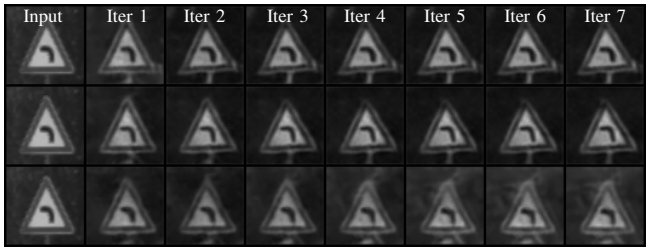
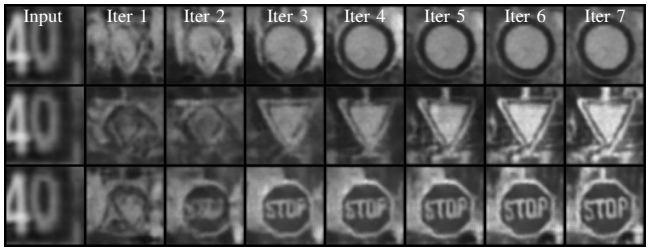
Autoencoder Attractors for Uncertainty Estimation

Steve Dias Da Cruz*†‡, Bertram Taetz‡, Thomas Stifter*, Didier Stricker†‡
*IEE S.A., †University of Kaiserslautern, ‡German Research Center for Artificial Intelligence (DFKI)
Email: steve.dias-da-cruz@iee.lu, bertram.taetz@dfki.de, thomas.stifter@iee.lu, didier.stricker@dfki.de

***Abstract*—The reliability assessment of a machine learning model's prediction is an important quantity for the deployment in safety critical applications. Not only can it be used to detect novel sceneries, either as out-of-distribution or anomaly sample, but it also helps to determine deficiencies in the training data distribution. A lot of promising research directions have either proposed traditional methods like Gaussian processes or extended deep learning based approaches, for example, by interpreting them from a Bayesian point of view. In this work we propose a novel approach for uncertainty estimation based on autoencoder models: The recursive application of a previously trained autoencoder model can be interpreted as a dynamical system storing training examples as attractors. While input images close to known samples will converge to the same or similar attractor, input samples containing unknown features are unstable and converge to different training samples by potentially removing or changing characteristic features. The use of dropout during training and inference leads to a family of similar dynamical systems, each one being robust on samples close to the training distribution but unstable on new features. Either the model reliably removes these features or the resulting instability can be exploited to detect problematic input samples. We evaluate our approach on several dataset combinations as well as on an industrial application for occupant classification in the vehicle interior for which we additionally release a new synthetic dataset.**



(a) Reconstructions of a same test sample from D_{in} (GTSRB)



(b) Reconstructions of a same OOD sample from D_{out} (SVHN)

Fig. 1. Multiple recursive reconstructions (from left to right) of identical samples (first column) from D_{in} and D_{out} by our novel model. Notice the evolution in the reconstructions over each iterative step for the OOD sample.

I. Introduction

Assessing the reliability of machine learning models' predictions is an important challenge for the deployment and applicability of statistical methods. This additional information allows the possibility to detect novel and exotic sceneries during the lifetime of a deployed model on which the model's predictions trustability can be determined. This knowledge also gives hints whether the collected training data needs to be extended or modified, e.g. in the case of active learning [1] and continuous learning [2]. Recent activities investigated the possibility for estimating the uncertainty in the case of deep learning based methods [3]–[6]. Monte Carlo (MC) dropout, i.e. using dropout during training and enabling the latter during inference for multiple runs, has been shown to produce good uncertainty quantification [7] on several tasks while limiting the additional overheat during training and inference.

It has been shown that recursive applications of autoencoders, which are trained under the standard training regime, can be viewed as a dynamical system [8]. In mathematics [9], the analysis of fixed points, attractors and their basins of attraction are important tools to analyze and understand dynamical systems and their behavior. This iterative process can be viewed as associative memory [8] to retrieve perturbed training samples, but the models need to be trained long enough to ensure that the training samples become attractors. To the best of our knowledge, the recursive application of autoencoders and their attractors have not been investigated in view of generalization and uncertainty estimation.

