Improving Robot Localisation by Ignoring Visual Distraction

Oscar Mendez Matthew Vowels Richard Bowden

Abstract—Attention is an important component of modern deep learning. However, less emphasis has been put on its inverse: ignoring distraction. Our daily lives require us to explicitly avoid giving attention to salient features that confound the task we are trying to accomplish. This visual prioritisation allows us to concentrate on important tasks while ignoring visual distractors.

In this work, we introduce Neural Blindness, which gives an agent the ability to completely ignore objects or classes that are deemed distractors. More explicitly, we aim to render a neural network completely incapable of representing specific chosen classes in its latent space. In a very real sense, this makes the network “blind” to certain classes, allowing and agent to focus on what is important for a given task, and demonstrates how this can be used to improve localisation.

I. INTRODUCTION

The presence of extraneous, confusing or overwhelming information in an image has been shown to reduce task-specific performance in humans [22]. As such, it should not come as a surprise that similar performance drops are observed in machine learning applications. From localisation in crowded scenes to tracking in environments with multiple objects, visual clutter can have a significant effect on the performance of neural networks. This can be explained, in both humans and Artificial Intelligence (AI), as multiple objects competing for neural representation [22].

Object removal is an application in its own right, where the task is to fill in a “gap” left when an object is removed from an image. That removal is normally defined by a mask, which is required at runtime. Object removal can be used to remove visual clutter from images as a pre-processing step to any task. However, there are important limitations to this approach. Firstly, it requires expensive image processing to be done for every single datum in a dataset and, crucially, also at run-time. Secondly, it requires manual mask annotation of the objects that are to be removed. Finally, the approach must reliably fill the image with temporally consistent content in order to be useful for downstream tasks. Our key insight is that for some applications, it is less important that an object is removed from an image and more important that the computer is capable of completely ignoring that object. In this work, we propose an approach to allow a neural network to ignore chosen objects, not by simply removing them from the image, but by training the network to not “see” them in the first place.

Our approach, which we call Neural Blindness, aims to prevent a neural network from representing chosen objects in its latent space. Humans are capable of ignoring clutter when the task at hand requires them to do so. In a similar way, we train an autoencoder network to remove class-specific objects from its latent representation and demonstrate how this improves the performance of an autonomous agent.

II. LITERATURE REVIEW

In many computer vision tasks involving the use of latent spaces has become commonplace [8], [28], [31]. These latent spaces encode high-level semantic information about the image. Our task aims for the removal of targeted information/objects from the original image such that this information is never encoded in the representation to begin with. The task therefore requires learning invariant representations and the “forgetting” of information.

One way to obtain a latent space that is domain-specific or, equivalently, invariant to certain information, is via the use of Variational Autoencoders (VAEs) [1], [23], [18], [13]. A VAE [13] is a Bayesian autoencoder, whereby the training objective involves reconstruction of a high-dimensional observational distribution from a low-dimensional representation. By regularizing this representation so that it resembles a pre-specified prior distribution (often an isotropic Gaussian), the network undertakes variational inference and learns a smooth, traversable, latent manifold.

VAEs are extremely flexible and have been applied to a broad range of tasks including the generation of high quality images [24], disentangled representation learning [9], [20], object detection [16], motion synthesis [7], and neural image inpainting [8], [31].

Researchers have encouraged invariance or forgetting with VAEs by utilizing various combinations of weak supervision [1], [23], adversarial training [6], and by trading off reconstruction error against encoding capacity according to the information bottleneck principle [25]. Others have sought to
fully disentangle the latent space into independent partitions [14], thereby enabling the isolation of wanted from unwanted information. Whilst unsupervised methods for disentanglement exist [4], [5], recent research indicates that such methods do not achieve disentanglement consistently [19] and, even if they do, they do not allocate factors to the latent space in a predictable way. Furthermore, in order for VAEs to handle complex data and to reconstruct detailed images, the complexity of the latent space is increased, reducing the emphasis on achieving disentangled and semantically interpretable latent spaces [21], [26]. For instance, Vector Quantised VAE (VQ-VAE) [24], [27] deviates from the typical formulation of a VAE and utilizes a deterministic, discrete, vector-quantized codebook as a latent space. In this work, we incorporate a number of techniques into the VAE/autoencoder learning in order to achieve a representation that is invariant to targeted objects.

