

Visual Cross-View Metric Localization with Dense Uncertainty Estimates

Zimin Xia¹[0000-0002-4981-9514], Olaf Booij², Marco Manfredi²[0000-0002-2618-2493], and Julian F. P. Kooij¹[0000-0001-9919-0710]

¹ Intelligent Vehicles Group, Technical University Delft, The Netherlands

{z.xia,j.f.p.kooij}@tudelft.nl

² TomTom, Amsterdam, The Netherlands

{olaf.booij,marco.manfredi}@tomtom.com

Abstract. This work addresses visual cross-view metric localization for outdoor robotics. Given a ground-level color image and a satellite patch that contains the local surroundings, the task is to identify the location of the ground camera within the satellite patch. Related work addressed this task for range-sensors (LiDAR, Radar), but for vision, only as a secondary regression step after an initial cross-view image retrieval step. Since the local satellite patch could also be retrieved through any rough localization prior (e.g. from GPS/GNSS, temporal filtering), we drop the image retrieval objective and focus on the metric localization only. We devise a novel network architecture with denser satellite descriptors, similarity matching at the bottleneck (rather than at the output as in image retrieval), and a dense spatial distribution as output to capture multi-modal localization ambiguities. We compare against a state-of-the-art regression baseline that uses global image descriptors. Quantitative and qualitative experimental results on the recently proposed VIGOR and the Oxford RobotCar datasets validate our design. The produced probabilities are correlated with localization accuracy, and can even be used to roughly estimate the ground camera’s heading when its orientation is unknown. Overall, our method reduces the median metric localization error by 51%, 37%, and 28% compared to the state-of-the-art when generalizing respectively in the same area, across areas, and across time.

1 Introduction

Ground-to-aerial/satellite image matching, also known as cross-view image matching, has shown notable performance in large-scale geolocalization [37,40,15,8,16,25,48,34]. Usually, this global localization task is formulated as image retrieval. For each ground-level query image the system retrieves the most similar geo-tagged aerial/satellite patch in the database and uses the location of the center pixel in that patch as the location of the query. In practice, global localization can also be obtained by other means in outdoor robotics, such as temporal filtering or coarse GPS/GNSS [31,42,41], but can still have errors of tens of meters [42,41,4]. In this work, we therefore follow [31,42,41] by exploiting a coarse location estimate, and zoom

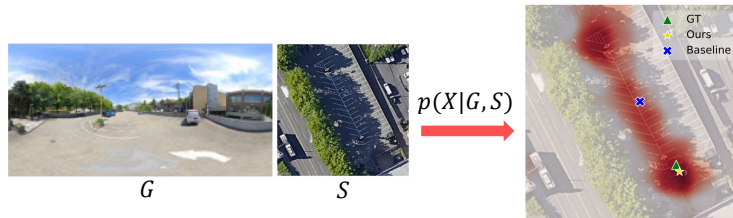


Fig. 1. Example of visual cross-view metric localization. Given a ground-level image G (left), and a satellite patch S (middle) with its local area, we aim to identify the location X within S where G was taken. Our method estimates a dense probability distribution over the satellite image. The resulting (log) probability heat map is overlaid in red on top of the satellite patch (right). Compared to the regression-based baseline that tends to roughly regress to the midpoint among multiple modes, our method captures the underlying multi-modal distribution. Our final predicted location, $\text{argmax}(p(X|G, S))$, is closer to the ground truth.

into fine-grained *metric localization* within a known satellite image, i.e. to identify which image coordinates in the satellite patch correspond to the location of ground measurement. We adopt the common assumption [16,25,48,27,34] of known orientation, e.g. the center of a ground panorama points north, though we will seek to loosen this restriction in our experiments and roughly estimate the camera’s heading too.

In vision, even though ground-to-ground metric localization is a well-studied task [1,13,6], so far in the cross-view setting, the only end-to-end approach that considers metric localization is the regression-based approach proposed in [48], which we will refer here to as *Cross-View Regression* (CVR) for simplicity. CVR tries to solve both the global coarse localization and local metric localization. As a result, its metric localization regressor is built on top of global image descriptors and might miss fine-grained scene information from the satellite image.

Rather than formulating visual cross-view metric localization as a regression task, we propose to produce a dense multi-modal distribution to capture localization ambiguities, and avoid regressing to the midpoint between multiple visually similar places, see Figure 1. To capture more spatial information, we compute multiple local satellite image descriptors rather than a single global one, and train these in a locally discriminative manner. We note that dense uncertainty output for localization was shown to be successful with range-sensing modalities, like LiDAR and Radar, for localization within top-down maps [44,3,38]. However, these methods are not directly applicable to monocular vision, as they rely on highly accurate depth information which images lack.

Unlike existing literature [37,40,8,16,25,48,27,34], we address local metric localization as a standalone task in visual cross-view matching, and make the following contributions: (i) We propose to predict a dense multi-modal distribution for localization, which can represent localization ambiguity. For this, we propose a new Siamese-like network that exploits multiple local satellite descriptors and

uses similarity matching in the fusion bottleneck. It combines the metric learning paradigm from image retrieval with dense probabilistic output via a UNet-style decoder, found previously only in range-based cross-view localization. (ii) We show that the produced distribution correlates with localization quality, a desirable property for outlier detection, temporal filtering, and multi-sensor fusion. Besides, we also achieved significantly lower median localization error than the state-of-the-art. (iii) We show our proposed method is robust against small perturbations on the assumed orientation, and that the model’s probabilistic output can even be used to classify a ground image orientation when it is unknown.

Our experiments use the recent large-scale VIGOR dataset for standalone cross-view metric localization to test generalization to new locations in both known and unknown areas. We also collect and stitch additional satellite data for data augmentation and metric localization on the Oxford RobotCar dataset, testing generalization to new measurements along the same route across time.¹

2 Related Work

We here review the works most related to visual cross-view metric localization.

Cross-view image retrieval is a special case of image retrieval. For place recognition [17], a majority of works [35,5,36] construct a reference database using ground-level images, but it is infeasible to guarantee the coverage of the images everywhere. Alternatively, satellite images provide continuous coverage over the world and are publicly available. Given this advantage, a series of approaches [15,40,37] have been proposed to solve large-scale geolocalization using ground-to-satellite cross-view image retrieval. CVM-Net [8] adopts the powerful image descriptor NetVLAD [2] to summarize the view-point invariant information for the cross-view image retrieval. In [16], the authors encode the azimuth and altitude of the pixels in the ground-level query to guide the ground-to-satellite matching. To explicitly minimize the visual difference between satellite and ground domains, various improvements have been proposed. SAFA [25] proposes to use a polar transformation to warp the satellite patch towards the ground-level panorama and uses attention modules to extract the specific features that are visible from both views. In [21,34], a conditional GAN [10] is used to generate synthetic satellite images from the ground-level panorama or to synthesize the panoramic street view from the satellite image to direct the cross-view matching. Instead of constructing a visually similar input, CVFT [27] tries to transport the features from the ground domain towards the satellite domain inside an end-to-end network. Some works [47,26,37] jointly estimate the orientation of the ground query during retrieval without any metric localization. Recently, transformers [43,46] are also used in cross-view image retrieval.

Limitations in cross-view image retrieval are also evident despite its increasing popularity for geolocalization. Recently, [48] points out that cross-view image retrieval methods assume that query ground images correspond to

¹ Models and code, plus extended data are available at
<https://github.com/tudelft-iv/CrossViewMetricLocalization>

the center of satellite patches in the database, and this assumption is not valid during test time. To break this assumption, [48] introduces a new cross-view matching benchmark VIGOR in which the ground images are not aligned with the center of satellite patches. Another limitation of retrieval is the trade-off between localization accuracy and computation or dataset density. To acquire meter-level localization accuracy, reference satellite patches often have a large overlap with each other, such as sampling the patch every 5m as done in [9,41].

Range sensing sensors-to-satellite metric localization received more attention than its visual counterpart. RSL-Net [31] localizes Radar scans on a known satellite image. This task is formulated as generating a top-down Radar scan conditioned on the satellite image using [10], and then comparing the online scan to synthetic scan for pose estimation. Later, this idea is extended to self-supervised learning [30]. In [29], the top-down representation of a LiDAR scan is compared to UNet [22] encoded satellite features for metric localization. The range information is crucial in representing the measurement in a top-down view.

LiDAR-to-BEV map metric localization is another frontier that benefited from the range sensing. Dense pixel-to-pixel matchable LiDAR and bird’s eye view (BEV) map embeddings can be learned by a deep network [3]. Localization becomes finding the position that has the maximum cross-correlation between two embeddings. Later work [38] shows that it is possible to localize the online LiDAR sweep on HD maps in a similar manner. Those works deliver a dense probabilistic output by formulating the localization task as a classification problem. This property is ideal in probabilistic robot localization [32], as it enables multi-sensor fusion and temporal filtering.

Visual ground-to-satellite metric localization cannot directly reuse the same architecture used to localize LiDAR scans in a BEV map, since an RGB ground image does not provide reliable depth information. Hence pixel-level dense comparison, such as cross-correlation, cannot be leveraged. [45] predicts ground-view semantics from aerial imagery for orientation estimation, and shows only qualitatively that metric localization is possible by comparing the predicted semantics across viewpoints. To the best of our knowledge, CVR [48] is the only end-to-end approach in the vision domain that attempts metric localization on a satellite patch. Given a ground-level query, it first retrieves the matched satellite patch and then regresses the offset between the ground image and satellite patch center. However, its offset regression is based on global feature descriptors, which might cause the regression head to miss detailed scene layout information, and it limits the output to uni-modal estimates. Plus, CVR lacks dense uncertainty estimation to identify ambiguous locations, or a way to filter out unreliable results. A concurrent work [24] perform unimodal localization and orientation estimation by warping features across views and solving an iterative optimization.

