@1 across all domains.

formed separately for each of the 5 target domains (Snow, Rain, Sun, Night and Overcast). As evaluation metric, we use the percentage of correctly localized queries within the first N predictions, known as recall@N, as standard practice for place recognition [2, 36, 20, 26, 22, 43]. A query is deemed correctly localized if at least one of the top N retrieved gallery images is within 25 meters from the ground truth position of the query. Results for each method over each domain are shown in Tab. 2. Our AdAGeo framework outperforms all other approaches with both AlexNet and ResNet18 encoders while using two orders of magnitude less target domain images, which verifies the effectiveness of our method. Moreover, AdAGeo presents good results with both encoders, showing the stability of the framework, while other methods are highly dependent on the architecture of the features extractor. More comparisons of each method are shown in Fig. 5a. The supplementary material provides an additional qualitative comparison between our method and the best baseline by visualizing some retrieval results.

For fairness of comparison, we ran all experiments 3 times in a fully deterministic environment, with seeds 0, 1 and 2,

Method	Snow	Rain	Sun	Night	Overcast	Avg
	R@1	R@1	R@1	R@1	R@1	
Baseline	50.1	36.5	17.7	1.6	60.0	33.2
Baseline+DDDA	61.3	45.3	23.3	6.1	71.1	41.4
Baseline+Att	49.4	39.9	24.5	3.3	64.0	36.2
Baseline+DA	65.3	49.7	25.4	6.0	75.2	44.3
Baseline+DDDA+Att	66.6	54.5	27.3	5.5	72.2	45.2
Baseline+DDDA+DA	67.2	51.5	24.8	9.4	78.4	46.3
Baseline+Att+DA	66.0	49.1	24.8	3.2	76.1	43.8
AdAGeo	**73.3**	**55.7**	**29.6**	**10.5**	**80.1**	**49.8**

Table 3: Ablation table of our proposed framework on the SVOX+RobotCar dataset in a 5-shot setting with ResNet18 as encoder. R@1 = recall@1, DDDA = Domain-Driven Data Augmentation, Att = Attention layer and DA = Domain adaptation layer.

and we present the mean over the 3 runs.

5.4. Ablation study

We evaluate the components of our method by conducting an extensive ablation study over each target domain of SVOX+RobotCar. The results are shown in Tab. 3, where all experiments have been run in a 5-shot environment (except for the experiments where the target domain is not used) and all the modules combination are tried. As baseline, we use a ResNet18 encoder (cropped at the last convolutional layer) followed by a NetVLAD [2] descriptor aggregator. Then, each component is added to the baseline: Baseline + Domain-driven data augmentation module (DDDA), Baseline + Attention module (Att), Baseline + Domain adaptation module (DA) and all their combinations (Baseline+DDDA+Att, Baseline+DDDA+DA, Baseline+Att+DA) until the entire AdAGeo architecture (Baseline+DDDA+Att+DA). As shown in Tab. 3, each module produces an improvement w.r.t. the baseline. The ablation study also proves that the modules are orthogonal to each other, giving consistent improvements when used alone as when used together. In particular, the attention module yields a 3% improvement on the baseline, and 3.5% on the final model, although it does not see the target domain at training time. Finally, the three modules together show an improvement of more than 16% on average over the baseline.

6. Conclusions

In this work we propose AdAGeo, a method to tackle the problem of cross-domain visual place recognition using only few unlabeled target images. The key improvements over previous architectures are due to an attention mechanism, and two orthogonal domain adaptation techniques. We extensively show the robustness of AdAGeo, especially when only few target images are available for domain adaptation at training time, being able to outperform current state of the art with two orders of magnitude less target images. Moreover, we propose a new dataset, called SVOX, which, extends Oxford RobotCar and can be used as a large scale multi-domain dataset for visual place recognition, presenting a realistic scenario for future research on the field.

References

[1] A. Anoosheh, T. Sattler, R. Timofte, M. Pollefeys, and L. V. Gool. Night-to-day image translation for retrieval-based localization. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5958–5964, 2019.

[2] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[3] S. Benaim and L. Wolf. One-sided unsupervised domain mapping. In *Advances in Neural Information Processing Systems 30*, pages 752–762. Curran Associates, Inc., 2017.

