ImPosing: Implicit Pose Encoding for Efficient Camera Pose Estimation

Arthur Moreau\textsuperscript{1,2}, Thomas Gilles\textsuperscript{1,2}, Nathan Piasco\textsuperscript{1}, Dzmitry Tsishkou\textsuperscript{1}, Bogdan Stanciulescu\textsuperscript{2}, and Arnaud de La Fortelle\textsuperscript{2}

\textsuperscript{1} IoV team, Paris Research Center, Huawei
\textsuperscript{2} Center for Robotics, MINES ParisTech, PSL University

Abstract. We propose a novel learning-based formulation for camera pose estimation that can perform relocalization accurately and in real-time in city-scale environments. Camera pose estimation algorithms determine the position and orientation from which an image has been captured, using a set of geo-referenced images or 3D scene representation. Our new localization paradigm, named Implicit Pose Encoding (ImPosing), embeds images and camera poses into a common latent representation with 2 separate neural networks, such that we can compute a similarity score for each image-pose pair. By evaluating candidates through the latent space in a hierarchical manner, the camera position and orientation are not directly regressed but incrementally refined. Compared to the representation used in structure-based relocalization methods, our implicit map is memory bounded and can be properly explored to improve localization performances against learning-based regression approaches. In this paper, we describe how to effectively optimize our learned modules, how to combine them to achieve real-time localization, and demonstrate results on diverse large scale scenarios that significantly outperform prior work in accuracy and computational efficiency.

Keywords: Camera pose estimation, implicit scene representation, real-time vehicle localization

1 Introduction

Positioning systems are a necessary component for automated vehicles, mobile robots and augmented reality applications. The precise ego-position inside of a known environment can be recovered in multiple ways using a wide range of sensors. Visual-based localization algorithms \cite{24} predict the 6 degrees of freedom camera pose of a query image, given a set of reference images captured in the environment and labeled with corresponding poses.

Intuitively, solving the camera pose estimation problem in large environments can be seen as the combination of several sub-tasks: extracting relevant image features, encoding 3D scenes content, retrieving the coarse localization (i.e. which area is depicted in the image) and then refining the precise camera pose. Most successful methods \cite{30,49} use this structure by connecting 2D image features to
3D points stored in memory with their corresponding descriptors. The resulting accuracy comes at the cost of a high memory footprint that increases linearly with the environment size. Direct learning-based methods circumvent this limitation by learning the entire task with a single neural network that directly regresses the camera pose from the image. These solutions are convenient for real-time deployment but entangle image features extraction and map memorization in the network’s weights, resulting in a limited accuracy, slow scene specific training and poor ability to adapt to large environment.

The common approach to represent scenes in computer vision is to use explicit representations such as point clouds, octrees, voxels or meshes. However, all of them store discrete information, while the underlying signal they represent is inherently continuous. As a consequence, these representations involve a trade-off between resolution and memory consumption. Recently, implicit neural representations, that connect scene coordinates to latent codes with a neural network, have shown great success for many computer vision tasks thanks to their ability to model continuous signals embedded into fixed-size network’s weights.

In this paper, we propose a new direct approach for camera pose estimation that perform better than pose regression methods by dissociating image and map encodings, while avoiding the computational cost and memory footprint of structure-based methods thanks to an implicit map representation. The core idea is to connect image and camera pose representations, which are learned separately by two distinct neural networks, in a common latent space. We use an implicit neural representation to encode a specific viewpoint in the scene (i.e. a 6-DoF camera pose) into a higher dimensional vector. With this formulation, the continuous representation of any camera pose in the scene (even a pose not observed in reference images) can be computed in a single network forward pass. We take advantage of this property to solve the localization task by searching the poses candidates which are the most similar to the learned image representation. To do so, we introduce a hierarchical sampling process able to retrieve the correct camera viewpoint using only a few batched queries on the pose encoder network. Our localization method, called Implicit Pose Encoding (ImPosing), provides real-time sub-metric localization performances that can be rapidly deployed on large areas.

We evaluate our system on a wide range of visual localization datasets, including several kilometers-scale road environments with challenging conditions (seasonal and appearance changes, limited training data). We observe that our method outperforms its regression-based competitors in terms of accuracy and training efficiency, especially in large-scale scenarios.

2 Related work

Image-based localization. Camera localization from raw images for real-time application or embedded deployment can be tackled by different classes of prior methods discussed below:
**Absolute pose regression** addresses the problem through end-to-end supervised regression between the input image and the camera pose using deep neural networks. PoseNet [13] is the pioneering work of these methods, and uses an encoder-decoder architecture where the encoder is a CNN pretrained on ImageNet and the decoder regresses the pose with fully connected layers. Since then, many architectural improvements have been proposed: notably, VidLoc [8] incorporates spatio-temporal constraints using consecutive video frames, AtLoc [44] uses an attention-based module before the regression step, Xue et al. [47] model the problem with graph neural networks, TransPoseNet [35] with transformers, and CoordiNet [21] uses a fully-convolutional architecture with geometrical inductive biases in the decoder layers. The main advantages of this class of methods are the compatibility with real-time deployment thanks to fast inference, low memory requirements and uncertainty estimation [14,21] which enables to filter out failure cases. The localization accuracy exhibited by absolute pose regression is limited compared to other methods [32], but has been observed to be highly dependent on the quantity and diversity of available training images, which can be improved with novel view synthesis [22]. ImPosing does not not explicitly regress the pose of the camera but learns a low dimensional latent representation of the image to efficiently and hierarchically explore the domain of poses encoded in its implicit map. In the following, we show through experiments that this formulation is better suited than absolute pose regression for localization in large urban area.

**Scene coordinate regression** learns the correspondence between the 2D image features and 3D scenes coordinates of observable image patches. It enables to retrieve the camera pose using projective geometry and by solving the Perspective-N-Points problem robustly with a RANSAC. Seminal work on scene coordinate regression rely on RGB-D images and use random forest to store the 3D coordinates [36]. Since then, the scene coordinate regression pipeline has been adapted to RGB images processed by fully convolutional networks [19,5]. The RANSAC step has been replaced by its differentiable counterpart DSAC [4], and ESAC [6] uses mixtures of expert to improve scaling to large environments. This class of methods exhibit higher accuracy than absolute pose regression and the efficiency enables real-time computation, however these methods are limited to relatively small environments [7]. By considering global image description instead of local features extraction, ImPosing is more consistent and faster to train and able to scale up to larger scenes, at the cost of minor loss in localization performances.