3 Methodology

In our work, we assume that a rough prior localization estimate is available, e.g. through GPS/GNSS, odometry, or some other robot-localization techniques [42,41,31].

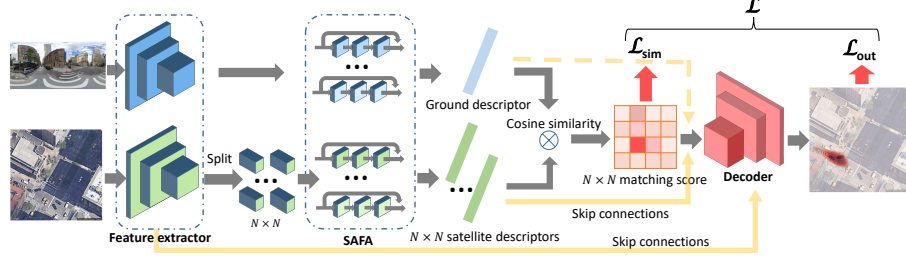


Fig. 2. An overview of the proposed cross-view metric localization architecture (trainable parts in bold). Dashed skip connection is optional, see ablation study. We overlay an exemplar output heat map on top of the input satellite image for intuition.

Given a ground-level image G and a top-down $L \times L$ satellite image S that represents the local area where G was taken, our metric localization objective is to estimate the 2D image coordinates $X \in [0, 1]^2$ within S that correspond to the ground location of the camera of G . Moreover, we aim for a dense probabilistic output to benefit a downstream sensor fusion task, similar to [3]. Note that in practice, G and S are often provided with their heading pre-aligned [48, 16], such that the center vertical line of G points in the up direction of S .

Both the baseline CVR [48] and our proposed method adapt a common cross-view image retrieval architecture [25]. This basic backbone is a Siamese-like architecture without weight-sharing. Both the ground and satellite input branches consist of a VGG [28] feature extractor. E.g. for the satellite branch, these features form a $L' \times L' \times 512$ volume. On the feature volume 8 Spatial-Aware Feature Aggregation (SAFA) modules [25] are applied, each generating a 512-dimensional vector, which is all concatenated. Each branch thus yields a single global $1 \times 1 \times 4096$ -dimensional descriptor. In an image retrieval task, this network would be trained through metric learning such that descriptors of matching (S, G) pairs are close together in the 4906-dimensional space.

Importantly, our proposed architecture and CVR make distinct choices on (1) the used descriptor representation for S , (2) how the descriptors are fused, (3) how the output head represents the localization result, and (4) consequently, the losses. We explain these choices for both methods in turn.

3.1 Baseline Cross-View Regression

The CVR method in [48] uses a single architecture for a two-step approach. First global localization is done through image retrieval by comparing descriptor G to descriptors of all known satellite patches. After retrieving satellite patch S , metric localization is performed using the already computed descriptors of both G and S . We employ CVR here for the metric localization task only, and therefore keep its proposed architecture, but will not train it for image retrieval. Focusing on metric localization only, our CVR baseline makes the following design choices:

Feature descriptors: CVR follows the image-retrieval concept of encoding the satellite and ground image each into a single image-global 4096-dimensional descriptor. Both descriptors are fed as-is to the fusion step. **Fusion:** CVR simply concatenates the two feature descriptors into a single 8192-dimensional vector. **Output head:** A multi-layer perceptron is used on the fused descriptors which outputs the relative 2D offset ΔX between G 's true location within S and the center $X_S = (0.5, 0.5)$ of the satellite patch, s.t. $X = X_S + \Delta X$. **Loss:** The standard L2 regression loss is used on the predicted offset and true offset.

We note that most of these choices follow from the need to use a single global descriptor for a whole satellite patch, as such descriptors are necessary for image retrieval. Our argument is however that if a localization prior is already available and global image retrieval is not necessary, this state-of-the-art architecture is sub-optimal for metric localization only compared to our proposed approach.

3.2 Proposed method

Our proposed architecture starts with a mostly similar Siamese-like backbone. The method overview is shown in Figure 2. It differs from CVR as follows:

Feature descriptors: Instead of building one image-global descriptor to represent S , we increase the top-down spatial resolution by splitting the satellite $L' \times L' \times 512$ feature volume along spatial directions into $N \times N$ sub-volumes, where N is a hyper-parameter. Now the 8 SAFA [25] modules are applied to each $L'/N \times L'/N \times 512$ sub-volume in parallel, resulting in an $N \times N \times 4096$ descriptor $g(S)$ for the satellite branch, shown as the green vectors in Figure 2. Let $g(S)^{ij}$ denote the i -th row j -th column of the satellite descriptor, $1 \leq i, j < N$. The ground image is still encoded as a single global 4096-dimensional descriptor $f(G)$, shown as the blue vector in Figure 2.

Fusion: To help distinguish different satellite image sub-regions, we compute the cosine similarity between $f(G)$ and each $g(S)^{ij}$, and use this similarity as a feature itself at this fusion bottleneck. This similarity computation results in a $N \times N \times 1$ matching score map M , thus $M^{ij} = \text{sim}(f(G), g(S)^{ij})$. To complete our fusion step, the M is concatenated to the satellite descriptors $g(S)$ through a skip connection, shown as the upper yellow solid arrow in Figure 2. Optionally, one could also concatenate $f(G)$ again into the fused descriptor (yellow dashed arrow), similar to CVR; we explore this in our experiments.

Output head: Rather than treating metric localization as a regression task, we seek to generate a dense distribution over the image coordinates. Such output enables us to represent localization ambiguities and estimate the (un)certainly of our prediction. Towards this, we feed the fusion volume to a decoder which can progressively up-sample the $N \times N$ matching map to higher resolutions. Akin to the UNet architecture [22], skip connections between satellite encoder and decoder are used to pass the fine-grained scene layout information to guide the decoding. Finally, a softmax activation function is applied on the last layer, and outputs a $L \times L \times 1$ heat map H , where each pixel $H^{u,v} = p(X \in c(u,v) | G, S)$ represents the probability of G being located within pixel area $c(u,v)$. This heat map is useful by itself, e.g. in a sensor fusion framework. For a single frame

estimate, we simply output the center image coordinates $\bar{c}[\cdot]$ of the most probable pixel, i.e. $X = \bar{c}[\operatorname{argmax}_{(u,v)} H^{u,v}]$.

Losses: A benefit of our framework is that we can add losses on both the final output and the fusion bottleneck. The full loss $\mathcal{L} = \mathcal{L}_{\text{out}} + \beta \times \mathcal{L}_{\text{sim}}$ is thus a weighted sum of the output loss, \mathcal{L}_{out} , and the bottleneck loss, \mathcal{L}_{sim} , where β is a hyper-parameter. We discuss each term next.

Since the output H is a discrete probability distribution that sums to one, we treat our task as a multi-class classification problem. \mathcal{L}_{out} is simply a cross-entropy loss over the $L \times L$ output cells. The ground truth is one-hot encoded as a heat map with the same $L \times L$ resolution and label 1 at the true location and 0 elsewhere. In practice, we will apply Gaussian label smoothing to the one-hot encoding of the output head, and tune the smoothing σ as part of the hyperparameter optimization.

To guide the model to already learn locally discriminative satellite descriptors at the fusion bottleneck, we apply the infoNCE loss [20] from contrastive representation learning [11], which can be seen as a generalized version of triplet loss [23] used in image retrieval in the case of multiple negative samples are presented at the same time,

$$\mathcal{L}'(ij^+) = -\log \frac{\exp(\operatorname{sim}(f(G), g(S)^{ij^+})/\tau)}{\sum_{i,j} \exp(\operatorname{sim}(f(G), g(S)^{ij})/\tau)}. \quad (1)$$

Here τ is a hyper-parameter introduced by [20], and its role is similar to the margin between positive and negative samples in triplet loss, and (ij^+) is the cell index of the positive satellite descriptor w.r.t. the ground descriptor.

We reuse the smoothed one-hot encoding from the output loss to allow multiple soft positives if the true location is near a cell border. We max-pool the $L \times L$ target map to the $N \times N$ resolution and renormalize it to generate ‘positiveness’ weights w_{ij}^+ for each cell $1 \leq i, j \leq N$. Our bottleneck loss is simply a weighted version of Equation (1), $\mathcal{L}_{\text{sim}} = \sum_{i,j} w_{ij}^+ \mathcal{L}'(ij)$.

4 Experiments

In this section, we first introduce the two datasets and evaluation metrics for our experiments. Then we motivate each of our design choices and provide a detailed ablation study. Finally, our model is compared to the baseline CVR approach [48] to show our advantage in metric localization in generalizing to new measurements in the same area, across areas, and across time.