Our contribution consists of the extension of the recursive application of autoencoder models, thus dynamical systems and attractors, in view of generalization capacities. We combine this strategy with MC dropout and we exploit characteristics of both design choices to determine whether new input samples are close or far from the training distribution by analyzing the behavior of multiple inferences, as shown in Fig. 1: the test sample is converging to a similar attractor, while the out-of-distribution (OOD) sample converges to different attractors of different classes. We show that uncertainty estimation is improved compared to vanilla MC dropout and deep ensemble models across three metrics and in view of the entropy distribution. Our ablation study shows that the recursive application is key to the success of our approach. Our analysis is performed on several commonly used OOD dataset combinations as well as on an industrial application. We consider occupant classification in the vehicle interior and highlight some additional challenges. To this end we release a synthetic dataset for uncertainty estimation which will extend the existing SVIRO [10] dataset for occupant classification.

II. Related Works

Attractors: There are several types of models achieving associative memory, e.g. discrete and continuous Hopfield Networks [11]–[13] and Predictive Coding [14]. The former needs an energy function to be defined, while the latter is biologically inspired. However, we focus on associative memory achieved by the recursive application of autoencoder models [8], previously trained with gradient descent, due to their elegant simplicity and analogy to dynamical systems, which has been investigated extensively in mathematics and physics [9]. While a few works investigate properties of this model design [8], [15], [16], only one [17] considers attractors for classification and uncertainty estimation. However, the latter adopts this only for speech recognition with respect to noise robustness and combines it with a hidden Markov model. We, on the contrary, apply this methodology to computer vision and assess the robustness against novel classes and unseen samples from either new datasets or the test distribution.

Uncertainty estimation: A lot of research [18] is focusing on estimating the uncertainty of a model's prediction regarding OOD or anomaly detection, both of which are tightly related. However, only a few works consider the use of autoencoder models for assessing uncertainty: Autoencoders can be combined with normalizing flow [19], refactor ideas from compressed sensing [20] or use properties of Variational Autoencoders [21], [22]. More commonly, autoencoders are used for non image based datasets [23]–[25]. Other deep learning approaches are based on evidential learning [6], [26], Bayesian methods [27], Variational Bayes [28] or on Hamiltonian Monte-Carlo [29]. Also non deep-learning approaches have shown significant success, but are less scalable, as for example Gaussian Processes [30] or approaches based on support vector machines [31]. Since our approach borrows ideas from MC dropout [7], we limit our comparison against the latter and the commonly used deep learning golden standard of using an ensemble of trained models [32], [33].

III. Method

We start by introducing both approaches, dynamical systems based on autoencoders and their attractors and uncertainty estimation by MC dropout. Next we introduce our method, which we call Monte-Carlo Attractor Autoencoder (MCA-AE), combining both of the aforementioned design choices.

A. Preliminaries - Attractors

A good overview on the basic analysis of autoencoders, associative memory and attractors is provided in [8]. Let f be an autoencoder trained under the standard training regime, i.e. minimizing the reconstruction loss $\mathcal{L}$ between input x and target $f(x)$, i.e. $\mathcal{L} = \mathrm{r}\left(f(x) - x\right)$, where $\mathrm{r}(\cdot)$ is a reconstruction loss of choice. Consider an input sample x, an index set $\mathcal{I} = \{1, 2, \ldots, N\}$ for some $N \geq 1$ and the sequence $\{f^k(x)\}_{k\in\mathcal{I}}$, where $f^k(x) = \left(f \circ f \circ \cdots \circ f\right)(x)$ (k times) denotes k compositions of f applied to x. A point x is a **fixed point** x^* of f if $f(x) = x$, where we allow the equality to be weakened, i.e. $f(x) = x + \epsilon \approx x$ for some small ϵ, because the reconstruction will never be perfect. The sequence $\{f^k(x)\}_{k\in\mathcal{I}}$ then converges to x^*. A fixed point x^* is an **attractor** of f if there exists an open neighborhood $\mathcal{O}$ around x^* such that for all $x \in \mathcal{O}$ the sequence $\{f^k(x)\}_{k\in\mathcal{I}}$ converges to x^* if $k \to \infty$. The set of all such points is called the **basin of attraction** of x^* for f. Even disturbed training samples converge to the initial training sample [8]. We show that this property can be used to generalize to test samples, when they are close enough to the training distribution. If the latter is violated, the sample might not be stable in its convergence, which will be exploited by our next design choice.

