RADA: Robust Adversarial Data Augmentation for Camera Localization in Challenging Conditions

RADA: Robust Adversarial Data Augmentation for Camera Localization in Challenging Conditions

Jialu Wang[1], Muhamad Risqi U. Saputra[2], Chris Xiaoxuan Lu[3], Niki Trigoni[1] and Andrew Markham[1]

Abstract— Camera localization is a fundamental problem for many applications in computer vision, robotics, and autonomy. Despite recent deep learning-based approaches, the lack of robustness in challenging conditions persists due to changes in appearance caused by texture-less planes, repeating structures, reflective surfaces, motion blur, and illumination changes. Data augmentation is an attractive solution, but standard image perturbation methods fail to improve localization robustness. To address this, we propose RADA, which concentrates on perturbing the most vulnerable pixels to generate relatively less image perturbations that perplex the network. Our method outperforms previous augmentation techniques, achieving up to twice the accuracy of state-of-the-art models even under 'unseen' challenging weather conditions. Videos of our results can be found at https://youtu.be/niOv7-fJeCA. The source code for RADA is publicly available at https://github.com/jialuwang123321/RADA.

I. Introduction

Camera localization refers to the problem of recovering the 6-DoF camera poses from input images. It is a fundamental problem in robotics for applications such as augmented reality and autonomous driving. Traditional hand-crafted features are vulnerable to changes in illumination, scene dynamics, or texture-less regions [1]. Deep-learning-based models have been proposed to extract informative features automatically [1], [2], but maintaining robustness in the face of challenging conditions remains a struggle, especially if these conditions have not been observed (or only sporadically) in the training set [3]. Such conditions include changes in lighting, dynamic objects, texture-less surfaces, repeating patterns, reflective objects, or motion blurs [4], [5], [6].

One possible solution to tackle this problem is to use well-known basic data augmentation techniques, such as shifting RGB pixel values or using Gaussian noise, have been employed [2], [1], [7], [8], [9]. However, these "data agnostic" techniques may corrupt important information by perturbing all image pixels uniformly. Adversarial training (AT) [10], a deep-learning-based data augmentation method, has been shown to improve robustness in classification problems [10], [11], [12], but has not yet been used to improve deep learning-based camera localization.

[1]Jialu Wang, Niki Trigoni and Andrew Markham are with Department of Computer Science, University of Oxford, UK, jialu.wang@cs.ox.ac.uk, niki.trigoni@cs.ox.ac.uk, andrew.markham@cs.ox.ac.uk

[2]Muhamad Risqi U. Saputra is with the Data Science Department, Monash University, Indonesia, risqi.saputra@monash.edu

[3]Chris Xiaoxuan Lu is with the School of Informatics, the University of Edinburgh, UK, xiaoxuan.lu@ed.ac.uk

In this paper, we propose a novel adversarial data augmentation approach for robust camera localization, dubbed as RADA. RADA learns to generate relatively less image perturbations that are still capable of perplexing the network by concentrating more on perturbing the most vulnerable pixels. As a result, RADA can improve the robustness of the localization models in challenging and even 'unseen' cross-domain situations. This is the product of the generalization benefits of RADA which allows us to work without prior-information of the target domain [13]. We show that if we trained a localization model using RADA with outdoor dataset taken only from good weather condition, the localization model can have satisfactory robustness in a variety of challenging and unseen test scenarios (e.g., snow, over-exposure sunny day etc.).

It is worth noting that Generative Adversarial Network (GAN) [14] and Adversarial Training (AT) [10] are different concepts [15], although they share terminology. AT improves model robustness using adversarial attacks, while GAN generates synthetic images. They can be combined for complementary advantages [10]. While GAN has been used for camera localization, it requires some target domain information [16], [17], [18], whereas AT can generalize without prior information of the target domain [13]. Furthermore, some other works also use AT technique, but they focus on different tasks [19], [20], [21], [3], [8].

In summary, our key contributions are given as follows:

- We propose RADA, a Robust Adversarial Data Augmentation system, which is to our knowledge the first adversarial training (AT) approach applied to the problem of camera localization. It creates relatively less alteration to the original image by selectively perturbing only the most vulnerable pixels.
- We demonstrate that the proposed RADA approach can significantly improve the robustness of the localization models in challenging and even 'unseen' cross-domain conditions.
- We evaluate RADA and competing approaches on diverse datasets (e.g., RGB-D, RGB, and 3D point clouds) and various localization models, demonstrating its efficacy for diverse localisation problems, namely both 2D (e.g., RGB) and 3D data (e.g., point clouds).