4.1 Datasets

The first used dataset, **VIGOR** [48], contains geo-tagged ground-level panoramic images and satellite images collected in four cities in the US. Unlike previous cross-view image retrieval datasets [16,14,33,45], the satellite patches in VIGOR seamlessly cover the target area. Importantly, the ground-level panoramas are

not located at the center of satellite patches. Each satellite patch corresponds to $72.96m \times 72.96m$ ground area with a ground resolution of $0.114m$. The orientation of the satellite patch and ground panorama are aligned in a way that the vertical line at the center of the panorama corresponds to the north direction in the satellite patch. Typically, each patch has $\sim 50\%$ overlap with its neighboring patch in the North, South, East, and West direction. This means every ground image is covered by 4 satellite patches. If the ground image is at the center $1/4$ area of a satellite patch, the patch is denoted as “positive”, otherwise “semi-positive”. In practice, “positive” samples simulate the case that the global localization prior is more accurate, e.g. error $< \sqrt{2} \times 18.24m$ in the case of VIGOR. Similarly, “positive + semi-positive” samples would be a result of a coarser localization prior, e.g. error $< \sqrt{2} \times 36.48m$. During training, we include both “positives” and “semi-positives” samples. Our main evaluation will be based on positive samples, since it is representative for most real-world situations, e.g. localization prior from GNSS positioning in an open area or temporal filtering. For completeness, we also evaluate on “positive + semi-positive”, to showcase how the methods behave with a less certain localization prior, e.g. GNSS positioning in an urban canyon. We adopt the “same-area” and “cross-area” splits from [48] to test the model’s generalization in the same cities and across different cities. To find one set of hyper-parameters for both “same-area” and “cross-area”, we create a subset of the shared training data from New York as a smaller “tuning” split with 11108/2777 training/validation samples.

The second dataset, **Oxford RobotCar** [18,19] contains multi-sensor measurements from multiple traversals over a consistent route through Oxford collected over a year. The original dataset does not contain satellite images. To enable cross-view metric localization, we stitch the satellite patches provided by [42,41] and our additionally collected ones to create a continuous satellite map that covers the target area. We follow the same data split as in [41] to test how our method generalizes to new ground images collected at different time. In total, there are 17067, 1698, and 5089 ground-level front-viewing images in the training, validation, and test set respectively. The test set contains 3 traversals collected at later times of day than the training recordings. Benefiting from a full continuous satellite map, we randomly and uniformly sample satellite patches around the ground image locations during training for data augmentation. Each patch is rotationally aligned with the view direction of the ground image and has a resolution of $800pixels \times 800pixels$, which corresponds to $73.92m \times 73.92m$ on the ground. A fixed set of satellite patches is used for validation and testing. Each patch has a 50% area overlap with the closest neighboring patches. We pair each ground image with the patch at the closest center location to allow for mutual information between the satellite and ground front-facing views.

Similarly, during training, we also control the sampled locations to make sure the ground image locates inside the central area of the sampled satellite patch.

4.2 Evaluation metrics

To measure the localization error, we report the mean and median distance between the predicted location and ground truth location in meters over all samples. Note that the mean error can be biased by a few samples with large error, and including the median error provides a measurement more robust w.r.t. outliers. In practice, a localization method that operates on a single image frame can be extended to process a sequence of data using a Bayesian filter [9,41]. In such a setting, the estimated probability at the ground truth location plays an important role in accurately localizing over the whole path. Motivated by this, we include the probability at the ground truth pixel area as an additional metric. The baseline CVR method does not have any probability estimation on its output. Hence, we post-process the baseline output by assuming the regressed location is the mean of an isotropic Gaussian distribution, and we estimate the standard deviation of this Gauss on the validation set.

4.3 Hyper-parameters and ablation study

We first discuss our hyper-parameter choices, then investigate the main components in our proposed architecture. The weight β in our loss function is set to 10^4 , and the τ in infoNCE loss is set to 0.1, as done in [20]. The loss is optimized by Adam optimizer [12] with a learning rate of 1×10^{-5} , and the VGG feature extractors are pre-trained on ImageNet [7]. Our main model variations and hyper-parameters are now compared on the VIGOR “tuning” split.

We initially set $N = 8$ for the matching at the bottleneck. The infoNCE loss at the bottleneck is key to improve the final model output. The model trained with it achieves a much better mean error, $14.30m$, than the model trained without it, $19.25m$. Label smoothing with $\sigma = 4$ pixels further reduces the mean error to $13.39m$ (we tested $\sigma = 1, 2, 4, 8$). We use both in all future experiments.

Next, we study the influence of different resolutions $N \times N$ at the model bottleneck. When $N = 1, 2, 4, 8, 16$, the mean error is 19.62, 15.98, 15.23, **13.39**, 15.04 meters respectively (best in bold). With $N = 1$ no infoNCE loss is applied, and the decoder receives a single matching score concatenated with features from the satellite branch. Increasing N improves the spatial resolution at the model bottleneck. However, with larger N the decoder also operates on larger inputs but with fewer upsampling layers. We observe a balance at $N = 8$.

To further explore the role of metric learning at the bottleneck and the feature concatenation, we create four extra model variations in Table 1. Directly concatenating the ground with a single global satellite descriptor (see “1,S+G”) is akin to CVR’s fusion with a decoder head instead of regression, but this change alone does not perform well. The model does not work when the decoder operates on only a single channel map (“8,M”) without any context from the satellite patch. Increasing the satellite resolution is also still insufficient with only ground descriptors (“8,S+G”), the descriptors must also be trained to be locally discriminative. Interestingly, we do not observe any benefit from also concatenating the ground descriptor (“8,S+M+G”) to our default of satellite

Table 1. Fusion bottleneck, error on tuning split: “S” stands for satellite descriptors $g(S)$, “G” for ground descriptors $f(G)$, “M” for cosine similarity feature. Best in bold

N , descriptors	1,S+G	8,M	8,S+G	8,S+M+G	8,S+M
Mean error (m)	18.62	24.37	18.35	13.83	13.39

Table 2. Localization error on VIGOR. Best in bold. “Center-only” denotes using satellite patch center as the prediction. The term “Positives” stands for evaluation on positive satellite patches. “Pos.+semi-pos.” takes the mean over the results from the positive satellite patches and all semi-positive satellite patches

	<i>Same-area</i>				<i>Cross-area</i>			
	Positives		Pos.+semi-pos.		Positives		Pos.+semi-pos.	
<i>Error (m)</i>	Mean	Median	Mean	Median	Mean	Median	Mean	Median
Center-only	14.15	14.82	27.78	28.85	14.07	14.07	27.80	28.89
CVR [48]	10.55	9.31	16.64	13.82	11.26	10.02	18.66	16.73
Ours	9.86	4.58	13.45	5.39	13.06	6.31	17.13	7.78

descriptors with matching scores (“8,S+M”). In all next experiments, we fuse only the satellite descriptors and the matching score.

We note that to forward-pass an input pair from VIGOR on a Tesla V100 GPU, CVR uses 0.020s, and our best-performing model 0.034s (i.e. ~ 30 FPS).

4.4 Generalization in the same area / across areas

We now compare our method to CVR on the VIGOR splits for generalizing to unseen ground images inside the same area, and across areas. When tested on the “same-area” correctly retrieved samples, CVR trained for only regression has a better mean (-1.5m) and median (-1.1m) localization error than CVR trained for both retrieval and regression, as expected. From now on, we will always train CVR model for regression only as the baseline.

Metric error: The quantitative comparison against CVR on both VIGOR splits is summarized in Table 2. To highlight the value of conducting cross-view metric localization, we also include a “center-only” prediction which always outputs $X = (0.5, 0.5)$. Note that retrieval-only methods typically assume that the center of a satellite patch is representative of the true location.

When the ground image is compared to the positive-only satellite patches, our model reduces the median error by 51% over CVR when generalizing within the same area (4.58m vs 9.31m), and by 37% when generalizing across areas (6.31m vs 10.02m). Generally, our model improves over the baselines, but across areas, our mean error is higher than that of CVR. The error (cumulative) distribution in Figure 3 confirms that this is due to a few large-error outliers in our prediction. These outliers are a result of selecting a wrong mode, or of large uncertainty in our multi-modal output, whereas regression might pick an averaged location in the middle resulting in neither small nor very large errors. We will show below that our location’s probability can be used to detect such potential large error

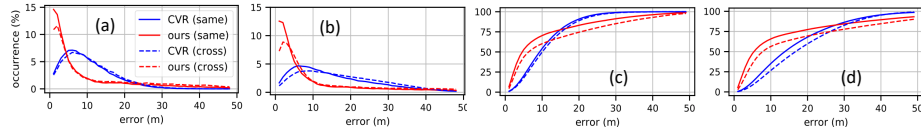


Fig. 3. Error distributions (plots a,b: regular, c,d: cumulative, a,c: positives, b,d: positives and semi-positives) on VIGOR, for same-area and cross-area experiments.

cases. This would aid an external sensor fusion module, which can also directly integrate the distribution to reduce the uncertainty through other measurements.

When ground images could be located further from the center, as in the “positive+semi-positive” test cases, there is less matchable visual information between the two views. In this case, the performance of both CVR and our model somewhat degenerates, though our model suffers less than CVR. Our mean and median errors are lower than CVR’s, both within the same area and across areas. Moreover, our method’s advantage in the median error further increases. In the Sup. Mat. we furthermore investigate the effect of using a CVR-like regression layer and loss on top of our dense output but find it hurts median performance.

Qualitative results: To intuitively understand where our advantage comes from, we provide qualitative examples of success and failure cases in Figure 4. In the context of image-based cross-view metric localization, there can exist multiple visually similar locations on the satellite image given a ground image. In such cases, it is important for the model to have the capability to express the underlying localization uncertainty. In our model, the uncertainty is already present at the model bottleneck, see Figure 4 top row. This distribution is up-sampled by the decoder and aligned with the observed environmental features, such as roads and crossings, resulting in the dense multi-modal uncertainty map. We emphasize that no explicit semantic map information, e.g. on road layout, was used during training. Since the regression-based baseline method forces the output to be a single location, it risks ‘averaging’ multiple similar locations and provides a wrong final estimate without any uncertainty information.