The latent representations in neural networks can be used for downstream tasks such as classification and regression. In this area, one of the key applications is camera pose regression and the most common family of approaches is PoseNet [12], [10], [11], [3] and its derivatives. Fundamentally, these approaches rely on sensor/pose pairs to train a CNN that can regress pose given an image. Normally, they start with a pre-trained encoder network and modify the final fully connected layers to regress 6-Degrees of Freedom (DoF) camera poses. The network then learns a mapping between the latent representation in the encoder, and the camera poses. A key limitation of these approaches is that they indiscriminately map latent representations to poses. More explicitly, the network may learn to map a parked red corvette to a set of camera poses regardless of the fact the car may move. As such, the network may not be able to successfully recover the poses when the car moves or indeed predict them at different locations where red corvettes exist. Work by Barnes et al. [2] has attempted to explicitly remove objects using an ephemerality mask, but relies on pre-mapped areas using LiDAR. While invariance to such salient objects might be achieved with sufficient data, acquiring it is not always feasible. In this work, we propose to explicitly remove such transient objects from the latent space of a network before it is ever trained for localisation. Such a network would have the ability to learn to regress poses in areas with large amounts of distractors, such as car parks and crowded areas, with relatively small amounts of data.

III. METHODOLOGY

It is tempting to think of neural blindness as a two-step process of identification and removal. However, the purpose of this paper is to achieve a subtle and more complicated task. Class-specific neural blindness requires the neural network to be fundamentally incapable of representing those classes in its latent space. Therefore, identifying the object before it can be removed fundamentally defeats the purpose, as the object has to be present in the latent space for removal to occur in the generated image. Our architecture, loss functions and pre-training strategies force the network to learn how to represent images while removing the presence of specific classes from its latent space.

A. Architecture

Our goal is to be able to remove a chosen class \( c \in C \), where \( C \) is the set of all classes in the dataset, from the latent space of the network. In order to accomplish this, we adopt a Siamese architecture where each arm is a VQ-VAE, shown in Figure 1. This means that we can feed the network two versions of every image: the original image \( x_c \) that does not contain \( c \) and \( x_{c'} \) where the \( c \) has been added to the image. Since VQ-VAEs are capable of producing both an informative embedding and a crisp reconstruction, we can compare the two arms of the network to ensure that the arm that processes \( x_{c'} \) does not learn to represent \( c \). We use a VQ-VAE as we have found, empirically, that a standard VAE struggles produce the informative embedding and crisp reconstruction required for training.

The VQ-VAEs employed in the Siamese architecture are two-level hierarchical, which means they each sample from two trained code-books. The quantised output of both code-books are then concatenated into a single latent space, shown in Figure 1 as \( e_c \) and \( e_{c'} \) for \( x_c \) and \( x_{c'} \), respectively. These subsets of the code-book are fed to the decoder arms in order to generate the output images. Employing a hierarchical model allows our approach to specialise the code-book such that the top code-book encodes low-frequency image detail while the bottom encodes high-frequency detail. Since the code-books are trained to focus on different image “levels” it is easier to discriminate for class-specific elements.

In order to ensure the code-book is trained to correctly encode the images, we use a quantisation loss to train our latent space. Given an input image \( x_c \), encoder \( E \), quantiser \( Q \) and decoder \( D \), the loss function for the code-book is

\[
L_q (x_c) = ||s_g[E(x_c)] - e_c||_2 + \beta ||s_g[e_c] - E(x_c)||_2 \tag{1}
\]
where \( e_c = Q(E(x)) \) is the quantised encoder output and \( s_g \) defines a stop-gradient operation that prevents backpropagation. The first term in equation 1 ensures the codebook learns to represent the input correctly, while the second term ensures the codebook does not keep changing and instead “commits” to a set of vectors. Therefore, \( \beta \) describes how reluctant the network is to update its codebook. We also adopt the standard practice of [27] and replace the first term with an exponential moving average update step for the codebook.