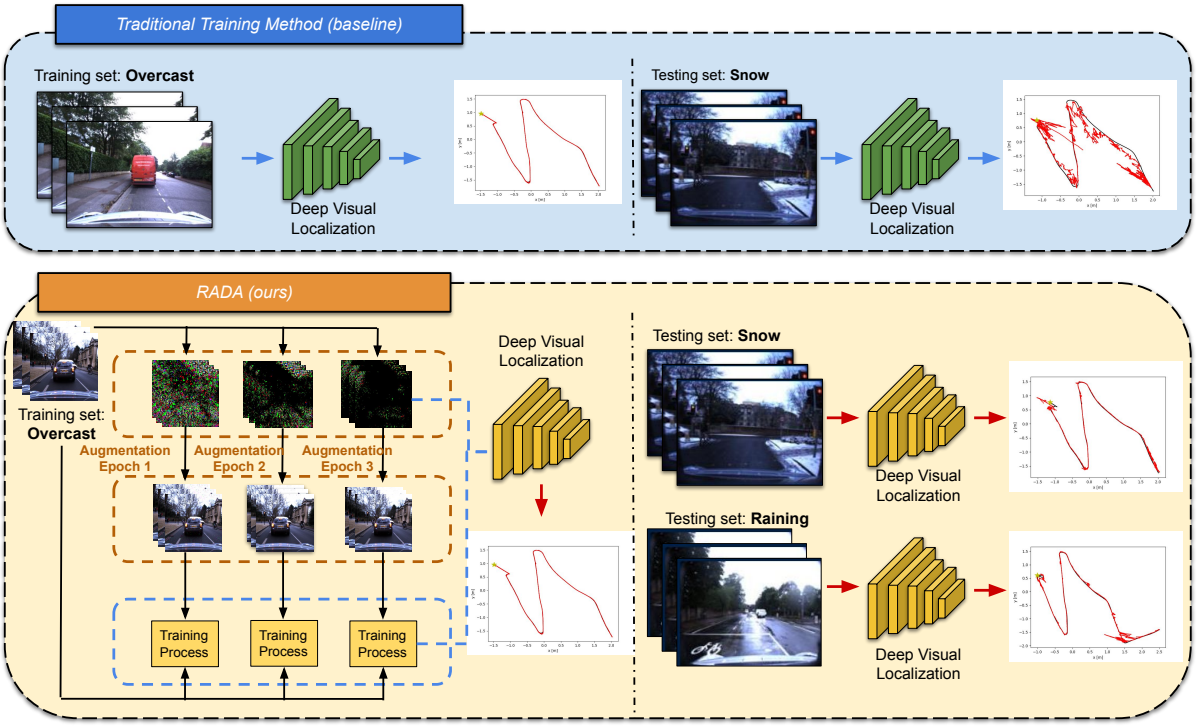


Fig. 1. Upper: Traditional training method for deep learning-based camera localization. Lower: Our proposed training method with RADA.

II. Related Work

A. Camera Localization

Camera localization, also known as camera pose estimation, aims at recovering camera 6-DoF pose from given query images under a world coordinate system. This problem was initially solved as a place recognition problem [22], [23] which estimate the pose of the query image by retrieving the most similar images from the database. With the rapid progress of deep learning in the field of computer vision, the solutions to camera localization problems have developed from the traditional structure based camera localization methods (recovering the camera pose from 3D scene through the extracted local feature descriptors [24], [25], or using a random forest [26], [27]) to Absolute Pose Regression (APR) (directly predicting the 6-DoF pose from input images by optimizing an CNN [1], [28], [2]).

B. Data Augmentation Approaches

Data augmentation improves model robustness by artificially generating data without changing labels or requiring target domain data [29], [30], [15]. Approaches can be grouped into two categories: 1) Basic image manipulation, which employs random color and geometric transformations or Gaussian noise, and offers limited improvements in robustness. 2) Deep-learning approaches, such as GAN [14] and AT [10], which utilize generator networks and can be combined for complementary advantages.

C. Adversarial Training (AT)

Adversarial training is a deep-learning-based data augmentation technique proposed by [13]. FGSM [31] improved training speed by using a small step size, while FGM [32] used a unit vector to avoid perturbing different pixels by the same step size. PGD [10] added a constraint through multiple iterations, while EAT [33] increased sample diversity and ALP [11] used a "pairing loss." JSMA [34] proposed saliency maps to select features with the most significant impact on the output.

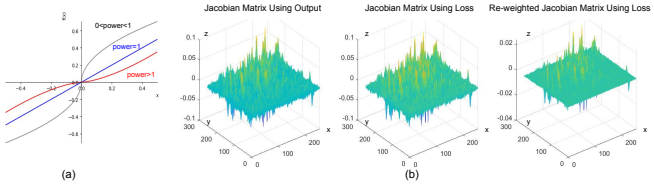


Fig. 2. (a) Visualization of $f(x) = (J_x)^{pow}$ with different power range. The x-axis refers to J_x^L, while the y-axis is their scaled values. When $power > 1$ (red), it re-assign higher weights to greater J_x^L, and vice versa. We do not use either $0 < power < 1$ (gray) that gives an unwanted opposite effect, or the linear function $power = 1$ (blue) that cannot even widen their disparity. (b) Visualized distribution of different Jacobian matrix. The x-y-plane is the input pixels while the z-axis is the corresponding Jacobian matrix. Since the loss $L(p, p^*)$ does not change the monotonicity of the output p, three Jacobian matrices J_x^p (left), J_x^L (middle) and $J_x^{L^{pow}}$ (right) have similar distributions. We can therefore replace J_x^L with the J_x^p (obtained in step 1) to save the computational cost.