**Image retrieval algorithms for localization** solve a slightly different task: instead of computing a pose for the query image, these methods retrieve the closest geo-referenced image from the query within a large database [1,11,26,25]. The most similar image or the top ranked images are used to associate a pose to the query images. Poses averaging [42] or specific re-ranking based on GPS information [31] are used to improve the localization performances. Image retrieval methods use global image descriptors obtained by features maps pooling [1] or dense local features extraction [41] to represent in a discriminating way the scene
content and compare the images between them. These methods naturally scale to very large environments [33]. However, their accuracy is bounded by design by the density and diversity of reference images in the scene. Such a large reference database is difficult to collect and enlarging it linearly increases the memory footprint and the nearest neighbour search computational cost. This characteristic make image retrieval an appealing solution for visual place recognition but not convenient for camera pose estimation. The similarity comparison used in ImPosIng is inspired by the global features comparison presents in image retrieval except that the geo-referenced image database is replaced by our implicit map representation. Such representation permits to get rid of the two major drawbacks of retrieval methods: our scene layout becomes continuous and with a fixed memory size, independently of the number of collected images.

**Structure-based methods** compare local 2D image features to a 3D model to estimate the camera pose. 2D features are extracted from the query image using a CNN such as SuperPoint [9], and matched against the 3D model [29] to establish robust 2D-3D correspondences, that enable to compute the pose with PNP + RANSAC [28] or by Levenberg-Marquadt optimization [43,30]. The 3D model, usually represented as a point cloud of descriptors, enables to use geometric reasoning to solve the task. However, in large dynamic environments, highly accurate 3D reconstructions are challenging to make and memory demanding. ImPosIng does not rely on a 3D model of the scene and operate only with images and references poses.

**Implicit representations.** Neural networks performances highly depend on the representation used for a given space. Recent research has shown that using fully-connected neural networks to represent 3D data offers many benefits: the representation is continuous, memory-efficient and convenient to learn in any differentiable pipeline [46]. Successful examples of neural representation include 3D shapes [23,2], sound [37], static [20,38,12] and dynamic scenes [18]. Notable applications of implicit scene representations are Neural Radiance Fields [20] that enable photorealistic novel view synthesis from a sparse set of posed images and iMAP [39] that performs real-time RGB-D SLAM by optimizing a neural map.

In this paper, we aim to learn an effective representation of the map for camera relocalization inside of a target area. Our map can be characterized by a set of 6D camera poses: a 3D translation vector and a 3D rotation represented by quaternions, euler angles, axis-angle or rotation matrix. Zhou et al. [50] have demonstrated that none of these rotation representations are continuous, in the sense of continuously mapping coordinates to a latent space produced by a neural network, which is precisely our problem of interest. Zhu et al. [51] proposed a learned camera pose representation which is beneficial for view synthesis and pose regression. We propose to use a related camera pose representation optimized to be directly matched against the input image representation, enabling pose estimation by iterative sampling and evaluation of pose candidates.
3 Method

Our method, ImPosing, estimates the 6-DoF camera pose \((t, q) \in SE(3)\) of a query image \(I\), where \(t\) is a translation vector and \(q\) is a unit quaternion. We train our solution using a reference dataset of posed images \((I_k)\) collected in the target area and we do not make use of an additional 3D model of the scene.

The proposed algorithm, presented in figure 1, computes a vector that represents the image through the image encoder. Then, the camera pose is searched by evaluating initial pose candidates distributed across the map. Poses are processed by the pose encoder to produce a latent representation that can be matched against the image vector. A score is attributed to each pose candidate, based on distance between camera and candidate poses. High scores provide a coarse localization prior which is used to select new candidates. By repeating this process several times, our pool of candidates converges to the actual camera pose.

![Diagram](image)

**Fig. 1: Implicit pose encoding for hierarchical image localization.** A set of initial map signatures is compared to the image signature to determine the most probable localization of the camera. The similarity scores guides the selection of a new batch of pose candidates that are used to compute the new map signatures for the second refined localization step. This process is repeated multiple times to predict the final camera pose.

3.1 ImPosing localization process

This section describes the localization process step by step from the image to the final camera pose estimate as displayed in figure 1.

1. **Image encoder:** we extract a global image features vector \(f_I(I) \in \mathbb{R}^d\) from an input image \(I\) using our image encoder. The encoder architecture consists in a pretrained CNN backbone followed by a Global Average Pooling, and a fully-connected layer with \(d\) output neurons. The feature vector is one order of magnitude smaller than global image descriptors commonly used in image retrieval (we use \(d = 256\) whereas Revaud et al. [26] use \(d = 2048\)) in order to efficiently compare it to a large set of pose candidates at later steps.
2. **Initial pose candidates:** we sample $N$ camera poses $(p_n)_0$ from the set of reference poses (= training poses). By biasing the initial set of poses toward training poses, we avoid unreachable areas. Through this initial pose candidates selection, we introduce a prior for the localization process, similar to the anchors poses in [27] or regression methods that compute relative instead of absolute pose [10].

3. **Pose encoder:** we project a camera position with a 4 layers MLP $f_M : (t, q) \rightarrow v \in \mathbb{R}^d$ with ReLU activations on intermediate layers. Following Tancik et al. [40], we embed each component of the 7 dimensional pose $(tx, ty, tz, qx, qy, qz, qw)$ in higher dimension using Fourier features: $x \rightarrow (x, \sin(2kx), \cos(2kx))_{0 \leq k \leq 10}$, generating $F_i = 7 \times 11$ input features. We set other layers width to $d = 256$. Each set of pose candidates is computed in a single batched forward pass.