We argue that in practice our outliers are still more acceptable than CVR’s errors. When our model is uncertain about the exact location, our output heat map can be rather homogeneous. As shown in Figure 4 example 3, given a ground image taken on the road, our model assigns high probability to roads in the center and on the left. In this situation, the distance between our predicted location and the ground truth can be large. Instead, CVR tends to output the average between the visually similar areas, which can result in a location near the center but that is intuitively unreasonable, e.g. within some vegetation, even though it may have a smaller distance to the ground truth location.

Probability evaluation: Apart from the metric localization error, we will compare how well each model can predict the probability at the ground truth location. For CVR we estimate this assuming a fixed Gaussian error distribution, see Section 4.2. Table 3 reports both mean and median probabilities at the ground truth pixel. Our multi-modal approach outperforms the fixed er-



Fig. 4. Top: input ground images and matching score maps at the model bottleneck, bottom: input satellite image overlaid with outputs from CVR and our method. From left to right: 1: VIGOR, same-area, 2,3: VIGOR: cross-area, 4: Oxford RobotCar.

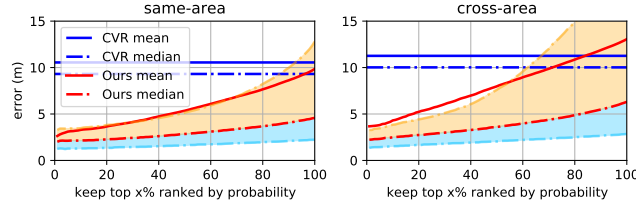


Fig. 5. Ranking the predictions using their probabilities on VIGOR “positives”. Red lines show our error statistics over the top $x\%$. Cyan/orange: error between median and 25% / 75% quantile line. As x decreases, only the more probable predictions are kept. Blue lines: CVR cannot rank predictions this way.

ror distribution for CVR. Importantly, the probability at our predicted location (the maximum in H) is correlated to its localization error. If we apply a rejection threshold to only keep the top $x\%$, predictions, we can reduce the expected error. See Figure 5 with the statistics over the top-ranked estimates. These properties are beneficial when the single frame localization results are temporally filtered or fused with other sensors.

Orientation: Till now, we have relied on a known orientation during test time, e.g. estimated in the preceding retrieval step [47,26,37] or by the sensor stack [39]. We here study our model’s robustness against orientation perturbations and ability to infer orientation when it is unknown *without retraining*. To test robustness, we uniformly sample angular noise from a range up to $\pm 20^\circ$ [27] to horizontally shift the ground panoramas (i.e. “rotate” the heading) at test time. As shown in Figure 6 left, the predicted location of our model remains stable under such noise.

Still, the model’s confidence is not invariant to orientation shift, as our prediction confidence can help classify a ground panorama’s unknown orientation. We rotate the ground panorama by multiples of 22.5° up to 360° , apply our model to each rotated panorama with the satellite patch, and collect all 16 acti-

Table 3. Probabilities at the ground truth pixel on VIGOR. Best in bold. The magnitude of the probabilities is low due to the normalization over the 512×512 grid. “Uniform” shows for reference the prob. at GT for a homogeneous map, $1/(512 \times 512)$

<i>Prob. at GT,</i>	Same-area		Cross-area	
<i>Positives</i>	Mean	Median	Mean	Median
Uniform	3.81×10^{-6}	3.81×10^{-6}	3.81×10^{-6}	3.81×10^{-6}
CVR [48]	1.55×10^{-5}	1.70×10^{-5}	1.57×10^{-5}	1.72×10^{-5}
Ours	2.93×10^{-4}	1.17×10^{-4}	1.54×10^{-4}	7.06×10^{-5}

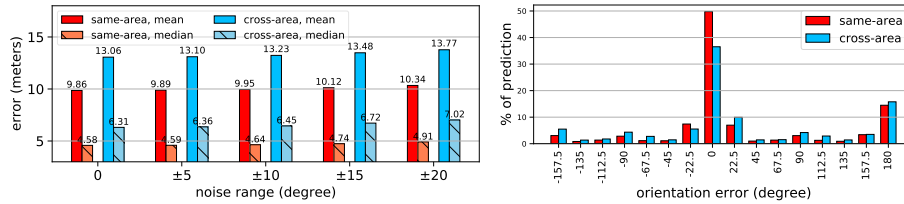


Fig. 6. Left: robustness of our model against small perturbations in orientation. Right: directly using our model to infer the unknown orientation.

vation maps before the final softmax operation. The classification output is the orientation of the map with the highest activation. As shown in Figure 6 right, for same and across areas, our model correctly classifies 50% and 37% samples into the true orientation class out of 16 classes. Most erroneous predictions have an error of 180° , corresponding to the opposite driving direction.

4.5 Generalization across time

Finally, we test how our method generalizes to new measurements collected at different times and days on the Oxford RobotCar dataset. For a comparison to cross-view image retrieval, we include “GeolocalRetrieval”, which was previously also proposed on the Oxford RobotCar dataset [41]. While regular image retrieval is trained to be globally discriminative, this method learns descriptors that are only discriminative for nearby satellite patches within a $50m$ radius, and thus assumes a localization prior during both training and testing, similar to our task. To increase its localization accuracy, we feed it a larger idealized dataset of satellite patches at more densely sampled locations (200+ patches in a $50m$ radius) including even patches centered on the actual test locations. Therefore GeolocalRetrieval could obtain zero meter error if it correctly retrieves the exact satellite patch at each test image location.

Table 4 shows the localization error among all included methods. As expected, with the idealized satellite patches, GeolocalRetrieval delivers lower error than “center-only”. However, metric localization methods (CVR and ours) show a clear advantage over GeolocalRetrieval. This highlights the benefit of conducting metric localization over simply densifying the dataset for retrieval.

Table 4. Localization error on Oxford RobotCar. Shown are the average \pm standard deviation of ‘mean’ and ‘median’ errors over 3 test traversals. Best results in bold. \star : uses the same training and test ground images, but more overlapping satellite patches to obtain finer localization through image retrieval only

<i>Error (meters)</i>	Mean	Median
Center-only	12.09 ± 0.02	12.65 ± 0.01
GeolocalRetrieval \star [41]	6.01 ± 0.68	4.62 ± 0.49
CVR [48]	2.29 ± 0.31	1.72 ± 0.21
Ours	1.77 ± 0.25	1.24 ± 0.10

Moreover, using our model for metric localization reduces the mean error by 23% and the median error by 28% compared to using CVR. Qualitatively, we again observe the benefit of expressing multi-modal distribution over the CVR’s regression even without the use of panoramic ground images, see Figure 4 example 4. The probability evaluation also aligns with the findings on VIGOR. Our probability at the ground truth pixel are consistently higher than that under CVR with its estimated error distribution over three test traversals. Averaged over three test traversals, the mean/median probability at the ground truth pixel for CVR are $1.67 \times 10^{-4}/1.89 \times 10^{-4}$, and for ours are $1.54 \times 10^{-3}/1.38 \times 10^{-3}$.

We also test classification of the orientation on this non-panoramic dataset. Instead of shifting the ground image, we now rotate the satellite patch 16 times with 22.5° , starting at 0° where north points in the vertical up direction. The orientation of a ground image is inferred by selecting the peak probability as we did for VIGOR. On three test traversals, 72.3%, 70.7%, and 70.6% of the test samples are predicted with the correct orientation out of the 16 possible directions. More details on orientation classification and the localization results with unknown orientation can be found in the Supplementary Material.

To summarize, also for test images at new days our method shows all-round superiority, similar to generalization within the same and across areas.

5 Conclusion

In this work, we focused on visual cross-view metric localization on a known satellite image, a relatively unexplored task. In contrast to the state-of-the-art regression-based baseline, our method provides a dense multi-modal spatial distribution. We studied the architectural design differences, and showed generalization to new measurements in the same area, across areas, and generalizing across time on two state-of-the-art datasets. Our method surpasses the state-of-the-art by 51%, 37%, and 28% respectively in the median localization error. In a few cases the multi-modal output yields higher distance errors, e.g. when an incorrect mode is deemed more probable. Still, our probabilities can be used to filter such large errors and have less risk of excluding the true location. We show that our method is robust against small orientation noise, and is capable to roughly classify the orientation from its prediction confidence. Future work will address temporal filtering and fine-grained orientation estimation.

Acknowledgements. This work is part of the research programme Efficient Deep Learning (EDL) with project number P16-25, which is (partly) financed by the Dutch Research Council (NWO).