III. Methods

A. Background

In this section, we take a deeper dive into the two techniques that our proposed approach is closer to (and inspired by), namely FGSM and JSMA, highlighting their strengths and limitations.

FGSM [31] is one of the fastest AT methods that uses gradient-based perturbations. Similar to other gradient ascent methods, the perturbation r_{adv} is calculated by Eq. (1), where x is the input image, constant ε is the step size, $\bigtriangledown_x L$ is the gradient of loss function w.r.t the image, $(\cdot)^{pow}$ is the element-wise power. Existing methods either set *pow* to 0 (e.g., FGSM) or 1 (e.g., FGM, PGD, etc.).

$$r_{adv} = \varepsilon \cdot sign(\bigtriangledown_x L) \cdot (\bigtriangledown_x L)^{pow} \tag{1}$$

Then they perform $x_{adv} = x + r_{adv}$ to perturb the original image along the gradient sign direction. However, Using it directly for camera localization would perturb all pixels equally, even irrelevant ones, creating confounding dots (see Fig. 4).

JSMA [34] advances that, vulnerable pixels are those that have a higher impact on the output of a network in a computer vision task. In other words, they are the pixels that, when perturbed or altered, are more likely to cause mis-classification or errors in the network's output. By focusing on perturbing most vulnerable part of inputs, even a constant perturbation (e.g., $r_{adv} = 1$) can lead to successful attacks. Assume F is the network output and x is the input image in a computer vision task, the vulnerable pixels can be identified through the Jacobian matrix $J_x^F = \frac{\partial F(x)}{\partial x}$, where greater magnitude elements indicate which pixels are more vulnerable. However, this method incurs significant computational overhead from repetitive forward derivative estimation.

B. Proposed Adversarial Training for Camera Localization

As discussed before, JSMA perturbs only the top-k vulnerable pixels but is slow due to the Jacobian matrix computation, while FGSM is fast but perturbs all pixels. RADA combines the strengths of both methods and mitigates their limitations, proposing a novel AT technique in four steps.

Step 1: Calculate gradient-based perturbations. A localization model is trained to predict camera poses by back-propagating the loss function $L_\theta^r(p, p^*)$ with respect to the input image x. The perturbation is obtained by computing the gradient $\bigtriangledown_x L$ and substituting it into Eq. (1).

Step 2: Find the most vulnerable pixels. In a similar way to JSMA, we use the Jacobian Matrix to find out vulnerable pixels of the input image. Notice that, in the camera localization problem, large derivative of the loss function $L_\theta^r(p, p^*)$ is typically linked with large elements in the Jacobian matrix of the model's output $F(x) = p$ (see Fig. 2(a)). Therefore, when calculating the Jacobian matrix, we replace $J_x^{F(x)} = \bigtriangledown_x p$ with $J_x^{L_\theta^r(p,p^*)} = \bigtriangledown_x L_\theta^r(p, p^*)$ which was obtained in step 1 (see Fig. 2(b)). This reduces the computational cost (We tested naively combining JSMA and FGSM attacks on DSAC$\star$, but the model didn't converge after 7 days. In contrast, RADA converged in 14 hours, compared to the typical 7 hours for vanilla DSAC$\star$).

Step 3: Re-weigh and apply perturbations. We use a soft re-weighting function to amplify the perturbation on the most vulnerable pixels and preserve more localization information compared to JSMA's piece-wise approach. The function, $f(J_x) = (J_x)^{pow}$ assigns higher weights to more vulnerable pixels and $pow > 1$ amplifies the perturbation super linearly. Increasing *pow* amplifies the effect of perturbing the most vulnerable pixels.

Next, we scale $(J_x)^{pow}$ to the pixel value range by multiplying it with a scaling factor. $M_{k\%}$ denotes the set of pixels x_i, whose $\left|J_{x_i}^{pow}\right|$ are ranked in the top $k\%$ among those of other pixels. Its corresponding scaling factor $\varepsilon_{k\%}$ is calculated as in Eq. (2)).

$$\varepsilon_{k\%} = \left(\sum \frac{|J_{x_i}^{pow}|}{|x_i|}\right)_{avg} \qquad \text{if } x_i \in M_{k\%}, \tag{2}$$

We use two scaling factors, $\varepsilon_{0.1\%}$ and $\varepsilon_{50\%}$, due to the large slope variation in $f(J_x)$. $\varepsilon_{0.1\%}$ is applied to the most vulnerable pixels in the top 0.1 percentile of $\left|J_{x_i}^{pow}\right|$, while $\varepsilon_{50\%}$ is applied to the remaining pixels in the image based on the top 50th percentile.