4. **Similarity scores:** we obtain a similarity score $s$ by computing the cosine similarity between $f_I(I)$ and $f_M(p)$ for each image-pose pair $(I, p)$. We add a ReLU layer after the dot product, such that $s \in [0, 1]$. Intuitively, we aim to learn high scores for poses candidates close to the actual camera pose. With this formulation, we can evaluate hypotheses on the camera pose and search for pose candidates with high scores. Formally, our score is defined by:

$$s(I, p) = \langle f_I(I), f_M(p) \rangle \frac{1}{\|f_I(I)\| \|f_M(p)\|} \mathbb{1}_{\langle f_I(I), f_M(p) \rangle > 0}$$ (1)

5. **Candidates proposer:** new poses $(p_n)_k$ are selected for the $k^{th}$ iteration based on scores obtained with poses $(p_n)_{k-1}$ at the previous iteration. First, we select the poses with top $B = 100$ higher scores $(h_i)_{0 \leq i < B} \subset (p_n)_{k-1}$. Then, new candidates are sampled from $(h_i)$ in a Gaussian Mixture Model with density:

$$P(x) = \sum_{i=1}^{100} \pi_i \mathcal{N}(x|h_i, v/k)$$

where $\pi_i = \frac{s(I, h_i)}{\sum_{l=1}^{100} s(I, h_l)}$. (2)

$v = [v_{tx}, v_{ty}, v_{tz}, v_{rx}, v_{ry}, v_{rz}]$ is the variance of the sampling process, a hyperparameter composed of a translation vector and Euler angles.

6. **Iterative pose refinement:** we repeat $K$ times the evaluation of pose candidates described in steps 3 to 5. After each iteration, the noise vector $v$ is divided by 2, such that new candidates are sampled closer to previous high scores. As a result, we can converge to a precise pose estimate in a kilometers-scale map while only evaluating a limited sparse set of poses. We evaluate each camera frame independently at each time step, however one could use localization priors from previous time steps to reduce the number of iterations in mobile robots and vehicles navigation scenarios. An example of selected poses at each iteration is shown in Fig. 2.
7. **Pose averaging**: our final camera pose estimate is a weighted average of the 256 pose candidates with higher scores, which exhibits better interpolation properties than selecting the best score pose. We use scores as weighting coefficients and 3D rotation averaging is implemented following [17].

Fig. 2: **Iterative candidates refinement.** At each $k$ step of the localization process, top scored poses are selected to sample the new candidate poses at step $k + 1$. From left to right: top scored poses at $k = 0$ to $k = 5$, yellow points are positions of the training example, blue arrows are pose candidates and red arrows are the selected poses among the candidates.

The entire inference procedure requires 1 forward pass on the image encoder and $K$ passes on the pose encoder.

### 3.2 Training procedure

We do not train the system by minimizing the error on the final camera pose estimate. Instead, we apply our loss function directly on the predicted scores. As a result, one training iteration provides supervision on the $K \times N$ image-pose pairs that contains more information than the single localization error. This property results in superior training efficiency than regression approaches (see 4.4). We define target scores $s_t$ based on translation and rotation distances between the camera pose $p_I = (t_I, q_I)$ and the candidate pose $p = (t, q)$:

$$s_t(I, p) = \phi(1 - \lambda_t \|t - t_I\|_2 - \lambda_r G(q, q_I)),$$

$$\phi(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{else} \end{cases}$$  \hspace{1cm} (3)

where $\lambda_t$ and $\lambda_r$ are hyperparameters and $G$ is geodesic distance, defined as the minimal angle between 2 rotations:

$$G(q_1, q_2) = \cos^{-1}\left(\frac{tr(M_{q_1}M_{q_2}^{-1}) - 1}{2}\right),$$  \hspace{1cm} (4)

$M_q$ being the 3D rotation matrix associated with rotation $q$.

We train $f_I$ and $f_M$ by computing scores between reference images and pose candidates sampled at $K$ different resolutions as described in section 3.1. For
training purpose, we add to initial poses an uniform noise sampled in $[-v, v]$ as we observed that it reduces overfitting. We also use poses associated with the top target scores in the candidates proposer, in addition with top predicted scores in order to guide training convergence in early iterations.

Finally, our optimization objective is:

$$L = \frac{1}{N} \sum_{k=0}^{K} \sum_{n=0}^{N-1} |s(I, p_{n,k}) - s_t(I, p_{n,k})|$$

(5)

An analogy can be made with content-based image retrieval [1,26]: global descriptors are usually trained using image triplets composed of a query image, a positive and a negative example. Positive samples are data close to the query, in metric or semantic domain depending on the final application, and negative samples are images with unrelated content to the query. Global descriptors can be trained by minimizing a triplet margin loss [1]. In our case, positive examples are the poses with a non-zero score whereas negative examples are candidates farther from the camera pose than an arbitrary threshold. Instead of binary classification (positive or negative example), we rank the relative importance of the positive samples according to their distance to the ground truth label.

## 4 Experiments

In this section, we compare our approach against recent methods on several datasets covering a wide range of autonomous driving scenarios in large scale outdoor maps. This task is highly challenging due to the dynamic part of outdoor environments (moving objects, illumination, occlusions, etc.). We verify that our formulation enables accurate localization in 9 different large outdoor scenes. Then we show that our method can be naturally extended to multi-map scenarios and we report results using this setup. We also compare the computational efficiency of our method with competitors and finally present an ablation study on several hyperparameters of ImPosing. Further localization results including more benchmarks, trajectories and video are provided in the supplementary material.

**Implementation details:** ImPosing is implemented in PyTorch. Images are computed at a small resolution $135 \times 240$. The image encoder uses a ResNet34 backbone pretrained on ImageNet. $N = 4096$ pose candidates are evaluated at each of the $K = 6$ refinement steps. The noise vector used for candidates sampling is set to $v = [8.0m, 0.2m, 8.0m, 1^\circ, 5^\circ, 1^\circ]$ where $y$ is the altitude axis (except for small scale scenes, see supplementary). We train the image encoder and pose encoder for 250 epochs with Adam optimizer at a constant learning rate of $1e^{-4}$. Details about datasets configuration are provided in supplementary materials.