References

1. Agarwal, P., Burgard, W., Spinello, L.: Metric localization using google street view. In: IEEE/RSJ IROS. pp. 3111–3118 (2015)
2. Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: NetVLAD: CNN architecture for weakly supervised place recognition. In: Proc. of IEEE/CVF CVPR. pp. 5297–5307 (2016)
3. Barsan, I.A., Wang, S., Pokrovsky, A., Urtasun, R.: Learning to localize using a lidar intensity map. In: CoRL (10 2018)
4. Ben-Moshe, B., Elkin, E., et al.: Improving accuracy of gnss devices in urban canyons. In: CCCG. pp. 511–515 (2011)
5. Chen, D.M., Baatz, G., Köser, K., Tsai, S.S., Vedantham, R., Pylvänäinen, T., Roimela, K., Chen, X., Bach, J., Pollefeys, M., et al.: City-scale landmark identification on mobile devices. In: Proc. of IEEE/CVF CVPR. pp. 737–744 (2011)
6. Clement, L., Gridseth, M., Tomasi, J., Kelly, J.: Learning matchable image transformations for long-term metric visual localization. *IEEE Robotics and Automation Letters* **5**(2), 1492–1499 (2020). <https://doi.org/10.1109/LRA.2020.2967659>
7. Deng, J., Dong, W., Socher, R., et al.: Imagenet: A large-scale hierarchical image database. In: Proc. of IEEE/CVF CVPR. pp. 248–255 (2009)
8. Hu, S., Feng, M., Nguyen, R.M., Hee Lee, G.: CVM-net: Cross-view matching network for image-based ground-to-aerial geo-localization. In: Proc. of IEEE/CVF CVPR. pp. 7258–7267 (2018)
9. Hu, S., Lee, G.H.: Image-based geo-localization using satellite imagery. *IJCV* pp. 1–15 (2019)
10. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proc. of IEEE/CVF CVPR. pp. 1125–1134 (2017)
11. Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised contrastive learning. *arXiv preprint arXiv:2004.11362* (2020)
12. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. *ICLR* (2014)
13. Lategahn, H., Stiller, C.: Vision-only localization. *IEEE Transactions on Intelligent Transportation Systems* **15**(3), 1246–1257 (2014). <https://doi.org/10.1109/TITS.2014.2298492>
14. Lin, T.Y., Belongie, S., Hays, J.: Cross-view image geolocalization. In: Proc. of IEEE/CVF CVPR. pp. 891–898 (2013)
15. Lin, T.Y., Cui, Y., Belongie, S., Hays, J.: Learning deep representations for ground-to-aerial geolocalization. In: Proc. of IEEE/CVF CVPR. pp. 5007–5015 (2015)
16. Liu, L., Li, H.: Lending orientation to neural networks for cross-view geo-localization. In: Proc. of IEEE/CVF CVPR. pp. 5624–5633 (2019)
17. Lowry, S., Sünderhauf, N., Newman, P., Leonard, J.J., Cox, D., Corke, P., Milford, M.J.: Visual place recognition: A survey. *IEEE Transactions on Robotics* **32**(1), 1–19 (2015)
18. Maddern, W., Pascoe, G., Linegar, C., Newman, P.: 1 year, 1000 km: The oxford robotcar dataset. *IJRR* **36**(1), 3–15 (2017)
19. Maddern, W., Pascoe, G., et al.: Real-time kinematic ground truth for the oxford robotcar dataset. *arXiv preprint: 2002.10152* (2020)
20. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018)
21. Regmi, K., Shah, M.: Bridging the domain gap for ground-to-aerial image matching. In: Proc. of IEEE/CVF ICCV. pp. 470–479 (2019)

22. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
23. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: Proc. of IEEE/CVF CVPR. pp. 815–823 (2015)
24. Shi, Y., Li, H.: Beyond cross-view image retrieval: Highly accurate vehicle localization using satellite image. In: Proc. of the IEEE/CVF CVPR (2022)
25. Shi, Y., Liu, L., Yu, X., Li, H.: Spatial-aware feature aggregation for image based cross-view geo-localization. In: NeurIPS. pp. 10090–10100 (2019)
26. Shi, Y., Yu, X., Campbell, D., Li, H.: Where am i looking at? joint location and orientation estimation by cross-view matching. In: Proc. of IEEE/CVF CVPR. pp. 4064–4072 (2020)
27. Shi, Y., Yu, X., Liu, L., et al.: Optimal feature transport for cross-view image geo-localization. In: Proc. of AAAI. pp. 11990–11997 (2020)
28. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
29. Tang, T.Y., De Martini, D., Newman, P.: Get to the point: Learning lidar place recognition and metric localisation using overhead imagery. *Robotics: Science and Systems* (2021)
30. Tang, T.Y., De Martini, D., Wu, S., Newman, P.: Self-supervised learning for using overhead imagery as maps in outdoor range sensor localization. *IJRR* **40**(12-14), 1488–1509 (2021)
31. Tang, T.Y., De Martini, D., Barnes, D., Newman, P.: Rsl-net: Localising in satellite images from a radar on the ground. *IEEE Robotics and Automation Letters* **5**(2), 1087–1094 (2020)
32. Thrun, S., Burgard, W., Fox, D.: Probabilistic robotics. MIT press (2005)
33. Tian, Y., Chen, C., Shah, M.: Cross-view image matching for geo-localization in urban environments. In: Proc. of IEEE/CVF CVPR. pp. 3608–3616 (2017)
34. Toker, A., Zhou, Q., Maximov, M., Leal-Taixe, L.: Coming down to earth: Satellite-to-street view synthesis for geo-localization. In: Proc. of IEEE/CVF CVPR. pp. 6488–6497 (June 2021)
35. Torii, A., Arandjelovic, R., Sivic, J., Okutomi, M., Pajdla, T.: 24/7 place recognition by view synthesis. In: Proc. of IEEE/CVF CVPR. pp. 1808–1817 (2015)
36. Torii, A., Sivic, J., Okutomi, M., Pajdla, T.: Visual place recognition with repetitive structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **37**(11), 2346–2359 (2015). <https://doi.org/10.1109/TPAMI.2015.2409868>
37. Vo, N.N., Hays, J.: Localizing and orienting street views using overhead imagery. In: ECCV. pp. 494–509. Springer (2016)
38. Wei, X., Bårnsan, I.A., Wang, S., Martinez, J., Urtasun, R.: Learning to localize through compressed binary maps. In: Proc. of IEEE/CVF CVPR. pp. 10316–10324 (2019)
39. Won, D., et al.: Performance improvement of inertial navigation system by using magnetometer with vehicle dynamic constraints. *Journal of Sensors* (2015)
40. Workman, S., Souvenier, R., Jacobs, N.: Wide-area image geolocalization with aerial reference imagery. In: Proc. of IEEE/CVF ICCV. pp. 3961–3969 (2015)
41. Xia, Z., Booi, O., Manfredi, M., Kooij, J.F.P.: Cross-view matching for vehicle localization by learning geographically local representations. *IEEE Robotics and Automation Letters* **6**(3), 5921–5928 (2021). <https://doi.org/10.1109/LRA.2021.3088076>

42. Xia, Z., Booi, O., Manfredi, M., Kooij, J.F.: Geographically local representation learning with a spatial prior for visual localization. In: ECCV Workshops. pp. 557–573. Springer (2020)
43. Yang, H., Lu, X., Zhu, Y.: Cross-view geo-localization with layer-to-layer transformer. In: NeurIPS. pp. 29009–29020 (2021)
44. Yin, H., Chen, R., Wang, Y., Xiong, R.: Rall: end-to-end radar localization on lidar map using differentiable measurement model. *IEEE Transactions on Intelligent Transportation Systems* (2021)
45. Zhai, M., Bessinger, Z., Workman, S., Jacobs, N.: Predicting ground-level scene layout from aerial imagery. In: Proc. of IEEE/CVF CVPR. pp. 867–875 (2017)
46. Zhu, S., Shah, M., Chen, C.: Transgeo: Transformer is all you need for cross-view image geo-localization. In: Proc. of the IEEE/CVF CVPR. pp. 1162–1171 (2022)
47. Zhu, S., Yang, T., Chen, C.: Revisiting street-to-aerial view image geo-localization and orientation estimation. In: Proc. of IEEE/CVF WACV. pp. 756–765 (2021)
48. Zhu, S., Yang, T., Chen, C.: Vigor: Cross-view image geo-localization beyond one-to-one retrieval. In: Proc. of IEEE/CVF CVPR. pp. 3640–3649 (2021)

Appendix

The content in this appendix is related to the main paper as follows:

- A More details on our used datasets in **Section 4.1 Datasets**.
- B Supplemental details on **Section 4.2 Evaluation metrics** about our post-processing on baseline CVR for generating probability estimation.
- C Additional results for **Section 4.4 Generalization in the same area / across areas**.
- D Additional results for **Section 4.5 Generalization across time**.
- E A detailed network diagram to supplement **Section 3.2 Proposed method**.

A Supplemental details on datasets

The original **VIGOR dataset** [48] does not provide a validation set for either the “same-area” nor the “cross-area” split. Therefore, we randomly split 20% of data from the training set as our validation set, used to determine the stopping epoch during training. The test set is kept as is and is used to report the evaluation results. Thereby, in the “same-area” split, there are 42087, 10522, and 52605 ground images in training, validation, and test set respectively. All three sets share the 90618 satellite images. In the “cross-area” setting, there are 41216, 10304, and 53694 ground images in training, validation, and test set respectively, and there are 44055 satellite images for training and validation in the first two cities, and 46563 satellite images for across-city testing.

Note that we do not use our validation set to find one set of hyper-parameters for both the “same-area” and “cross-area” experiments because the “same-area” validation set partially overlaps with the “cross-area” test set and vice versa. Hence, we make use of the “same-area” training image from New York to create a separate “tuning” set (11108 training and 2777 validation) to search for one set of hyper-parameters for all experiments and speed up the hyper-parameter search, see Figure 7. We will release our data splits.