After evaluating scaling factors for the two classes of pixels, we then apply the perturbation r_{adv} as shown in Eq. (3).

$$r_{adv} = \begin{cases} \varepsilon_{0.1\%} \cdot sign(\bigtriangledown_{x_i} L_\theta^r(p, p^*)) \cdot (\bigtriangledown_{x_i} L_\theta^r(p, p^*))^{pow} \\ \qquad\qquad \text{if } x_i \in M_{0.1\%}, \\ \varepsilon_{50\%} \cdot sign(\bigtriangledown_{x_i} L_\theta^r(p, p^*)) \cdot (\bigtriangledown_{x_i} L_\theta^r(p, p^*))^{pow} \\ \qquad\qquad \text{Otherwise.} \end{cases} \tag{3}$$

Step 4: Threshold and Clipping. Then we used two mechanisms to add constraints on the magnitude of obtained perturbation (r_{adv}) and the final x_{adv} respectively.

(1) Threshold. Before each batch of training, we calculate a perturbation threshold (η_{th}) from the pixel range so as to limit the size of the perturbation. This design enables the perturbation to be adaptive to the input dataset. As shown in Eq. (4), x_{max} and x_{min} are the maximum and minimum pixel values of each batch. The number of thresholds η is calculated accordingly to limit the percentage of threshold-ed out pixels to each image to a preset upper bound.

$$\eta_{th} = (x_{max} - x_{min}) \div \eta \tag{4}$$

(2) Clipping. To avoid invalid values, we further clip the value of the perturbed pixels into the range of [0,255]. Our ablation study in IV-G shows that this restriction can effectively reduce the confounding pixels.

The overall workflow of generating RADA adversarial samples is shown in Algorithm 1. The RADA system pipeline is illustrated in Fig. 1.

Algorithm 1: RADA Algorithms

for each epoch do
1. Use the range of x to calculate the perturbation threshold η_{th} (Eq. (4)) and the adversarial step size $\varepsilon_{k\%}$ (Eq. (2));
2. Freeze the model parameter θ and calculate the perturbation r_{adv} (Eq. (3));
3. If any r_{adv} exceeds the η_{th}, assign η_{th} to that perturbation;
4. Calculate adversarial sample x_{adv} using $x_{adv} = x + r_{adv}$;
5. If any pixel value in x_{adv} exceeds the range $[0, 255]$, assign the nearest boundary (0 or 255) to that pixel;
6. Use x_{adv} to train the model and update the weights θ;

end

IV. EXPERIMENTS

A. Localization Models and Evaluation Metrics

We enhance DSAC⋆ [35], MapNet [2], and AtLoc [1], which are state-of-the-art camera localization models, using our proposed RADA approach. DSAC⋆ predicts 3D point clouds and derives the pose afterward, while MapNet and AtLoc belong to the APR branch, which directly regress the pose of each image. MapNet is a deep learning-based camera localization model that recovers the absolute camera pose from input images while achieving a relatively higher robustness to illumination changes. Atloc is a camera localization model that uses a self-attention mechanism to handle dynamic objects and changing illumination. DSAC⋆ estimates scene coordinates from RGB images, RGB-D images or 3D point clouds using a CNN and then determines the optimal camera pose by evaluating multiple sampled pose hypotheses through a RANSAC optimization process. In this paper, we used the same evaluation strategies and datasets as the original papers for all models, whether augmented or vanilla versions, to ensure a fair comparison. More specifically, mean translation error (m) and mean rotation error (degree) for MapNet and AtLoc, and median translation error (cm) and median rotation error (degree) for DSAC⋆.

B. Datasets

TABLE I
RobotCar [36] Dataset Selection

Sequence	Time	Tag	Model
loop	2014-06-26-09-24-58	overcast	Training
loop	2014-06-26-08-53-56	overcast	Training
loop	2014-06-23-15-36-04	overcast	Testing
loop	2014-06-24-14-09-07	over-exposure	Testing
fullA	2014-11-28-12-07-13	overcast	Training
fullA	2014-12-02-15-30-08	overcast	Training
fullA	2014-11-25-09-18-32	rain	Testing
fullB	2015-02-13-09-16-26	overcast	Training
fullB	2015-02-03-08-45-10	snow	Testing

Oxford RobotCar Dataset [36] is a large-scale dataset collected from a 10km autonomous driving route in central Oxford. For this study, we used input images from the stereo centre camera sequence 01 with a resolution of 1280 x 960, and ground truth poses obtained from INS data interpolations. 7Scenes Dataset [37] is an indoor localization dataset with RGB and RGB-D image sequences from seven scenarios, with ground truth camera poses obtained from KinectFusion. Cambridge Landmarks Dataset [28] is an outdoor localization dataset with RGB images, 3D point clouds, and ground truth camera poses reconstructed using structure-from-motion. All datasets contain challenging conditions, making them suitable for testing the robustness of localization models.