**Baselines:** ImPosing is comparable to absolute pose regression, as a direct learning-based method, and it also shares similarities with image retrieval for localization (see section 2 and 3). We compare our proposal to CoordiNet [21] that
report state-of-the-art results for absolute pose regression on Oxford Dataset. We use previously reported results on this dataset, and our own implementation for other datasets. We replace the EfficientNet backbone by ResNet34 for a fair comparison with ImPosing. We also compare localization results of retrieval methods NetVLAD [1] (VGG16 backbone) and Revaud et al. [26] (GeM pooling, Resnet101 backbone) to our proposal thanks to publicly available implementations.\(^3\) We use full sized images to compute global image descriptors and cosine similarity for features comparison, then we perform pose averaging on poses of top 20 database images as in [32]. Scene coordinate regression [6,7] can not scale to large environments thus are not considered for evaluation. In contrast with ImPosing, structure-based methods [30,49] require a 3D model to operate that represent a very high memory footprint for large maps. This property, coupled with computational cost, makes the deployment of structure based method in embedded devices complicated and thus we do not consider them as competitors.

### 4.1 Single scene localization

**Oxford RobotCar** dataset [15] contains images recorded by a vehicle in Oxford for a period of over a year. We reproduce experiments commonly reported for learning-based methods [21,44,47]: we evaluate on the Loop and Full scenes, using only 2 sequences for training. Results are reported in Table 1.

First we observe that image retrieval performs better than pose regression on this benchmark. Previous learning-based methods struggle due to the low-data regime [21,22] and the decrease of the regression accuracy in large maps. Oxford city is an environment with rich features similar to visual place recognition training datasets, that make NetVLAD [1] and GeM [26] strong baselines in this scenario. ImPosing exhibits state-of-the art accuracy on Oxford Loop scene, as well as the best mean error in average. These results are obtained by reducing a lot the number of large failure cases that occur with prior methods.

We also observe that despite newly provided RTK ground truth provided by the authors [16], the reference poses are largely inaccurate in some areas. As a result, evaluation metrics are not significant at a centimeter level and models training might be impacted by this erroneous pose labels. For this reason, we conduct a benchmark on two recently released datasets with more reliable ground-truth.

**Daoxiang Lake** dataset [49] has been collected in a 12km loop in Beijing during 4 months. 8 recordings are available, we use 7 for training and 1 for testing with images from the front camera only (see supplementary for details). This scene contains the largest map and training dataset of our experiments. Median and mean errors are shown in Table 1. Daoxiang Lake is a more challenging dataset than Oxford because of repetitive areas with few discriminative features

\(^3\) https://github.com/Nanne/pytorch-NetVlad and https://github.com/naver/deep-image-retrieval
and various environments (urban, peri-urban, highways, nature, etc.). As a result, image retrieval performs worse than pose regression. ImPosing is way more accurate and exhibits a median error 4 times smaller than competitors.

Table 1: Localization error on Oxford RobotCar and Daoxiang Lake datasets.

| Dataset          | Pose regression | Image retrieval | ImPosing |
|------------------|-----------------|-----------------|----------|
|                  | CoordiNet [21]  | AtLoc [44]      | NetVLAD [1] | GeM [26] | ImPosing |
| Oxford Full      | 3.55m/1.1°      | 1.1m/5.3°       | 1.42m/1.4° | 1.36m/1.3° | 1.90m/1.3° |
| Oxford Loop      | 2.27m/0.9°      | 5.36m/2.1°      | 2.16m/1.1° | 2.39m/1.0° | 1.93m/1.0° |
|平均               | 2.91m/1.0°      | 8.23m/3.7°      | 1.79m/1.2° | 1.88m/1.1° | 1.92m/1.1° |
| Daoxiang Lake    | 6.82m/0.4°      | —               | 8.92m/0.8° | 27.13m/1.1° | 1.62m/0.3° |
|平均               | 25.18m/1.0°     | —               | 152.2m/15.5° | 328.8m/19.5° | 8.40m/0.5° |

4 seasons dataset [45] contains data recorded in Munich area in various scenes (city, residential neighborhoods, countrysides, ...) with heterogeneous seasonal conditions. We selected 6 scenes where at least 3 different recordings are provided: we use 1 for testing and others as training images (more details are provided in the supplementary). This benchmark is highly challenging due to extreme appearance changes between sequences, small data regime for some scenes, featureless environments (see illustration in supplementary materials) and kilometers-scale maps. Results are reported in table 2.

Table 2: Median localization error on 4Seasons dataset.

| Dataset details | Image retrieval | CoordiNet [21] | ImPosing |
|-----------------|-----------------|----------------|----------|
| Road length     | Images          | NetVLAD [1]    | GeM [26] | ImPosing |
| Neighborhood    | 2000            | 6              | 16520    | 0.72m/0.9° | 0.69m/0.9° | 0.74m/0.6° | 0.53m/0.7° | 0.82m/1.0° |
| Office loop     | 2600            | 5              | 20915    | 6.85m/3.0° | 6.39m/2.8° | 6.25m/1.5° | 0.99m/1.7° | 1.58m/1.3° |
| Countryside     | 6200            | 3              | 19804    | 32.24m/1.2° | 30.87m/1.3° | 47.33m/2.9° | 2.61m/0.9° | 5.46m/1.1° |
| Bus. campus     | 1000            | 2              | 6132     | 1.19m/1.3° | 1.96m/1.2° | 22.57m/6.0° | 1.16m/1.3° | 1.70m/1.6° |
| City loop       | 10000           | 2              | 17427    | 61.60m/3.5° | 317.4m/6.9° | 584.4m/14.4° | 5.32m/2.4° | 10.53m/2.5° |
| Old Town        | 4500            | 3              | 13959    | 3.45m/1.2° | 4.46m/1.6° | 50.83m/3.8° | 2.59m/1.2° | 3.71m/1.3° |
|平均             | -               | -              | 17.67m/1.8° | 60.30m/2.4° | 118.7m/4.9° | 2.2m/1.8° | 3.97m/1.5° |
| First, absolute localization accuracy is very heterogeneous between different scenes. We note that scenes with few training images are the most challenging. In particular, Countryside include navigation around fields and City Loop is a 10km map where the training dataset is composed of a winter sequence with snow and a rainy sequence with blur on camera lens. In these extreme cases, both pose regression and image retrieval fail to estimate reliable poses, whereas
ImPosing is able to provide a coarse localization. With sufficiently large training datasets, our method still exhibits the more precise pose estimation.