[4] J.-A. Bolte, M. Kamp, A. Breuer, S. Homoceanu, P. Schlicht, F. Huger, D. Lipinski, and T. Fingscheidt. Unsupervised domain adaptation to improve image segmentation quality both in the source and target domain. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[5] M. Chen, Z. Kira, G. Alregib, J. Yoo, R. Chen, and J. Zheng. Temporal attentive alignment for large-scale video domain adaptation. In *ICCV*, pages 6320–6329, 2019.

[6] Z. Chen, A. Jacobson, N. Sünderhauf, B. Upcroft, L. Liu, C. Shen, I. Reid, and M. Milford. Deep learning features at scale for visual place recognition. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[7] Z. Chen, F. Maffra, I. Sa, and M. Chli. Only look once, mining distinctive landmarks from convnet for visual place recognition. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9–16, 2017.

[8] T. Cohen and L. Wolf. Bidirectional one-shot unsupervised domain mapping. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[9] H. Fu, M. Gong, C. Wang, K¿ Batmanghelich, K. Zhang, and D. Tao. Geometry-consistent generative adversarial networks for one-sided unsupervised domain mapping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[10] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1180–1189, Lille, France, 2015. PMLR.

[11] S. Garg, N. Suenderhauf, and M. Milford. Lost? appearance-invariant place recognition for opposite viewpoints using visual semantics. In *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018.

[12] A. Gordo, J. Almazán, J. Revaud, and D. Larlus. End-to-end learning of deep visual representations for image retrieval. *International Journal of Computer Vision*, 124(2):237–254, 2017.

[13] A. Gretton, B. Sriperumbudur, D. Sejdinovic, H. Strathmann, S. Balakrishnan, M. Pontil, and K. Fukumizu. Optimal kernel choice for large-scale two-sample tests. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, 2012.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[16] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell. CyCADA: Cycle-consistent adversarial domain adaptation. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 1989–1998, 2018.

[17] W. Hong, Z. Wang, M. Yang, and J. Yuan. Conditional generative adversarial network for structured domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[18] S.-W. Huang, C.-T. Lin, S.-P. Chen, Y.-Y. Wu, P.-H. Hsu, and S.-H. Lai. AugGAN: Cross domain adaptation with gan-based data augmentation. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[19] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[20] H. J. Kim, E. Dunn, and J.-M. Frahm. Learned contextual feature reweighting for image geo-localization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[22] Y. Lou, Y. Bai, S. Wang, and L.-Y. Duan. Multi-scale context attention network for image retrieval. In *Proceedings of the 26th ACM International Conference on Multimedia*, 2018.

[23] W. Maddern, G. Pascoe, C. Linegar, and P. Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 2017.

[24] K. K. Nakka and M. Salzmann. Deep attentional structured representation learning for visual recognition. In *BMVC*, 2018.

[25] T. Naseer, G. L. Oliveira, T. Brox, and W. Burgard. Semantics-aware visual localization under challenging perceptual conditions. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2614–2620, 2017.

[26] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han. Large-scale image retrieval with attentive deep local features. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[27] N. Piasco, D. Sidibé, V. Gouet-Brunet, and C. Demonceaux. Learning scene geometry for visual localization in challenging conditions. *2019 International Conference on Robotics and Automation (ICRA)*, pages 9094–9100, 2019.

[28] H. Porav, W. Maddern, and P. Newman. Adversarial training for adverse conditions: Robust metric localisation using appearance transfer. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1011–1018, 2018.

[29] F. Radenović, G. Tolias, and O. Chum. Fine-tuning cnn image retrieval with no human annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7):1655–1668, 2019.

[30] P. Russo, F. M. Carlucci, T. Tommasi, and B. Caputo. From source to target and back: symmetric bi-directional adaptive gan. In *CVPR*, 2018.

[31] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, F. Kahl, and T. Pajdla. Benchmarking 6dof outdoor visual localization in changing conditions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[32] C. Shang, A. Palmer, J. Sun, K. Chen, J. Lu, and J. Bi. Vigan: Missing view imputation with generative adversarial networks. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 766–775, 2017.

[33] B. Sun and K. Saenko. Deep CORAL: Correlation alignment for deep domain adaptation. In *ECCV 2016 Workshops*, 2016.