C. Implementation Details

For RADA adversarial perturbations, we set $pow = 1.5$ and calculate the threshold η to limit the percentage of thresholded out pixels to 20%. FGSM perturbations use a typical value of $\varepsilon = 0.3$. The Gaussian perturbation uses a mean of 0 and variance of 0.05, both commonly used values. To ensure a fair comparison, we maintained the same hyper-parameters as the original papers for all models, whether augmented or vanilla versions. All models were trained until convergence and in IV-E, challenging test weather conditions were either hidden or excluded from the training data. We used all scenes of 7scenes and Cambridge Landmark datasets, and the details about selected sequences of RobotCar Dataset are provided in Table I. All models were trained to convergence using a single Nvidia RTX-3090 GPU.

D. Performance on Indoor and Outdoor Environment with Challenging Situations

We firstly trained DSAC⋆ on 7Scenes [37] and Cambridge [28] datasets. Table II shows the numerical evaluation results. All competing systems work properly when they were tested on the unchallenging data, but suffer from translation accuracy degradation on the challenging conditions (Stairs, Redkitchen, Pumpkin (7Scenes [37], indoor), OldHospital, StMarysChurch, GreatCourt, KingsCollege (Cambridge [28], outdoor)). RADA modified DSAC⋆ outperforms the other techniques on the

TABLE III
Comparing RADA with SOTA methods using MapNet and AtLoc

Testing Sequence	AtLoc [1]	AtLoc with Gaussian	AtLoc with FGSM	AtLoc with RADA (ours)
overcast	8.86m, 4.67°	10.93m, 5.97°	8.11m, 3.60°	6.93m, **3.40°**
over-exposure	22.17m, 17.72°	8.80m, 9.38°	9.54m, 8.76°	9.23m, 8.84°
rain	8.99m, 2.15°	12.40m, 2.96°	12.90m, 2.74°	6.87m, 1.90°
snow	37.99m, 8.18°	29.21m, 10.82°	24.56m, 11.12°	13.07m, 8.15°
average	19.50m, 8.18°	15.34m, 7.28°	13.78m, 6.55°	9.02m, 5.57°
Testing Sequence	MapNet [2]	MapNet with Gaussian	MapNet with FGSM	MapNet with RADA (ours)
overcast	9.84m, 3.96°	69.48m, 50.22°	16.89m, 11.65°	17.26m, 6.82°
over-exposure	18.49m, 12.88°	49.80m, 39.33°	19.74m, 8.27°	16.89m, 11.65°
rain	15.00m, 3.72°	16.11m, 3.60°	13.39m, 3.60°	12.10m, 2.56°
snow	29.16m, 9.06°	36.22m, 12.04°	23.18m, 8.63°	12.91m, 2.65°
average	18.12m, 7.41°	42.90m, 26.30°	18.3m, 8.04°	14.79m, 5.92°

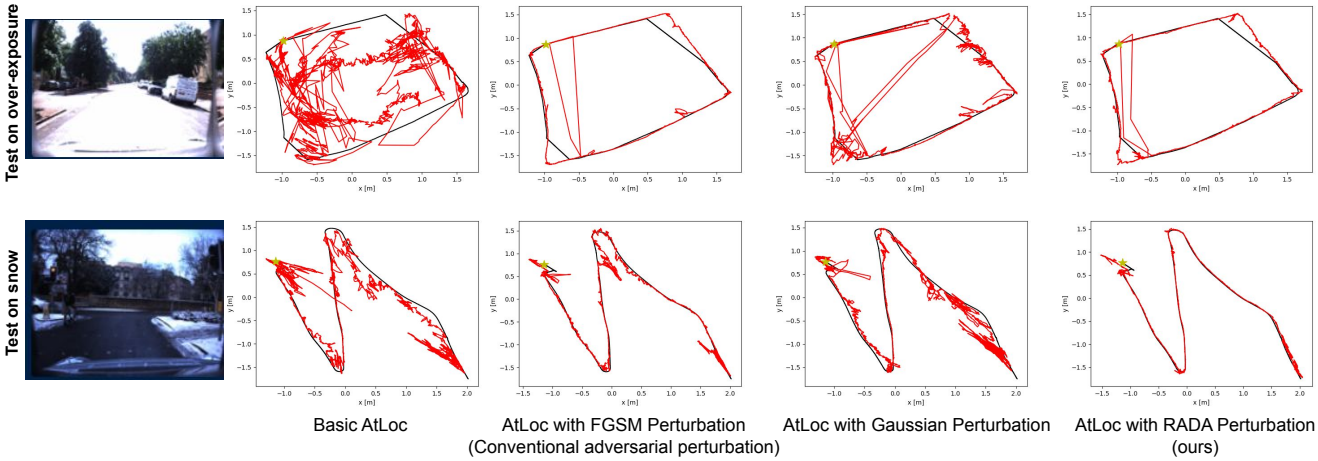


Fig. 3. Trajectories on over-exposure day (upper) and snowy day (lower) test set. The ground truth (black line), predicted trajectories (red line) and starting point (star) are visualized above.

majority of good and challenging weather conditions. The visualized trajectory for DSAC⋆ can be found in the supplementary material. On average, RADA yields 40.55% and 18.67%, 13.7% and 9.1% smaller translation and rotation error respectively for 7Scenes [37] and Cambridge [28] datasets.