A standard implementation of a VQ-VAE would combine this loss with a reconstruction loss defined as

\[
\mathcal{L}_r(x_c, e_c) = ||x_c - D(e_c)||_2 \tag{2}
\]

which forces the network to learn a codebook that accurately reconstructs the image. However, our objective is to ensure the network does not encode a certain class \( c \) into its codebook. Therefore, we modify the loss in equation 2 as:

\[
\mathcal{L}_r(x_{c'}, e_{c'}) = ||(x_{c'} - D(e_{c'})) \odot (1 - M_z^c)||_2 \tag{3}
\]

where \( \odot \) denotes element-wise multiplication and \( M_z^c \) is a mask for the class \( c \) we want to ignore. More explicitly, we ensure the network gets no reward for reconstructing the target class in the image.

The final VQ-VAE loss can then be defined as

\[
\mathcal{L}_{VQ} = \mathcal{L}_r + \gamma_q \mathcal{L}_q \tag{4}
\]

where \( \gamma \) is a hyperparameter.

These losses ensure that the network is capable of learning a codebook that can successfully reconstruct an image. However, while there is no reward for reconstructing the class \( c \), there is nothing in these loss functions that actively prevents the network from learning global features that can reconstruct the target class. This can happen because the network is never actively discouraged from learning the object representation. Therefore, a loss is required to actively prevent the network from learning to encode and reconstruct the target classes.

To accomplish this, we use a Siamese loss consisting of two individual losses. The first, ensures that the objects are not represented in the latent space, the second, that the objects are not present in the reconstructed image.

We assume the existence of an image pair \( x_c \) and \( x_{c'} \), where the presence of the target class \( (c) \) is the only difference. In principle, this would limit our approach to datasets where the target classes are added and removed from scenes while the images are being captured. Instead, we propose to generate this from pre-existing datasets in order to train and evaluate our approach. Given a set of object classes \( c \in \mathcal{C} \), where \( \mathcal{C} \) is the set of all classes in the dataset (\( \mathcal{Z} \)), we split the dataset into a subset of images that contain the category \( z_c \in \mathcal{Z}_c \) and images that do not contain the category \( z_c \in \mathcal{Z}_{c'} \) such that \( \mathcal{Z}_c \cap \mathcal{Z}_{c'} = \emptyset \) and \( \mathcal{Z}_c \cup \mathcal{Z}_{c'} = \mathcal{Z} \). For every image \( z_c \) that does not contain the chosen class, we randomly select an image \( z_{c'} \) that does and use a mask \( M_z^c \) to overlay one of the class objects into \( z_c \). This allows us to define an overlaid image as

\[
x_{c'} = O(z_c, z_{c'}, M_z^c) \tag{9}
\]

where \( O(\cdot) \) is the function that overlays a category from \( z_{c'} \) to \( z_c \). Using this overlay, a set of 3 images can be created: the original image \( x_c = z_c \), the original image with a category overlaid \( x_{c'} = O(z_c, z_{c'}, M_z^c) \) and a mask image \( M_z^c \) which defines the location of the overlaid class on \( x_{c'} \). These overlay images can be used in equations 5 and 6. At training time, these images are automatically generated therefore randomizing the overlay at every iteration. Although the images are sufficient to train the system to be blind, the network is never exposed to the classes in their natural context which may limit the networks ability to generalise outside of the training data. To address this concern, we show the

\[
\mathcal{L}_o(x_c, e_{c'}) = ||(x_c - D(e_{c'})) \odot M_z^c||_2 \tag{6}
\]

which ensures the network does not reconstruct the target class.
network natural images while training, which can be used in equation 3 to help the network generalise to natural data.

As previously mentioned, during training time, this dataset is dynamic, each iteration presents a new opportunity for the classes to be overlaid on different images, preventing the network from overfitting to the overlay while ensuring performance on natural images.

IV. RESULTS

In this section, we describe the use of a pre-trained latent space for an autonomous vehicle localisation task where blindness to visual clutter (cars) improves localisation performance. We also demonstrate qualitative “blindness” to objects by decoding latent spaces where the object has been removed.