4.2 Multi-scene localization

Learning-based methods for relocalization require scene specific training, inducing heavy computation for potential deployment in several areas at a large scale. Recent work [3,34] has extended absolute pose regression to multi-scene scenarios. The core idea is to train a system with images from several maps while sharing image encoder parameters that could learn to extract features in a generic way. As our method separate image and map representation, ImPosing naturally extends to multi scenes scenarios. To adapt ImPosing to a multi-map scenario, we perform the following modifications: the image encoder backbone is shared between all maps, whereas one specific pose encoder is learned for each scene. We also learn scene specific parameters for the final linear layer of the image encoder, to facilitate image features projection to the desired map representation.

We train a multi-scene model on the 6 maps of 4 seasons [45]. Results are reported in Table 2. The model has been trained for 20 epochs only because of computational constraints, but still outperform all competitors except single scenes ImPosing models. While the convergence for a single scene is slower in the multimap formulation (but training a multiscene on \( n \) maps is faster than performing \( n \) different trainings on each map, see supplementary materials), it enables to localize in huge areas with minimal memory storage requirements (see section 4.3). More multi-scene results and discussion are provided in supplementary materials.

4.3 Efficiency comparison

Storage footprint. We report in figure 3 the required memory for absolute pose regression, image retrieval and ImPosing methods. Absolute pose regression and ImPosing have a fixed memory requirement independently of the database size as the map information are directly encoded in the networks weights. To estimate in-device storage requirement of retrieval methods, we consider the size of the database image descriptor (2048 for GeM and 4096 for NetVLAD) along with the size of the image encoder. Storage requirement of retrieval methods exceed 1 GB for large scale scene with more than 100k reference images. To estimate the memory requirement of structured based methods we consider the numbers given in [28]: a 3D model built from 4328 images is composed of 685k 3D points. If we consider one local descriptor of size 128 by 3D points, we can derive a linear rule to determine the 3D model size according to the number of reference images. This estimation is a rough approximation but we can estimate that structure based method require at least 3 times more storage capacity than image retrieval methods.
Fig. 3: **In-device memory usage.** We compare the memory storage requirement of several methods for multiple large scale scenarios. Structure-based method (black) and image retrieval baselines (blue and purple) need more memory as the size of the dataset grows whereas absolute regression methods and ImPosing (pink and cyan) storage requirement do not depend on dataset size. We also report median position error inferior at 10m in red. Best viewed in color.

Fig. 4: Training time comparison between pose regression [21] and ImPosing.

**Training time.** We evaluate the training efficiency of our method by comparing translation test accuracy between Imposing and CoordiNet [21] on Neighborhood scene where both methods have a comparable accuracy. We observe that our formulation converges almost 2 times faster than pose regression.

**Inference time.** Our method takes 41ms to provide a camera pose estimate from a query image using a NVIDIA RTX 2080 GPU, enabling real-time deployment at 25 FPS.

**Summary.** ImPosing proposes the best trade-off on training time, memory footprint and inference speed of all the presented methods: it is a fixed-size solution compact as pose regression network with faster convergence property.
4.4 Ablation study

This section reports the influence of several hyperparameters on the localization accuracy of ImPosing. We evaluate the number of refinement steps $K$, the number of pose candidates $N$ and the number of best candidates used for pose averaging. We use our ImPosing model trained on Daoxiang Lake and change the parameters at test time.

![Graphs showing median localization errors depending on number of refinements, pose candidates, and final number averaged poses.](image)

Fig. 5: Median localization errors depending on number of refinements, pose candidates, and final number averaged poses.

Increasing the number of refinements and candidates improves localization accuracy, at the cost of a higher computational cost. We use a reasonable trade-off with $K = 6$ and $N = 4096$ as our default setup. We observe that pose averaging has a positive impact on accuracy, but the number of selected candidates is not critical.

5 Discussion

**What does the pose encoder learn?** Intuitively, we want our implicit representation to learn a continuous mapping between a given point of view in the map and the visual content observable at this location. The training data of the pose encoder is the set of all camera poses of interest in the map (i.e. close to a reference pose). This set of camera poses is a manifold of $SE(3)$ which is embedded by the pose encoder in a higher dimensional space. By defining target scores based on translation and rotation distances, we constraint the latent space to keep the structure of $SE(3)$. As ImPosing outperforms the pose regression approach, it proves that the pose encoder does not learn the identity function but a useful representation. We attempt to visualize the structure of this latent space by computing a PCA of the 256 dimensional representation of training poses. We associate the 3 principal components to a RGB color and display it in a 2D map in figure 6. We observe that our pose encoder learns a smooth representation of the map, where close representations share similar visual content (the portions of road in a same direction).

**Benefits, disadvantages and future work.** Our method keeps the main advantages of direct learning-based methods: we obtain the pose prediction efficiently with neural networks inference, we do not use a 3D model of the scene but only images labelled with corresponding poses and the memory footprint during online localization does not increase with the environment size but stays...
fixed and compact thanks to the implicit representation. We observe that accuracy our learning-based method highly depends on the quantity of training data available. Imposing is less accurate in small environments where geometrical reasoning with local features perform better. Similar to regression, our method does not extrapolate to camera positions far from training examples. However, recent approaches has shown that direct learning based methods can circumvent these limitations with uniformly distributed synthetic datasets [22]. The new paradigm we propose for camera pose estimation could be improved in many ways. It includes exploring better architectures for the pose encoder, inspired from recent work on coordinate-based representations [51]. Another interesting direction is to extend the implicit map representation to local features instead of global image signatures, by finding a way to represent implicitly a 3D model. One solution could be to use ray-tracing approaches similar to NeRF [20], but the computational cost is still prohibitive for real-time requirements [48].