E. Performance on Cross-Domain Situations

We then trained both AtLoc and MapNet on Oxford RobotCar Dataset [36]. We compared RADA with different traditional data augmentation methods, and then tested them in good weather as well as 'unseen' challenging weathers. For the non-augmented system, we trained basic AtLoc and MapNet [2] by utilizing the original, unmodified dataset. For naive augmentation system, we add Gaussian noise to the training dataset. For conventional deep learning-based augmentation system, we employ FGSM [31] to perturb the dataset. Apart from the basic MapNet and AtLoc, all models were trained by using both the original and the perturbed data. Fig. 3 visualizes some of the predicted trajectory for AtLoc.

Table III shows the numerical evaluation results. The unmodified AtLoc and MapNet models performed well during testing under good weather conditions (overcast). However, the MapNet model exhibited a degradation in translation-only accuracy during testing under rainy conditions. Additionally, both models experienced significant degradation in performance when tested under snowy and over-exposed conditions. Sometimes, Gaussian and FGSM can improve the network's robustness, but most of the time they confuse the network since the generated perturbation is either random or too noisy such that it corrupts critical pixel information used for camera localization. In contrast, AtLoc and MapNet modified with RADA outperform the other techniques on most of good and challenging weather conditions. On average, RADA yields 36.7% and 26.5% smaller translation and rotation error respectively for both AtLoc and MapNet.

TABLE II

Performance on Indoor/Outdoor Challenging Situations of DSAC⋆ with RADA

Dataset	Sequence	Method	Chess	Fire	Heads	Office	Pumpkin	RedKitchen	Stairs
7scenes	RGB	w.o. RADA	1.9cm, 1.11°	1.9cm, 1.24°	1.1cm, 1.82°	2.6cm, 1.18°	4.2cm, 1.41°	3.0cm, 1.7°	4.1cm, 1.42°
		w. RADA	1.64cm, 0.56°	1.73cm, 0.77°	1.0cm, 0.73°	2.5cm, 0.74°	3.63cm, 0.93°	2.7cm, 1.3°	4.0cm, 1.2°
	RGB-D	w.o. RADA	1.0cm, 1.03°	1.1cm, 1.05°	1.0cm, 1.88°	1.2cm, 1.0°	2cm, 1.17°	2.1cm, 1.41°	2.6cm, 1.15°
		w. RADA	0.98cm, 0.42°	1.0cm, 0.58°	0.9cm, 0.8°	1.2cm, 0.48°	1.8cm, 0.6°	2cm, 0.84°	2.5cm, 0.7°

Dataset	Sequence	Method	StmarysChurch	Great Court	Old Hospital	King's College	ShopFacade
Cambridge	3D point clouds	w.o. RADA	13.4cm, 0.45°	48.5cm, 0.25°	21cm, 0.41°	14.7cm, 0.29°	4.6cm, 0.25°
		w. RADA	11.4cm, 0.40°	39.0cm, 0.20°	19.5cm, 0.37°	13.3cm, 0.27°	5.0cm, 0.30°

F. Perturbation Histogram

We compared different perturbation methods using histograms. As shown in Fig. 4, Gaussian and FGSM perturbations are evenly distributed on the whole image, generating a lot of noise. In contrast, RADA produces targeted perturbations on small, informative regions such as trees and buildings. This improves the network's robustness to challenging and even 'unseen' cross-domain variations.

G. Ablation Study

We also conducted ablation study under different weather conditions using AtLoc model. In Table IV, basic AtLoc is compared with a version trained with complete RADA, a clipping version of RADA trained without threshold, a threshold version trained without clipping, and a no-threshold-no-clipping version of RADA. All models were trained in good condition (overcast) and then tested in different 'unseen' challenging conditions (over-exposure and rainy). The rest settings are kept the same for fair comparison. The complete RADA achieved the best performance among all. This comparison indicates that all incomplete RADA versions are less accurate than the complete RADA, showing that the threshold and clipping mechanism can produce effective constraints (see Fig. 5).