A. Localisation Task

A useful downstream task for neural blindness is localisation. Most learning-based localisation techniques employ an encoder to map an image to a latent space, then regress the pose of the camera. However, such approaches to localisation can easily overfit and often fail to generalise to new sequences. For example, training a system to learn pose from sequence A and then regress the correct pose on unseed sequence B means the latent space must be capable of overcoming environmental or structural changes between the two sequences. Consider elements of the scene that easily change such as people in crowded scenes or cars in automotive datasets. These factors need to be ignored so that localisation can focus on the consistent structure of the environment.

Pose-regression networks should be able to learn to ignore distractors. However, it isn’t always feasible to capture sufficient data representing the variability needed. In this section, we demonstrate that our neural-blind network is capable of mapping images to a more useful feature space by actively ignoring distractors, allowing us to train a distractor-robust pose-regression network without providing variance in the training data and potentially allowing one-shot training.

1) Car-Park Localisation: We first focus on localisation of a car in a multistory carpark. This is an ideal use-case for pose-regression as indoor carparks are often GPS denied. Furthermore, the very nature of a car park means that much of the visual input to any localisation system will include other cars, all of which have different appearance and change on a daily basis. Therefore, any localisation system needs to ignore cars and base its localisation on the structure of the building (as humans do).

We first pre-train a neural-blind network that cannot see cars using the MSCOCO dataset. Once the network is pre-trained, we fix the encoder and feed the representation space to a pose-regression network in order to estimate 6-DoF pose estimates for each input image. This architecture can be seen in Figure 2. It should be noted that the encoder part of the network is not fine-tuned during training of the pose regressor, nor are masks of the distractors required. Instead, we simply learn to regress a pose from a vehicle-blind representation space. Regressing poses from a representation that is incapable of encoding vehicles should lead to better one-shot learning and pose regression. To train the pose regression network we use a dataset that consists of three separate runs through a 6-floor multi-storey carpark. The runs were captured over two days, with two captures on the first day and a third capture on the second day. Each of these sequences traverses the entire 6 floors. We will refer to these sequences as D1T1, D1T2 and D2T1 to represent the Day (D) and Trajectory (T). The capture vehicle was equipped with 3 front-facing cameras and a 16-beam LiDAR. Of the three front facing cameras, there are two narrow field-of-view cameras (Left and Right) and a wide-angle (Centre) which we use as input to the pose regressor. The vehicle traversed 6 floors, over 6,500m of driving and a volume of 3000m³. We use the LiDAR to create the ground truth poses and extract between 2500 and 4000 images per-camera, per-trajectory.

2) Cross-Floor Validation: In this experiment we exploit the fact that the floors of the multistory carpark are structurally identical, as can be seen in Figure 3. However, despite structural similarity of the carpark, the appearance varies drastically due to the different population of parked cars. To test the effect of neural blindness we trained three different networks (Blind, Non-Blind and PoseNet [11]) on the second floor of the multistory carpark and evaluated 2D localisation
performance for floor one across D1T1, D1T2 and D2T1.

We train a neural-blind pose regression network for 100 epochs, with a learning rate of 0.0001 using an ADAM optimizer and a step learning rate scheduler with $\gamma = 0.5$. We use the same hyperparameters to train a non-blind network. As in other work, [12], we report the median error in meters and evaluate our networks against a state-of-the-art PoseNet [11] trained on the same data. All networks are trained using a homеседastic loss ([11]). For PoseNet, we use a pre-trained ResNet as an encoder and allow the network to fine-tune the entire architecture. For our networks, we do not fine-tune the encoders, and expect the network’s representation space to be robust enough for pose regression.

As can be seen in Table I, PoseNet has the lowest training error suggesting it can memorise the images. However, while the Non-blind VQ-VAE has comparable performance to PoseNet on the unseen sequences (D1T1, D1T2, D2T1), the Car-blind VQ-VAE has a significantly reduced localisation error. This demonstrates the benefit of a localisation framework that is blind to the distractors (cars) in the environment.