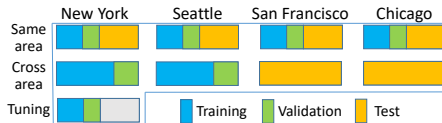


Fig. 7. Overview of our data split on VIGOR dataset

The original **Oxford RobotCar dataset** [18,19] does not contain satellite images. We make use of the satellite patches provided by [42,41]. We intend to conduct data augmentation such that, given a ground-level image, we can crop any satellite patch from a continuous satellite image that contains the ground



Fig. 8. Overview of our stitched continuous satellite image

image. However, the satellite patches from [42,41] do not provide enough coverage of the target area.

To enable the intended data augmentation, we collect additional satellite patches from Google Maps Static API², and stitch all satellite patches to create a continuous satellite map that covers the target area ($\sim 1.5km \times 1.8km$), see Figure 8. The geographical coordinates of each pixel in our continuous satellite image are known. Therefore, given the geographical location of a ground image, we can find its image pixel coordinates on the satellite image. We *will release* the raw data³ we collected and the code for stitching satellite patches.

As mentioned in **main paper Section 4.1**, we align the rotation between the ground image and satellite patch. Given a ground image and the heading of the camera we first crop a large satellite patch from the continuous map at the targeted cropping location and then rotate the satellite image such that the up direction in the cropped large patch corresponds to the viewing direction in the ground image. Then we crop the satellite patch with the required resolution from the rotated large patch. Note that, when testing classification of the orientation,

² <https://developers.google.com/maps/documentation/maps-static/overview>

³ We release the images in accordance with the “fair use” policy in Google Maps/Google Earth Terms of Service, version: Jan 12, 2022 (https://www.google.com/help/terms_maps/). The released data is for the reproducibility of scientific research and cannot be used for commercial purposes. Google remains the copyright of images.

main paper Section 4.5 second to the last paragraph, the satellite patches are rotated with a fixed set of angles. Thus the orientation-aligned patch will not appear in the set.

B Details on post-processing CVR localization

As we illustrated in **main paper Section 4.2**, to acquire the probability estimation from the baseline CVR method, we assume the regressed location from CVR is the mean of an isotropic Gaussian distribution and we estimate the standard deviation of this Gaussian distribution on the validation set.

Specifically, we record the error distribution on the validation set, and calculate the standard deviation SD using $SD = \sqrt{\frac{\sum_n e^2}{n}}$, in which e is the distance error of each ground image in the validation set and n is the number of ground images. On the VIGOR dataset, the estimated standard deviation is 12.36m on the same-area split and is 11.64m on the cross-area split. On the Oxford RobotCar dataset, the value is 3.36m.

C Additional results, generalization in the same area / across areas

Here, we provide more results for our experiments in **main paper Section 4.4**.

C.1 Extra model variations

We studied two extra variations of our proposed method.

Variation 1 has a multi-layer perceptron on top of our proposed model. It uses the output heat map to regress the relative 2D location offset between the ground camera’s location and the center of the satellite patch. Variation 2 drops the whole decoder and selects the location of the highest similarity score out of the 8×8 matching score map. The ground descriptor is then concatenated with the satellite descriptor at the selected location. The concatenated descriptor is used by a multi-layer perceptron to regress the relative 2D location offset between the ground camera’s location and the center of the small satellite patch corresponding to the selected location. Note that, both variations only regress a single location without dense uncertainty estimates. Similar to the post-processing on CVR’s output, see Section B, we assumed the regressed location is the mean of an isotropic Gaussian distribution and estimate the standard deviation on the validation set to acquire the probabilistic output.

In Table 5 and 6, we provide the comparison of all models on the VIGOR test splits. Variation 1 is better than CVR [48] all-round. This benefit might come from the larger model size and the dense feature before the final regression head. However, the regression-based variation 1 still has a significantly higher median error, which might be due to the regression head picking the midpoint between visually similar locations. Regarding variation 2, on the same-area split,

Table 5. Localization error on VIGOR satellite patches. Best in bold.

	Same-area				Cross-area			
	Positives		Pos.+semi-pos.		Positives		Pos.+semi-pos.	
<i>Error (m)</i>	Mean	Median	Mean	Median	Mean	Median	Mean	Median
Center-only	14.15	14.82	27.78	28.85	14.07	14.07	27.80	28.89
CVR [48]	10.55	9.31	16.64	13.82	11.26	10.02	18.66	16.73
Ours	9.86	4.58	13.45	5.39	13.06	6.31	17.13	7.78
Variation 1	8.01	6.29	12.26	9.20	9.62	7.57	14.25	11.42
Variation 2	10.69	5.88	13.40	6.07	12.49	6.92	15.29	7.30

Table 6. Probability at the ground truth location on VIGOR positive satellite patches. Best in bold.

<i>Prob. at GT,</i>	Same-area		Cross-area	
<i>Positives</i>	Mean	Median	Mean	Median
Uniform	3.81×10^{-6}	3.81×10^{-6}	3.81×10^{-6}	3.81×10^{-6}
CVR [48]	1.55×10^{-5}	1.70×10^{-5}	1.57×10^{-5}	1.72×10^{-5}
Ours	2.93×10^{-4}	1.17×10^{-4}	1.54×10^{-4}	7.06×10^{-5}
Variation 1	1.15×10^{-5}	8.40×10^{-6}	1.13×10^{-5}	9.36×10^{-6}
Variation 2	7.25×10^{-6}	6.89×10^{-6}	7.09×10^{-6}	6.86×10^{-6}

it performs worse than our model. This shows the advantage of having a decoder to process the matching score and satellite feature. However, on the cross-area split, our model performs slightly worse than variation 2. Possibly, given the more uncertain matching score map and features in unseen areas, the decoder does not know how to reduce this uncertainty effectively.

Importantly, variation 1 and 2 are not optimized for higher probability at the ground truth location. Thus, both variations have over one magnitude lower probability at ground truth than our model. We highlight again that probability estimation is crucial in both sensor fusion and temporal filtering. Last but not least, both variations miss the dense output to capture a multi-modal distribution. Thus, we conclude that our proposed model and problem formulation is better.

C.2 Probability evaluation

Supplementary to the cumulative statistics shown in **main paper Figure 5**, we show the noncumulative statistics of the predictions ranked by their probabilities in Figure 9.

C.3 Orientation

Additional to solely evaluating either localization or orientation, we also tested jointly evaluating both on the positive satellite patches. As in **main paper Section 4.4. Orientation**, the ground panorama is rotated by multiples of

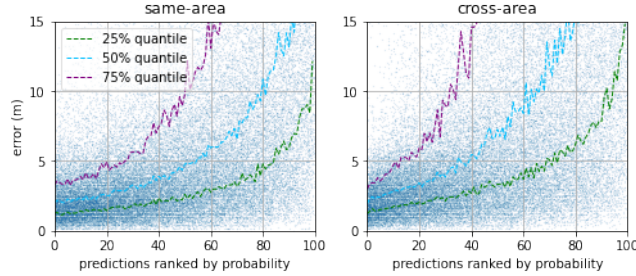


Fig. 9. Ranking the predictions using their probabilities on VIGOR “positives”.

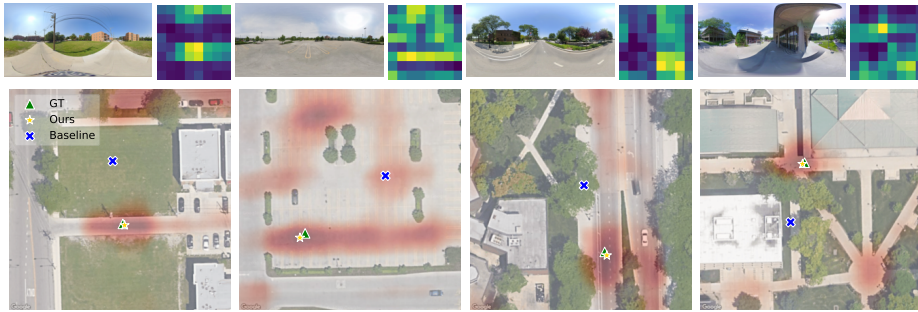


Fig. 10. Qualitative results on VIGOR dataset same-area split. Top: input ground images and matching score maps at the model bottleneck, bottom: input satellite image overlaid with outputs from CVR and our method.

22.5° up to 360° . We pair up each of the rotated panoramas with the satellite patch. Hence one mini-batch contains the same satellite image with 16 rotated ground panoramas. We collect all 16 activation maps and pass them into a single softmax operation. The location and corresponding orientation of the peak probability is then the localization and orientation estimation. Under this setting, the mean/median metric localization error increases from 9.86/4.58 (main paper Table 2 Same-area Positives columns) to 13.99/8.28 on the same-area test set, and from 13.06/6.31 (main paper Table 2 Cross-area Positives columns) to 18.18/14.61 meters on cross-area test set. This is a more challenging setting that leads to a noticeable decrease in performance.

C.4 Localization qualitative results

In addition, we provide more qualitative cross-view metric localization results in the same-area and cross-area splits in Figure 10 and 11.

As pointed out in the **main paper Section 4.4.**, our method has slightly more outliers than the CVR. In Figure 12, we show some of the failure cases where our localization error is higher than CVR. Importantly, as we stated in



Fig. 11. Qualitative results on VIGOR dataset cross-area split. Top: input ground images and matching score maps at the model bottleneck, bottom: input satellite image overlaid with outputs from CVR and our method.