6 Conclusion

We have proposed a new formulation for camera pose estimation that perform state-of-the-art accuracy for real-time localization in large environments. By using an implicit representation of observable camera poses in the map, we embed visual content and map structure in a common high dimensional manifold well suited for localization. We have shown that with a simple pose candidates sampling procedure, we are able to efficiently explore the implicit map representation to estimate the absolute pose of an image. Our proposal can be directly applied in autonomous driving systems, by providing an efficient and accurate image-based localization algorithm that can operate at large scales. We believe that, beyond our work, implicit scene representations, by their ability to model complex continuous signals in a fixed size neural network, are a promising research direction for camera pose estimation.
References

1. Arandjelović, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: NetVLAD: CNN architecture for weakly supervised place recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (2016)
2. Atzmon, M., Lipman, Y.: Sal: Sign agnostic learning of shapes from raw data. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
3. Blanton, H., Greenwell, C., Workman, S., Jacobs, N.: Extending absolute pose regression to multiple scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 38–39 (2020)
4. Brachmann, E., Krull, A., Nowozin, S., Sotton, J., Michel, F., Gunhold, S., Rother, C.: Dsac-differentiable ransac for camera localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6684–6692 (2017)
5. Brachmann, E., Rother, C.: Learning less is more - 6d camera localization via 3d surface regression. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
6. Brachmann, E., Rother, C.: Expert sample consensus applied to camera re-localization. In: ICCV (2019)
7. Brachmann, E., Rother, C.: Visual camera re-localization from RGB and RGB-D images using dsac. In: IEEE Transactions on Pattern Analysis and Machine Intelligence. pp. 1–1 (2021).
8. Clark, R., Wang, S., Markham, A., Trigoni, N., Wen, H.: Vidloc: A deep spatio-temporal model for 6-dof video-clip relocalization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6856–6864 (2017)
9. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: Self-supervised interest point detection and description. In: CVPR Deep Learning for Visual SLAM Workshop (2018), http://arxiv.org/abs/1712.07629
10. Ding, M., Wang, Z., Sun, J., Shi, J., Luo, P.: Camnet: Coarse-to-fine retrieval for camera re-localization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2871–2880 (2019)
11. Gordo, A., Almazan, J., Revaud, J., Larrus, D.: End-to-end learning of deep visual representations for image retrieval. IJCV (2017)
12. Jiang, C.M., Sud, A., Makadia, A., Huang, J., Nießner, M., Funkhouser, T.: Local implicit grid representations for 3d scenes. In: Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2020)
13. Kendall, A., Grimes, M., Cipolla, R.: Posenet: A convolutional network for real-time 6-dof camera relocalization. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 2938–2946 (2015)
14. Kendall, A., Cipolla, R.: Modelling uncertainty in deep learning for camera re-localization. In: 2016 IEEE international conference on Robotics and Automation (ICRA). p. 4762–4769. IEEE Press (2016).
15. Maddern, W., Pascoe, G., Linegar, C., Newman, P.: 1 Year, 1000km: The Oxford RobotCar Dataset. vol. 36, pp. 3–15 (2017)
16. Maddern, W., Pascoe, G., Gadd, M., Barnes, D., Yeomans, B., Newman, P.: Real-time kinematic ground truth for the oxford robotcar dataset (2020)
17. Markley, F.L., Cheng, Y., Crassidis, J.L., Oshman, Y.: Averaging quaternions. Journal of Guidance, Control, and Dynamics 30(4), 1193–1197 (2007)
18. Martin-Brualla, R., Radwan, N., Sajjadi, M.S.M., Barron, J.T., Dosovitskiy, A., Duckworth, D.: NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In: CVPR (2021)
19. Massiceti, D., Krull, A., Brachmann, E., Rother, C., Torr, P.H.: Random forests versus neural networks — what’s best for camera localization? In: 2017 IEEE International Conference on Robotics and Automation (ICRA). pp. 5118–5125 (2017). https://doi.org/10.1109/ICRA.2017.7989598
20. Mildenhall, B., Srinivisan, P.P., Tanick, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020)
21. Moreau, A., Piasco, N., Tsishkou, D., Stanciulescu, B., de La Fortelle, A.: Cooordinet: uncertainty-aware pose regressor for reliable vehicle localization. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2229–2238 (2022)
22. Moreau, A., Piasco, N., Tsishkou, D., Stanciulescu, B., de La Fortelle, A.: Lens: Localization enhanced by nerf synthesis. In: Proceedings of the 5th Conference on Robot Learning. Proceedings of Machine Learning Research, vol. 164, pp. 1347–1356. PMLR (2022), https://proceedings.mlr.press/v164/moreau22a.html
23. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
24. Piasco, N., Sidibé, D., Demonceaux, C., Gouet-Brunet, V.: A survey on visual-based localization: On the benefit of heterogeneous data. Pattern Recognition 74, 90–109 (2018)
25. Piasco, N., Sidibé, D., Gouet-Brunet, V., Demonceaux, C.: Learning scene geometry for visual localization in challenging conditions. In: 2019 International Conference on Robotics and Automation (ICRA). pp. 9094–9100 (2019). https://doi.org/10.1109/ICRA.2019.8794221
26. Revaud, J., Almazan, J., Rezende, R., de Souza, C.: Learning with average precision: Training image retrieval with a listwise loss. In: ICCV (2019)
27. Saha, S., Varma, G., Jawahar, C.: Improved visual relocalization by discovering anchor points. arXiv preprint arXiv:1811.04370 (2018)
28. Sarlin, P.E., Cadena, C., Siegwart, R., Dymczyk, M.: From coarse to fine: Robust hierarchical localization at large scale. In: CVPR (2019)
29. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: SuperGlue: Learning feature matching with graph neural networks. In: CVPR (2020). https://arxiv.org/abs/1911.11763
30. Sarlin, P.E., Unagar, A., Larsson, M., Germain, H., Toft, C., Larsson, V., Pollefeys, M., Lepetit, V., Hammarstrand, L., Kahl, F., Sattler, T.: Back to the Feature: Learning robust camera localization from pixels to pose. In: CVPR (2021)
31. Sattler, T., Havlena, M., Schindler, K., Pollefeys, M.: Large-scale location recognition and the geometric burstiness problem. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1582–1590 (2016)
32. Sattler, T., Zhou, Q., Pollefeys, M., Leal-taixé, L.: Understanding the Limitations of CNN-based Absolute Camera Pose Regression. In: CVPR. pp. 3302–3312 (2019)
33. Schindler, G., Brown, M., Szeliski, R.: City-scale location recognition. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR07). Minneapolis (June 2007)
34. Shavit, Y., Ferens, R., Keller, Y.: Learning multi-scene absolute pose regression with transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2733–2742 (2021)
35. Shavit, Y., Ferens, R., Keller, Y.: Paying attention to activation maps in camera pose regression. In: arxiv preprint, arxiv:2103.11477 (2021)
36. Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., Fitzgibbon, A.: Scene coordinate regression forests for camera relocalization in rgb-d images. In: Proc. Computer Vision and Pattern Recognition (CVPR). IEEE (June 2013), http://research.microsoft.com/apps/pubs/default.aspx?id=184826
37. Sitzmann, V., Martel, J.N., Bergman, A.W., Lindell, D.B., Wetzstein, G.: Implicit neural representations with periodic activation functions. In: Proc. NeurIPS (2020)
38. Sitzmann, V., Zollhöfer, M., Wetzstein, G.: Scene representation networks: Continuous 3d-structure-aware neural scene representations. In: Advances in Neural Information Processing Systems (2019)
39. Sucar, E., Liu, S., Ortiz, J., Davison, A.: iMAP: Implicit mapping and positioning in real-time. In: Proceedings of the International Conference on Computer Vision (ICCV) (2021)
40. Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J., Ng, R.: Fourier features let networks learn high frequency functions in low dimensional domains. Advances in Neural Information Processing Systems 33, 7537–7547 (2020)
41. Torii, A., Arandjelovic, R., Sivic, J., Okutomi, M., Pajdla, T.: 24/7 place recognition by view synthesis. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1808–1817 (2015)
42. Torii, A., Sivic, J., Pajdla, T.: Visual localization by linear combination of image descriptors. In: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops). pp. 102–109. IEEE (2011)
43. Von Stumberg, L., Wenzel, P., Khan, Q., Cremers, D.: Gn-net: The gauss-newton loss for multi-weather relocalization. IEEE Robotics and Automation Letters 5(2), 890–897 (2020)
44. Wang, B., Chen, C., Xiaoxuan Lu, C., Zhao, P., Trigoni, N., Markham, A.: Atloc: Attention guided camera localization. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 10393–10401 (2020)
45. Wenzel, P., Wang, R., Yang, N., Cheng, Q., Khan, Q., von Stumberg, L., Zeller, N., Cremers, D.: 4Seasons: A cross-season dataset for multi-weather SLAM in autonomous driving. In: Proceedings of the German Conference on Pattern Recognition (GCPR) (2020)
46. Xie, Y., Takikawa, T., Saito, S., Litany, O., Yan, S., Khan, N., Tombari, F., Tompkin, J., Sitzmann, V., Sridhar, S.: Neural fields in visual computing and beyond (2021). https://neuralfields.cs.brown.edu/
47. Xue, F., Wu, X., Cai, S., Wang, J.: Learning multi-view camera relocalization with graph neural networks. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11372–11381 (2020)
48. Yen-Chen, L., Florence, P., Barron, J.T., Rodriguez, A., Isola, P., Lin, T.Y.: inerf: Inverting neural radiance fields for pose estimation. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1323–1330. IEEE (2021)
49. Zhou, Y., Wan, G., Hou, S., Yu, L., Wang, G., Rui, X., Song, S.: Da4ad: End-to-end deep attention-based visual localization for autonomous driving. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)
50. Zhou, Y., Barnes, C., Jingwan, L., Jimei, Y., Hao, L.: On the continuity of rotation representations in neural networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
51. Zhu, Y., Gao, R., Huang, S., Zhu, S., Wu, Y.: Learning neural representation of camera pose with matrix representation of pose shift via view synthesis. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9954–9963. IEEE Computer Society, Los Alamitos, CA, USA (jun 2021). https://doi.org/10.1109/CVPR46437.2021.00983, https://doi.ieeecomputersociety.org/10.1109/CVPR46437.2021.00983
Supplementary Materials