In a second experiment, we train both a neural blind VQ-VAE Pose regression network and a standard PoseNet on the entire D1T2 (6 Floors) sequence to regress the full 6 degree of freedom vehicle pose. We train both networks for 500 epochs. We then test both networks on the full 6-floors, using unseen sequences D1T1 and D2T2. Tables II and III show the performance of PoseNet [11] as well as our approach. Our Neural-Blind Pose regression can easily outperform the standard PoseNet, reducing the error by a significant margin. This is because PoseNet cannot identify the cars as distractors. In this scenario, PoseNet learns to map car-features to poses, and when those features are modified (or absent), the localisation performance suffers. In contrast, our approach doesn’t see or encode car-features. Therefore, the network must rely on other features to estimate the pose and does not suffer in the presence of high variability of distractors. Since any class can be removed from the latent space, we expect this approach to work in any scene where visual clutter is a problem.

### B. Blindness Decoding

We now demonstrate that our network becomes blind to certain classes by decoding the latent space of a neural-blind VQ-VAE. We show, qualitatively, that the class the network is blind to is not reconstructed. It should be noted that our network is not rewarded for the quality of the image “inpainting” but is simply incapable of representing these classes in its latent space, the result is in-painting by proxy.
TABLE IV: Per-class quantitative results on Overlay Dataset.

![Image with Table]

**TABLE IV:** Per-class quantitative results on Overlay Dataset.

| Approach         | L1   | MSE  | PSNR |
|------------------|------|------|------|
| Person           |      |      |      |
| DeepFill V2 [29][30] (Mask) | 0.214 | 0.123 | 9.68 |
| Ours (No Mask)   | 0.219 | 0.136 | 9.30 |
| Car              |      |      |      |
| DeepFill V2 [29][30] (Mask) | 0.229 | 0.136 | 9.30 |
| Ours (No Mask)   | 0.148 | 0.069 | 13.28 |
| Fire-Hydrant     |      |      |      |
| DeepFill V2 [29][30] (Mask) | 0.215 | 0.125 | 9.55 |
| Ours (No Mask)   | 0.102 | 0.036 | 15.66 |
| Cow              |      |      |      |
| DeepFill V2 [29][30] (Mask) | 0.220 | 0.129 | 9.39 |
| Ours (No Mask)   | 0.119 | 0.046 | 14.33 |
| Fork             |      |      |      |
| DeepFill V2 [29][30] (Mask) | 0.195 | 0.107 | 10.20 |
| Ours (No Mask)   | 0.082 | 0.027 | 16.89 |

**V. CONCLUSIONS**

We have trained a network to not encode certain classes into its latent space and shown this latent space can be used for downstream tasks. This kind of “negative attention” is capable of revolutionising the way we think about certain tasks. It means that visual clutter can be removed from tasks like localisation in crowded scenes or tracking in busy environments. Our contributions allow networks to be trained with smaller amounts of data, as we can choose to remove attention from elements that may cause the network to overfit.