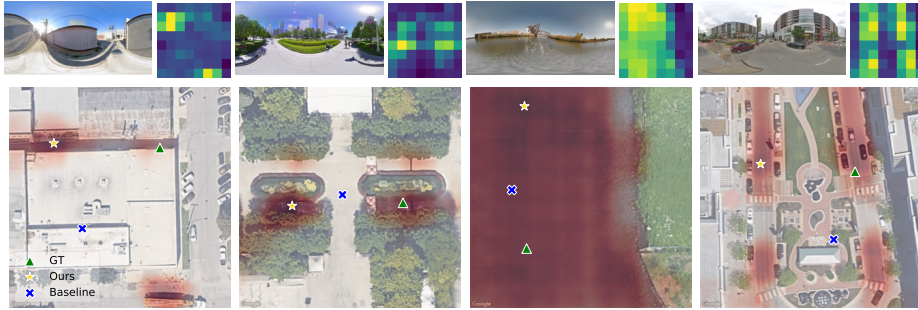


Fig. 12. Failure cases on VIGOR dataset. Top: input ground images and matching score maps at the model bottleneck, bottom: input satellite image overlaid with outputs from CVR and our method.

the main paper, instead of regressing to a location closer to the ground truth location as CVR, our method expresses the underlying uncertainty. This property is desirable in real-world applications since the ground truth location is also captured plus our prediction expresses the model’s underlying uncertainty.

Finally, we show qualitative results of localizing the ground image inside both positive and semi-positive satellite patches in Figure 13. The predicted location from our model is more consistent than that from CVR. This confirms our benefit of localizing against semi-positive satellite patches shown in the **main paper Section 4.4 Table 2**.

D Additional results, generalization across time

Next, we provide more results for **main paper Section 4.5**.

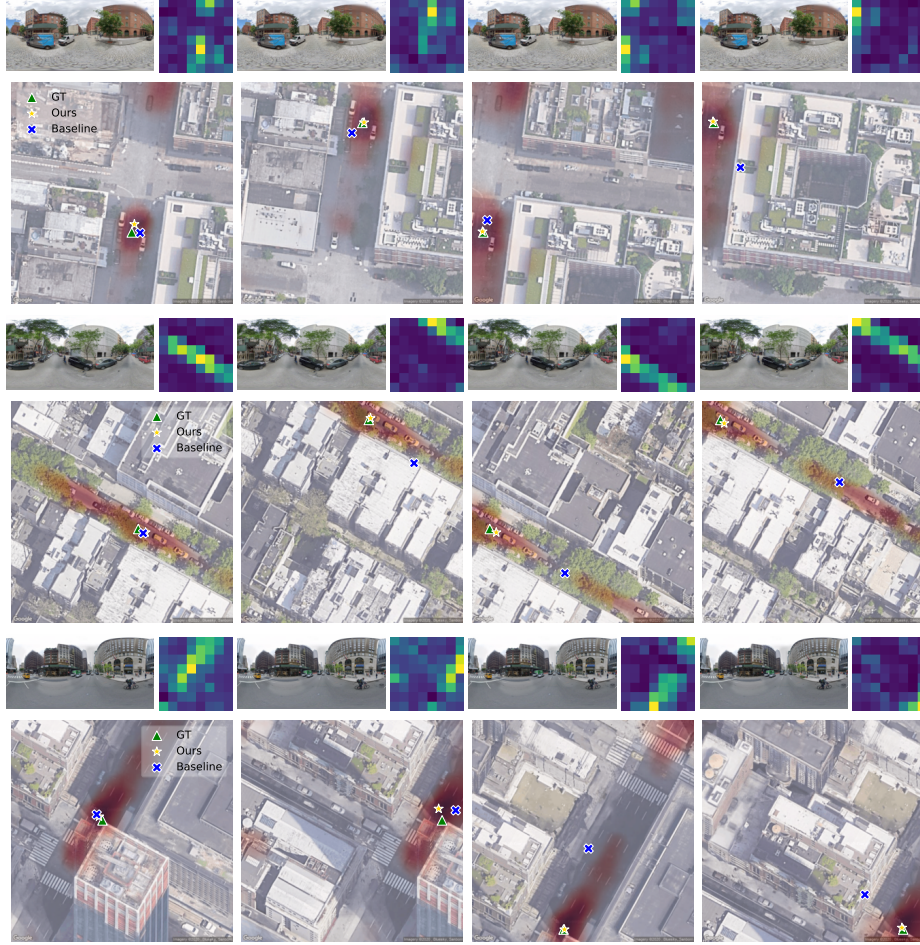


Fig. 13. Matching a ground image to the positive and all semi-positive (shifted) satellite patches on the VIGOR dataset. Each example includes an input ground image, a matching score map at our model bottleneck, and an input satellite image overlaid with outputs from CVR and our method. The first (from left to right) example in each row is the matching between the ground image and positive. The other three show the matching between the same ground image and 3 semi-positive satellite patches.

Table 7. Localization error on Oxford RobotCar test traversals. Last column: Average over all traversals

<i>Error (meters)</i>	Test 1	Test 2	Test 3	Average
CVR mean	1.88	2.64	2.35	2.29 ± 0.31
Ours mean	1.42	1.95	1.94	1.77 ± 0.25
CVR median	1.47	1.99	1.71	1.72 ± 0.21
Ours median	1.10	1.33	1.29	1.24 ± 0.10

Table 8. Probabilities at the ground truth pixel on Oxford RobotCar. Last column: Average over all traversals. For reference, the probability in a 512×512 uniform map is $3.81\text{e-}6$. Best in bold

<i>Prob. at GT</i>	Test 1	Test 2	Test 3	Average
CVR mean	1.78×10^{-4}	1.59×10^{-4}	1.65×10^{-4}	1.67×10^{-4}
Ours mean	1.76×10^{-3}	1.40×10^{-3}	1.46×10^{-3}	1.54×10^{-3}
CVR median	1.96×10^{-4}	1.81×10^{-4}	1.89×10^{-4}	1.89×10^{-4}
Ours median	1.69×10^{-3}	1.18×10^{-3}	1.28×10^{-3}	1.38×10^{-3}

D.1 Metric error and probability evaluation

As a supplement to the average metric localization error and average probability at the ground truth pixel over three Oxford RobotCar test traversals in the main paper, here, we provide the separate test results on each traversal in Table 7 and 8 for reference.

D.2 Orientation

Similar to Section C.3, we also report the localization error when jointly evaluating both localization and orientation estimation. We keep the same setting as in **main paper Section 4.5** that the satellite patches are rotated 16 times with 22.5° , starting at 0° where north points in the vertically up direction. In this case, the mean/median error increases from 1.42/1.10 (Table 7, Test 1) to 5.46/1.98, 1.95/1.33 (Table 7, Test 2) to 6.05/2.44, and 1.94/1.29 (Table 7, Test 3) to 5.77/2.35 meters on three test traversals.

Akin to the evaluation on VIGOR, **main paper Figure 6 right**, we also provide the histogram of orientation classification results on Oxford RobotCar. As shown in Figure 14, the most frequent wrong class is 180° . We show qualitative results of the orientation classification experiment in Figure 15. The model is trained with orientation-aligned satellite patches, which means if the ground image’s sight is along the road, the corresponding road in the satellite patch is along the vertical direction. In inference, our model exploits this information and assigns low probability when the road is not vertical. Importantly, our model is not a vertical road detector, since it also tries to match other objects, e.g. trees, across views, plus it can also differentiate the satellite patch rotated 180° from the correct orientation class in most cases. As shown in Figure 15, when the ground image indicates the vehicle is under the tree, our model tries to find the vegetation in the satellite view no matter the rotation angle.

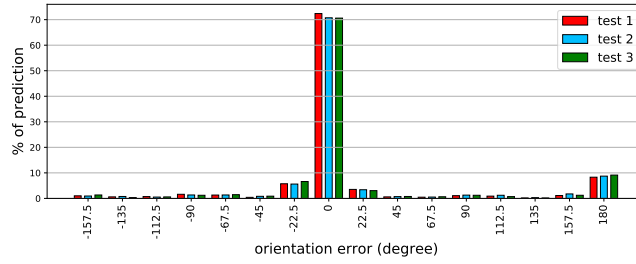


Fig. 14. An overview of orientation classification results on three test traversals from Oxford RobotCar.

D.3 Localization qualitative results

We provide more qualitative results on metric localization for both CVR and our method on three test traversals in Figure 16. Different traversals vary in time, weather, and lighting conditions. In Figure 17, we show the predictions from 3 test traversals at roughly the same location. Note that the headings of the ground camera are slightly different.

E Network Diagram

Finally, we provide our network diagram in Figure 18. We adopt the off-the-shelf SAFA module from [25] for image descriptor construction.