This document presents additional localization results of our method ImPosing (single scene and multi scenes on small scale benchmarks), as well as datasets preparation details. We also invite readers to view the supplementary video where localization results are shown on a wide range of scenarios.

A Supplementary video

The attached video file shows test videos of Daoxiang Lake and 4 seasons datasets, with corresponding predicted trajectories from single scenes ImPosing models. The input image is displayed on the top left corner. The right part of the video shows the current predicted trajectory in red, ground truth poses in green and training trajectories in gray. The bottom left corner displays the 256 best candidates selected for pose averaging in red, the predicted pose in black and the groundtruth pose in green. Finally, the last plot shows the score of all candidates in the entire map from transparent ($s = 0$) to red ($s = 1$).

B Small scenes datasets

For completeness, we evaluate ImPosing on Cambridge Landmarks [13] and 7scenes [36], which are widely used benchmarks for visual localization. These datasets evaluate visual localization algorithms in smaller environments where camera is carried by a human instead of equipped on a vehicle. Cambridge Landmarks depicts buildings with rich features observed from different viewpoints in outdoor dynamic conditions. 7scenes evaluates localization in small indoor static environments. In both cases, videos used for testing follow different paths than trajectories observed during training. Methods that use a 3D model usually perform well on these benchmarks thanks to geometric reasoning, whereas regression methods overfit on the small amount of training poses and exhibit a limited accuracy [22]. Results are reported in table 3.

On Cambridge Landmarks, ImPosing is less accurate than state-of-the-art pose regression and image retrieval competitors. However, in the multi scenes scenario, we observe that ImPosing is slightly better than single scene models, whereas accuracy of competitors decreases from single scene to multi scenes. On 7scenes, ImPosing outperforms Image Retrieval but present a higher error compared to pose regression, especially on rotation.