V. CONCLUSIONS

We propose RADA, an adversarial data augmentation system for camera localization, which aims to performs relatively less alteration to the original image by concentrating more strongly on perturbing the most vulnerable pixels. By using RADA, we demonstrated the possibility of using AT to improve the robustness of camera localization models in challenging and even 'unseen' cross-domain conditions. It is a simple plugin that can be used on virtually any deep-learning based camera localization network to generate adversarial training examples. As part of our future work, we plan to improve RADA's robustness for challenging testing conditions, such as nighttime scenarios. This will advance the field of robotics and autonomous systems and make RADA a more effective tool for real-world applications.

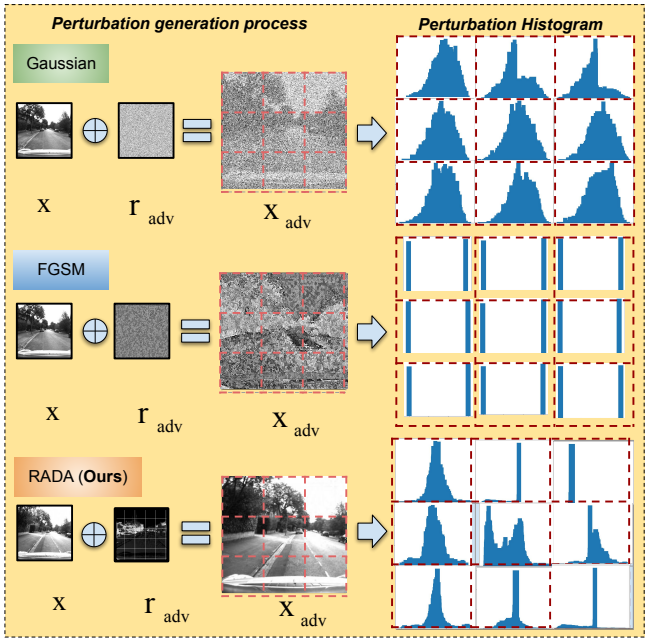


Fig. 4. The comparison of perturbation results between Gaussian, FGSM [32], and RADA (ours). x, Δx, and x' are the original image, the generated perturbations, and the perturbed image respectively. We obtained the histogram of perturbations by equally dividing $\Delta x'$ into 9 sub-squares and computing the frequency, omitting pixels without interference, and clipping Gaussian perturbed outputs within [0,1]. Both Gaussian and FGSM perturbations are uniformly distributed in all regions. In contrast, RADA perturbations are concentrated in regions with important geometrical structures for localization, such as trees and buildings. Training the localization model on x' mitigates over-fitting to weather-specific structures and improves the network's robustness to challenging and 'unseen' cross-domain variations.

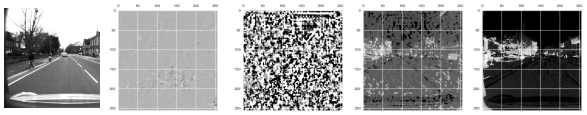
TABLE IV

Ablation study of RADA on RobotCar [36] (Basic AtLoc denotes the original AtLoc Model without RADA)

	Over-exposure	Rainy
Basic AtLoc	22.17, 17.72°	8.99, 2.15°
Complete RADA (ours)	9.23m, 8.84°	6.87m, 1.90°
No clipping RADA (ours)	12.39, 10.98°	10.96, 2.69°
No threshold RADA (ours)	11.99, 10.79°	8.82, 2.05°
no threshold, no clipping RADA (ours)	12.66, 12.75°	17.28, 2.95°

References

[1] B. Wang, C. Chen, C. X. Lu, P. Zhao, N. Trigoni, and A. Markham, "Atloc: Attention guided camera localization," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 06, 2020, pp. 10 393–10 401.



a) Original Image | b) Image with RADA Perturbation, no threshold, no Clipping (ours) | c) Image with RADA Perturbation, no Clipping (ours) | d) Image with RADA Perturbation, no threshold (ours) | e) Complete RADA Perturbation (ours)

Fig. 5. The perturbed images generated by RADA after turning off some building blocks. (a) Original image, (b) RADA without clipping and Threshold mechanisms (ours), (c) RADA without clipping mechanism (ours), (d) RADA without threshold mechanism (ours), and (e)Complete RADA attack (ours). It can be clearly seen that our proposed threshold and clipping mechanism can effectively avoid polluting crucial localization information or generating lots of confounding dots.

[2] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz, "Geometry-aware learning of maps for camera localization," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2616–2625.

[3] M. Shu, Y. Shen, M. C. Lin, and T. Goldstein, "Adversarial differentiable data augmentation for autonomous systems."

[4] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother, "Dsac-differentiable ransac for camera localization," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6684–6692.

[5] S. Schubert, P. Neubert, and P. Protzel, "Unsupervised learning methods for visual place recognition in discretely and continuously changing environments," in 2020 IEEE international conference on robotics and automation (ICRA). IEEE, 2020, pp. 4372–4378.

[6] S. Chen, X. Li, Z. Wang, and V. A. Prisacariu, "Dfnet: Enhance absolute pose regression with direct feature matching," in Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part X. Springer, 2022, pp. 1–17.