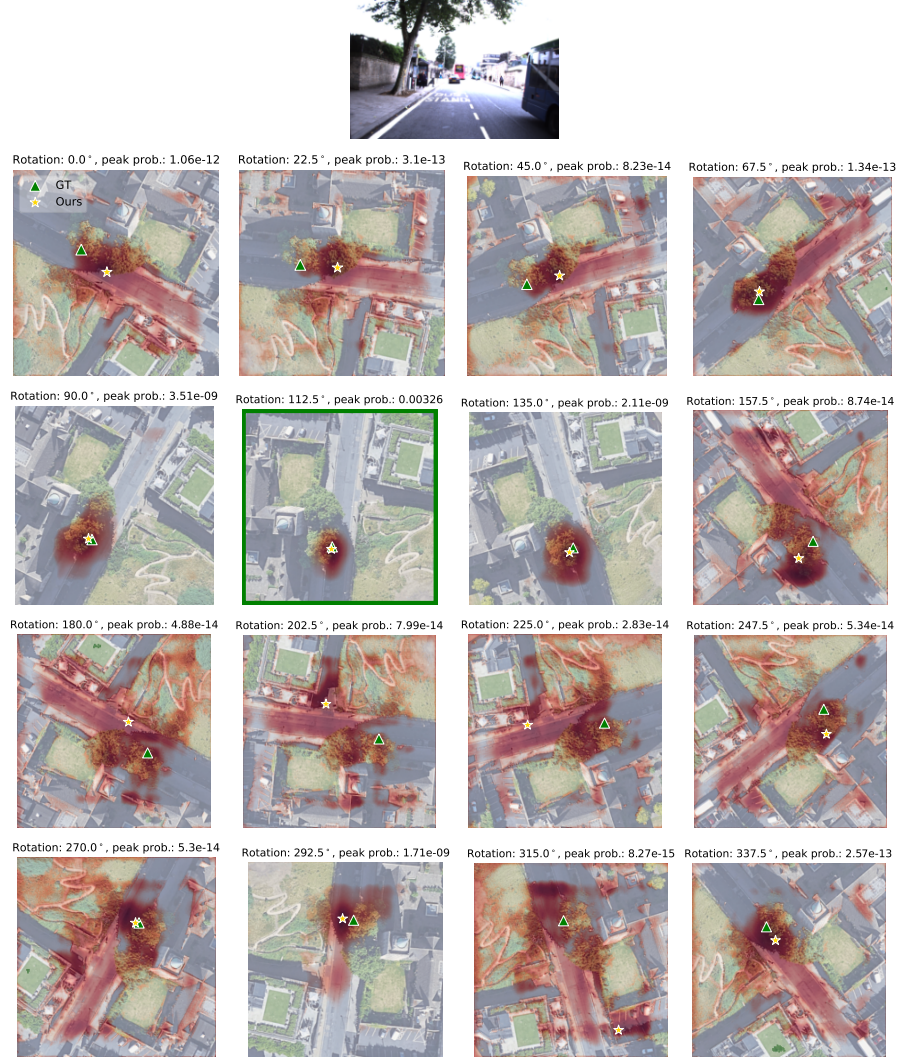


Fig. 15. Oxford RobotCar, the same ground image matched to 16 different rotated satellite patches. The ground truth orientation is 118° . The satellite patch with 0° rotation is the north-aligned satellite patch. The patch with a green box (112.5°) is the one prediction with the highest peak probability. The probabilities are generated by the softmax operation over a batch of inputs. For visualization, we re-scale the probability in each heat map.

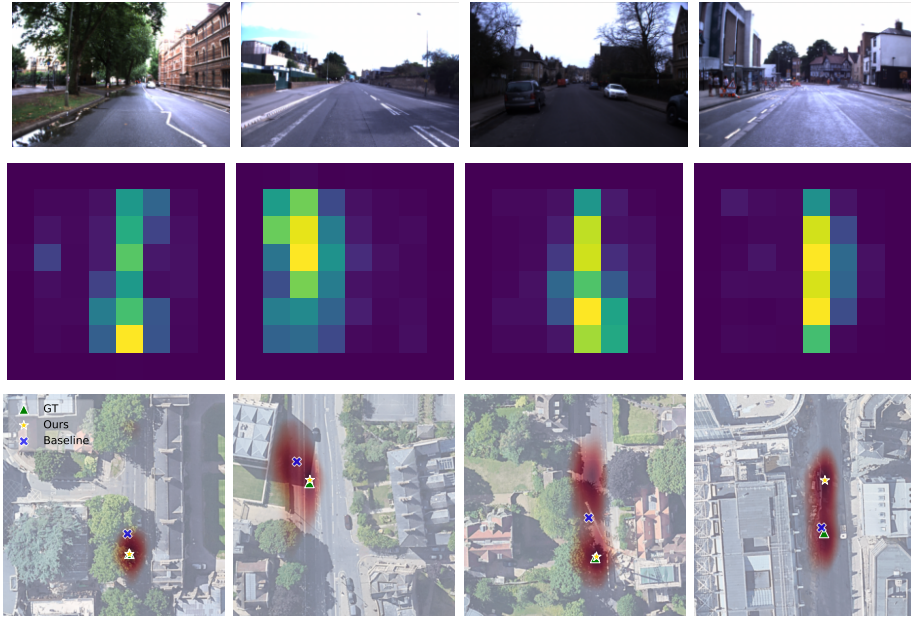


Fig. 16. Qualitative results on Oxford RobotCar dataset. Each column has an input ground image, a matching score map at our model bottleneck, and an input satellite image overlaid with outputs from CVR and our method. Column 1/2/3 (from left to right): good predictions in test traversal 1/2/3. Column 4: a failure case example.

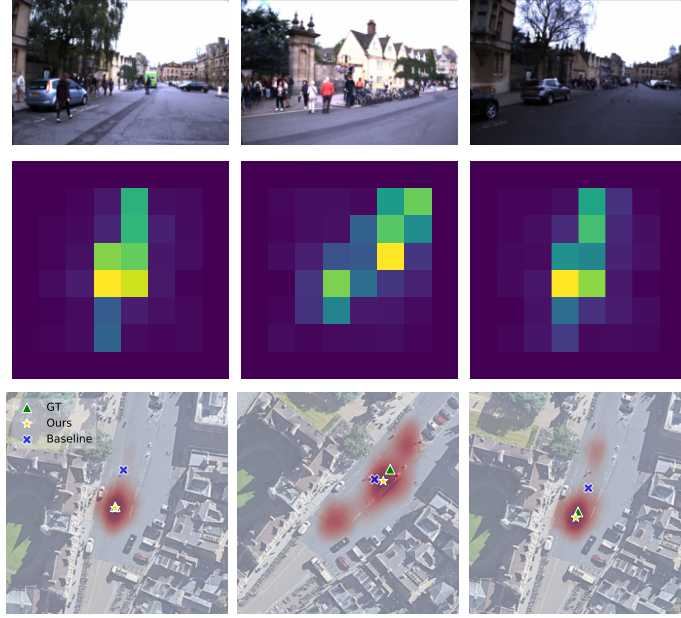


Fig. 17. Examples at a rough same location in 3 test traversals on Oxford RobotCar dataset. Ground images in different test traversals are collected at different times and weather. Each column has an input ground image, a matching score map at our model bottleneck, and an input satellite image overlayed with outputs from CVR and our method. Column 1/2/3 (from left to right): test traversal 1/2/3.

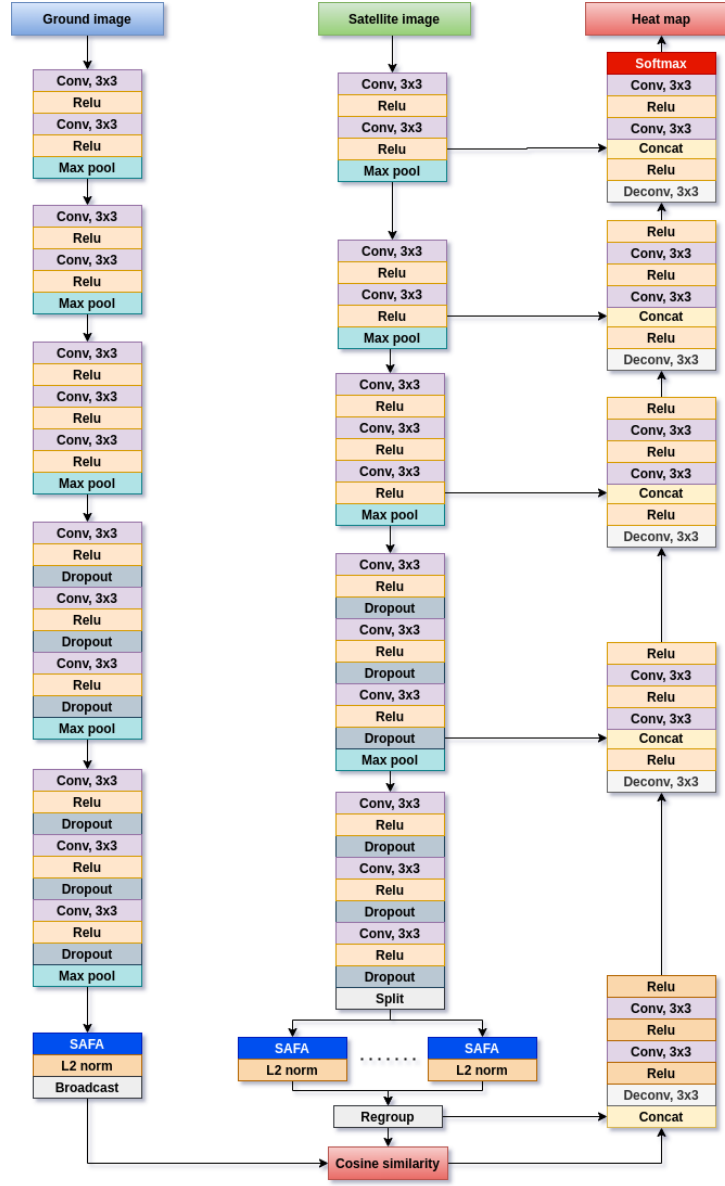


Fig. 18. Network diagram