We believe that our candidates sampling method is not well suited for such smaller scenes where the set of camera orientations is more diverse than in large scale driving scenarios where our method compares favorably. These experiments confirm that ImPosing is compatible with multi scene scenarios, which is an important property for large scale deployment of localization systems.
Table 3: Small-scale datasets (median localization error in meters/degrees).

| Dataset     | Image retrieval | Camera Pose Regression | ImPosing |
|-------------|-----------------|------------------------|----------|
|             | NetVLAD [1]     | PoseNet [3]            | Transformer [34] |
|             | Single sc.      | Multi sc.              | Single sc. | Multi sc. |
| K College   | 2.94/6.23       | 0.70/0.9               | 0.99/1.1  | 1.73/3.6 | 0.60/2.4 | 0.83/1.5 | 1.71/3.6 | 1.58/3.5 |
| OldHosp     | 4.87/9.2        | 0.97/2.1               | 2.17/2.9  | 2.55/4.1 | 1.45/3.1 | 1.81/2.4 | 2.45/5.1 | 2.39/5.0 |
| Shop        | 1.32/7.8        | 0.69/3.7               | 1.05/4.0  | 2.02/7.5 | 0.55/3.5 | 0.86/3.1 | 0.97/7.0 | 0.93/6.9 |
| Church      | 3.71/11.1       | 1.32/3.6               | 1.49/3.4  | 2.67/6.2 | 1.09/5.0 | 1.62/4.0 | 1.51/6.6 | 1.60/7.0 |
| Average     | 3.21/8.6        | 0.92/2.6               | 1.43/2.9  | 2.24/5.4 | 0.92/3.5 | 1.28/2.8 | 1.66/5.6 | 1.63/5.6 |
| Scenes      |                 |                        |           |          |          |          |          |          |
| Chess       | 0.24/10.4       | 0.21/11.0              | 0.14/4.5  | 0.09/4.8 | 0.08/5.7 | 0.11/4.7 | 0.17/8.5 | 0.17/9.6 |
| Fire        | 0.33/14.0       | 0.33/15.1              | 0.27/11.8 | 0.29/10.5 | 0.24/10.6 | 0.24/9.6 | 0.28/11.3 | 0.29/12.6 |
| Heads       | 0.18/16.4       | 0.17/16.7              | 0.18/12.1 | 0.16/13.1 | 0.13/12.7 | 0.14/12.2 | 0.13/13.0 | 0.13/13.8 |
| Office      | 0.30/11.1       | 0.29/12.0              | 0.20/5.8  | 0.16/6.8  | 0.17/6.3  | 0.17/5.7  | 0.25/9.0  | 0.25/9.1 |
| Pumpkin     | 0.38/11.2       | 0.37/11.0              | 0.25/4.8  | 0.19/5.5  | 0.17/5.6  | 0.18/4.4  | 0.27/10.4 | 0.29/9.6 |
| Kitchen     | 0.34/12.3       | 0.36/12.2              | 0.24/5.5  | 0.21/6.6  | 0.19/6.8  | 0.17/5.9  | 0.28/8.0  | 0.26/8.2 |
| Stairs      | 0.28/13.8       | 0.31/14.8              | 0.37/10.6 | 0.31/11.6 | 0.30/7.0  | 0.26/8.4  | 0.27/11.8 | 0.26/11.3 |
| Average     | 0.29/12.7       | 0.29/13.3              | 0.24/7.9  | 0.20/8.4  | 0.18/7.8  | 0.18/7.3  | 0.24/10.3 | 0.24/10.6 |

C Datasets preparation

This section contains dataset splits used in our experiments to ensure reproducibility. To the best of our knowledge, 4Seasons [45] and Daoxiang Lake datasets [49] had not been used previously to evaluate direct learning-based methods.

C.1 Oxford RobotCar dataset

The dataset can be downloaded [here]. We replicate experiments from previous methods using undistorted front camera images:

|          | Oxford Loop       | Oxford Full        |
|----------|-------------------|--------------------|
| Training set | 2014-06-26-09-24-58 | 2014-11-28-12-07-13 |
|          | 2014-06-23-15-41-25 | 2014-12-02-15-30-08 |
| Test set  | 2014-06-26-08-53-56 | 2014-12-09-13-21-02 |
|          | 2014-06-23-15-36-04 | 2014-12-09-13-21-02 |

C.2 Daoxiang Lake dataset

The dataset can be downloaded [here]. Vehicles are equipped with multiple sensors but we only use the front camera images with associated vehicle poses. We don’t use the train/test split provided by the dataset because the test set is not an entire held out sequence (images from the same sequence has been observed during training) and then is not a realistic test scenario.
C.3 4seasons dataset

The dataset can be downloaded here. Absolute poses are generated using available Python tools. We use keyframes from the left camera only.

|        | Neighborhood | Office Loop | Countryside | Bus. campus | City Loop | Old Town |
|--------|--------------|-------------|-------------|-------------|-----------|----------|
| Train  | 2020-03-26,13:32-55 | 2020-03-24,17:36-22 | 2020-04-07,11:33-45 | 2020-10-08,19:30-57 | 2020-12-22,11:33-15 | 2020-10-08,11:33-41 |
|        | 2020-10-07,14:47-515 | 2020-03-24,17:45-31 | 2020-06-12,11:26-43 | 2021-01-07,13:12-23 | 2021-01-07,14:36-17 | 2021-01-07,10:49-45 |
|        | 2020-10-07,14:53-52 | 2020-04-07,10:20-32 | 2021-01-07,13:30-07 | 2020-10-08,11:54-24 | 2020-06-12,10:10-57 | 2021-05-10,21:32-00 |
|        | 2020-12-22,11:54-24 | 2020-06-12,10:30-22 | 2021-01-07,12:04-03 | 2021-01-07,12:04-03 | 2021-01-07,12:04-03 | 2021-01-07,12:04-03 |
| Test   | 2021-05-10,18:32-32 | 2021-02-25,13:51-57 | 2020-10-08,09:57-28 | 2021-02-25,14:16-15 | 2021-02-25,11:09-49 | 2021-02-25,12:34-08 |