[7] J. R. Siddiqui, M. Havaei, S. Khatibi, and C. A. Lindley, "A novel plane extraction approach using supervised learning," Machine vision and applications, vol. 24, pp. 1229–1237, 2013.

[8] I. Sobh, A. Hamed, V. R. Kumar, and S. Yogamani, "Adversarial attacks on multi-task visual perception for autonomous driving," arXiv preprint arXiv:2107.07449, 2021.

[9] K. Zhou, C. Chen, B. Wang, M. R. U. Saputra, N. Trigoni, and A. Markham, "Vmloc: Variational fusion for learning-based multimodal camera localization," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 7, 2021, pp. 6165–6173.

[10] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," arXiv preprint arXiv:1706.06083, 2017.

[11] H. Kannan, A. Kurakin, and I. Goodfellow, "Adversarial logit pairing," arXiv preprint arXiv:1803.06373, 2018.

[12] D. Croce, G. Castellucci, and R. Basili, "Adversarial training for few-shot text classification," Intelligenza Artificiale, vol. 14, no. 2, pp. 201–214, 2020.

[13] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," arXiv preprint arXiv:1312.6199, 2013.

[14] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," arXiv preprint arXiv:1406.2661, 2014.

[15] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," Journal of Big Data, vol. 6, no. 1, pp. 1–48, 2019.

[16] A. Anoosheh, T. Sattler, R. Timofte, M. Pollefeys, and L. Van Gool, "Night-to-day image translation for retrieval-based localization," in 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019, pp. 5958–5964.

[17] H. Porav, W. Maddern, and P. Newman, "Adversarial training for adverse conditions: Robust metric localisation using appearance transfer," in 2018 IEEE international conference on robotics and automation (ICRA). IEEE, 2018, pp. 1011–1018.

[18] B. Chidlovskii and A. Sadek, "Adversarial transfer of pose estimation regression," in Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16. Springer, 2020, pp. 646–661.

[19] T. Ng, H. J. Kim, V. T. Lee, D. DeTone, T.-Y. Yang, T. Shen, E. Ilg, V. Balntas, K. Mikolajczyk, and C. Sweeney, "Ninjadesc: content-concealing visual descriptors via adversarial learning," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 12 797–12 807.

[20] Y. Liu, Y. Wang, and S. Wang, "Adversarial view-consistent learning for monocular depth estimation," arXiv preprint arXiv:1908.01301, 2019.

[21] C. Xiao, R. Deng, B. Li, T. Lee, B. Edwards, J. Yi, D. Song, M. Liu, and I. Molloy, "Advit: Adversarial frames identifier based on temporal consistency in videos," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 3968–3977.

[22] W. Zhang and J. Kosecka, "Image based localization in urban environments," in Third international symposium on 3D data processing, visualization, and transmission (3DPVT'06). IEEE, 2006, pp. 33–40.

[23] G. Tolias and H. Jégou, "Visual query expansion with or without geometry: refining local descriptors by feature aggregation," Pattern recognition, vol. 47, no. 10, pp. 3466–3476, 2014.

[24] Y. Li, N. Snavely, and D. P. Huttenlocher, "Location Recognition using prioritized feature matching," in Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part II 11. Springer, 2010, pp. 791–804.

[25] H. Germain, G. Bourmaud, and V. Lepetit, "Sparse-to-dense hypercolumn matching for long-term visual localization," in 2019 International Conference on 3D Vision (3DV). IEEE, 2019, pp. 513–523.

[26] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9. Springer, 2006, pp. 430–443.

[27] E. Brachmann and C. Rother, "Learning less is more-6d camera localization via 3d surface regression," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4654–4662.

[28] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in Proceedings of the IEEE international conference on computer vision, 2015, pp. 2938–2946.

[29] S. Cygert and A. Czyżewski, "Toward robust pedestrian detection with data augmentation," IEEE Access, vol. 8, pp. 136 674–136 683, 2020.

[30] K. Zhou, L. Hong, S. Hu, F. Zhou, B. Ru, J. Feng, and Z. Li, "Dha: End-to-end joint optimization of data augmentation policy, hyper-parameter and architecture," arXiv preprint arXiv:2109.05765, 2021.

[31] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," arXiv preprint arXiv:1412.6572, 2014.

[32] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," arXiv preprint arXiv:1611.01236, 2016.

[33] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," arXiv preprint arXiv:1705.07204, 2017.

[34] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in 2016 IEEE European symposium on security and privacy (EuroS&P). IEEE, 2016, pp. 372–387.

[35] E. Brachmann and C. Rother, "Visual camera re-localization

from rgb and rgb-d images using dsac," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021.
[36] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford robotcar dataset," The International Journal of Robotics Research, vol. 36, no. 1, pp. 3–15, 2017.
[37] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon, "Scene coordinate regression forests for camera relocalization in rgb-d images," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2013, pp. 2930–2937.