**REFERENCES**

[1] E. Banijamali, A. H. Karimi, A. Wong, and A. Ghodsi. Joint autoencoders for dis-entanglement. arXiv:1711.09163v1, 2017.
[2] D. Barnes, W. Maddern, G. Pascoe, and I. Posner. Driven to distraction: Self-supervised distractor learning for robust monocular visual odometry in urban environments. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1894–1900. IEEE, 2018.
[3] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz. Geometry-aware learning of maps for camera localization. In *IEEE/CVF Conf. on Comp. Vis. and Pattern Recognition (CVPR)*, pages 2616–2625, 2018.
[4] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner. Understanding disentangling in Beta-VAE. arXiv:1804.03599v1, 2018.
[5] E. Dupont. Learning disentangled joint continuous and discrete representations. *Neural Inf. Proc. Sys. (NeurIPS)*, 2018.
[6] Y. Ganin, E. Ustinova, H. Ajakan, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *arXiv:1505.07818*, 2016.
[7] I. Habibie, D. Holden, J. Schwarz, J. Yearsley, and T. Komura. A recurrent variational autoencoder for human motion synthesis. *British Machine Vis. Conf. (BMVC)*, 2017.
[8] C. Ham, A. Raj, V. Cartillier, and I. Essa. Variational image inpainting. *Workshop on Bayesian Deep Learning (NeurIPS)*, 2018.
[9] Z.-H. Jiang, Q. Wu, K. Chen, and J. Zhang. Disentangled representation learning for 3D face shape. arXiv:1902.09887v2, 2019.
[10] A. Kendall and R. Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 4762–4767. IEEE, 2016.
[11] A. Kendall and R. Cipolla. Geometric loss functions for camera pose regression with deep learning. In *IEEE/CVF Conf. on Comp. Vis. and Pattern Recognition (CVPR)*, pages 5974–5983, 2017.
[12] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *IEEE Int. Conf. on Comp. Vis. (ICCV)*, pages 2938–2946, 2015.
[13] D. P. Kingma and M. Welling. Auto-encoding variational Bayes. arXiv:1312.6114v1: 2014.
[14] T. D. Kulkarni, W. Whitney, P. Kohli, and J. B. Tenenbaum. Deep convolutional inverse graphics network. arXiv:1503.03167v4, 2015.
[15] C. H. Lee, Z. Liu, L. Wu, and P. Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *IEEE/CVF Conf. on Comp. Vis. and Pattern Recognition (CVPR)*, 2020.
[16] Z. Li, B. Sun and Y. Guo. SuperV AE: Superpixelwise variational autoencoder for salient object detection. *AAAI Conf. on Art. Int.*, 2019.
[17] G. Liu, F. A. Reda, K. Ji, Shih, T-C. Wang, A. Tao, and B. Catanzaro. Image inpainting for irregular holes using partial convolutions. arXiv:1804.07723v1, 2018.
[18] M. Ilse, J. M. Tomczak, C. Louizos, and M. Welling. DIVA: domain invariant variational autoencoders. arXiv:1905.10427, 2019.
[19] F. Locatello, S. Bauer, M. Lucic, G. Ratsch, G. Gelly, B. Scholkopf, and Bachem O. Challenging common assumptions in the unsupervised learning of disentangled representations. arXiv:1811.12359v3, 2019.
[20] F. Locatello, M. Tschannen, S. Bauer, G. Ratsch, and B. Scholkopf. Disentangling factors of variation using few labels. arXiv:1905.01258, 2019.
[21] L. Maaloe, M. Fraccaro, V. Lievin, and O. Winther. BIVA: a very deep hierarchy of latent variables for generative modeling. *Neural Inf. Proc. Sys. (NeurIPS)*, 2019.
[22] S. McMains and S. Kastner. Interactions of top-down and bottom-up mechanisms in human visual cortex. *Journal of Neuroscience*, 31(2):587–597, 2011.
[23] D. Moyer, S. Gao, R. Brekelmans, G. V. Steeg, and A. Galstyan. Invariant representations without adversarial training. *Neural Inf. Proc. Sys. (NeurIPS)*, 2018.
[24] A. Razavi, A. van den Oord, and O. Vinyals. Generating diverse high-fidelity images with vq-vae-2. arXiv:1906.00446, 2019.
[25] N. Tishby and N. Zaslavsky. Invariant representations without adversarial training. arXiv:1503.02406v1, 2015.
[26] A. Vahdat and J. Kautz. NVAE: A deep hierarchical variational autoencoder. *Neural Inf. Proc. Sys. (NeurIPS)*, 2020.
[27] A. van den Oord, O. Vinyals, and K. Kavukcuoglu. Neural discrete representation learning. *Neural Inf. Proc. Sys. (NeurIPS)*, 2017.
[28] W. Xiong, J. Yu, Z. Lin, J. Yang, X. Lu, C. Barnes, and J. Luo. Foreground-aware image inpainting. *IEEE/CVF Conf. on Comp. Vis. and Pattern Recognition (CVPR)*, 2019.
[29] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T.S. Huang. Generative image inpainting with contextual attention. In *IEEE/CVF Conf. on Comp. Vis. and Pattern Recognition (CVPR)*, pages 5505–5514, 2018.
[30] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T.S. Huang. Free-form image inpainting with gated convolution. In *IEEE Int. Conf. on Comp. Vis. (ICCV)*, pages 4471–4480, 2019.
[31] A. Zadeh, Y.-C. Lim, P. P. Liang, and L-P. Morency. Variational auto-decoder. arXiv:1903.00840v5, 2019.