[34] N. Sünderhauf, P. Neubert, and P. Protzel. Are we there yet? challenging seqslam on a 3000 km journey across all four seasons. In *Proc. of Workshop on Long-Term Autonomy, IEEE International Conference on Robotics and Automation (ICRA)*, page 2013, 2013.

[35] A. Torii, R. Arandjelović, J. Sivic, M. Okutomi, and T. Pajdla. 24/7 place recognition by view synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(2):257–271, 2018.

[36] Z. Wang, J. Li, S. Khademi, and J. van Gemert. Attention-aware age-agnostic visual place recognition. In *The IEEE*

International Conference on Computer Vision (ICCV) Workshops, Oct 2019.

[37] F. Warburg, S. Hauberg, M. Lopez-Antequera, P. Gargallo, Y. Kuang, and J. Civera. Mapillary street-level sequences: A dataset for lifelong place recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[38] R. Xu, G. Li, J. Yang, and L. Lin. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *ICCV*, pages 1426–1435, 2019.

[39] M. Zaffar, A. Khaliq, S. Ehsan, M. Milford, and K. McDonald-Maier. Levelling the playing field: A comprehensive comparison of visual place recognition approaches under changing conditions, 2019.

[40] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[41] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[42] J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, 2017.

[43] Y. Zhu, J. Wang, L. Xie, and L. Zheng. Attention-based pyramid aggregation network for visual place recognition. In *Proceedings of the 26th ACM International Conference on Multimedia*, 2018.

7. Appendix

Additional dataset details The main idea behind SVOX is to build a dataset that contains the RobotCar dataset [23], in order to test the accuracy of cross-domain visual geolocalization methods. To create the dataset we downloaded images from Google Street View, which provides 360° equirectangular panoramas at various resolutions. From each panorama we then cropped two rectangles at opposite sides, corresponding to the front and rear view of the car.

The original resolution of the images from the RobotCar dataset [23] is 1280x960, and we resized them to 512x384, keeping the original ratio of 4:3. We resized the images cropped from Google Street View panoramas to the same size, again keeping the same ratio.

Thanks to the Google Street View Time Machine we are able to download panoramas taken in the same location in different years. We chose to use images from the years of 2012 and 2014 as gallery and queries respectively, as these are the years with most panoramas in the Oxford area. Moreover, using gallery and queries taken in different years helps to ensure that methods that achieve accuracy must focus on long-term elements, instead of short-term or changing elements such as vegetation or scaffolding. The RobotCar dataset [23] was collected between 2014 and 2015, ensuring that the queries from RobotCar [23] are at least two years apart from the SVOX gallery. Some examples are shown in Fig. 6.

To build SVOX we chose a geographical area that would enclose the whole urban part of the city of Oxford. We then removed by hand images taken in the countryside, given the lack of buildings that are crucial to the geolocalization process. Moreover, we removed queries (from both SVOX and RobotCar [23]) which do not have a positive image within gallery, i.e. and image within 25 meters of distance. Finally we split SVOX in train, validation and test sets. As shown in Fig. 1 of the main paper, the RobotCar dataset [23] is included only in the train and test set, as it is intended to be used only as an unlabeled target dataset for domain adaptation, therefore not requiring a validation set.

Qualitative results In Figs. 7, 8 and 9 we show for each target scenario of RobotCar [23], some visualizations of top1 images retrieved by our method (AdAGeo) versus the best baseline (NetVLAD [2] + GRL [10]), which are trained and tested with the ResNet18 [15] as encoder.

(a) (b) (c) (d)

Figure 6: Examples of Oxford places at different times by means of Google Time Machine API. On the top row there are the images from 2012 used as gallery set, while on the bottom row there are the images from 2014 used as query set.

Figure 7: Comparison between our method and the best baseline, showing the top1 images retrieved for the target scenario Snow. The images with green border correspond with the ground truth, while the ones with a red border are wrong predictions.

(a)

(b)

Figure 8: Comparison between our method and the best baseline, showing the top1 images retrieved for the target scenarios Rain (a) and Sun (b). The images with green border correspond with the ground truth, while the ones with a red border are wrong predictions.

(a)

(b)

Figure 9: Comparison between our method and the best baseline, showing the top1 images retrieved for the target scenarios Night (a) and Overcast (b). The images with green border correspond with the ground truth, while the ones with a red border are wrong predictions.