B. Preliminaries - MC Dropout

The use of dropout during training and inferences, called Monte Carlo (MC) dropout, has been introduced [7] to model uncertainty in neural networks without sacrificing complexity or test accuracy for several machine learning tasks. For standard classification or regression models, an individual binary mask is sampled for each layer (except the last layer) for each new training and test sample. Consequently, neurons are dropped randomly such that during inference we sample a function f from a family, or distribution of functions $\mathcal{F}$, i.e. $f \in \mathcal{F}$. Uncertainty and reliability can then be assessed by performing multiple runs for the same input sample x, i.e. retrieve $\{f_j(x)\}_{j\in\mathcal{J}}$ for $\mathcal{J} = \{1, 2, \cdots, M\}$ for some $M \geq 1$. The models predictive distribution for an input sample x can then be assessed by computing $p = f(x) = \frac{1}{M}\sum_{j=1}^{M} \mathrm{softmax}(f_j(x))$. Uncertainty can be summarized by computing the normalized entropy [34] of the probability vector p, i.e. $H(p) = -\frac{1}{\log(C)}\sum_{c=1}^{C} p_c \log(p_c)$, where C is the number of classes. We use the latter in all our experiments to compute the uncertainty of the prediction and decide based on its value whether a sample is rejected or accepted for prediction or whether the sample is in- or out-of-distribution.

C. MCA-AE

Our introduced method is a combination of both previously detailed model designs. Instead of training the autoencoder model under a standard training regime as done by related works thus far, we train the model using dropout and enabling dropout during inference as well. This causes an interesting model feature: if we repeat the recursive application of the trained autoencoder several times for the same input sample x, then each iteration uses a different function f from the same distributions of functions $\mathcal{F}$. Hence, we obtain different, but similar, dynamical systems for inference which should behave similarly for training and test samples, but not consistently for novel feature variations in the input. Each iteration can hence converge to a different attractor, potentially of different classes. The latter is useful to detect inconsistencies and hence uncertainty: if the model converges to attractors of the same class we can assume a trustful prediction, if it converges to attractors of different classes the convergence is unreliable.

MCA-AE: Let x be an input sample and $\mathcal{F}$ be the family of functions consisting of autoencoders learned by using dropout during training and enabling it during inference as well. We

repeat the recursion M times, sampling each time a new f_j for each recursion $\mathcal{J} = \{1, 2, \cdots, M\}$. This results in a predictive distribution $\{f_j^k(x)\}_{j \in \mathcal{J}}$, where k is the number of compositions performed for each recursion. As a reminder, for a fixed f_j the dropout mask is the same for each recursive step k. The latter implies that the dropout mask needs to be implemented manually such that it can be fixed for multiple inferences. Since we are adopting this strategy for autoencoders, we refrain from using dropout in the latent space. Classification of the resulting iteratively reconstructed sample is performed in the latent space of the kth iteration. For the latter we use a MLP classifier with a single hidden layer of the same size as the latent dimension. To summarize this heuristic:

Train autoencoder model using dropout to get $\mathcal{F}$
Enable dropout for inference
Define the number of recursions N
Train classifier $\mathrm{g}(\cdot)$ in latent space after N recursions
Define the number of inferences per sample M
Define uncertainty threshold U
for each input sample x **do**
  for $l \leftarrow 1$ to M **do**
    for $k \leftarrow 1$ to N **do**
      if $k = 1$ **then**
        Sample a new dropout mask and keep it fixed
        This gets you $f_j \in \mathcal{F}$, where $f_j(x) = d_j(e_j(x))$
      end if
      $z = e_j(x)$ {encoding}
      $x = d_j(z)$ {decoding}
    end for
    $y_l = g(z)$ {probability distribution of classification}
  end for
  $p(y) = \frac{1}{M} \sum_{l=1}^{M} y_l$
  $H(p(y)) = -\frac{1}{\log(C)} \sum_{c=1}^{C} p_c(y) \log(p_c(y))$
  if $H(p(y)) \leq U$ **then**
    $y = \mathrm{argmax}(y_l)$
  else
    Reject sample
  end if
end for

For training samples to become attractors it is necessary to train the autoencoders for a large number of epochs, i.e. we used 25000. A lot of hyperparameters are defined for the inference process instead of the training process. The number of recursions and the number of different runs is independent from the training. The classifier can be chosen after the autoencoder training. The uncertainty threshold needs to be adapted according to the use case and it is a tradeoff between the required sensitivity and precision.

IV. Experiments

We evaluate our method on two scenarios: First, we want to assess the predictive uncertainty where the model should provide a high uncertainty in case it wrongly classifies a sample. This is made more difficult in the case of the vehicle interior: unseen objects should be classified as empty seats, i.e. the model should only identify known classes and neglect everything else. Our results will show that this is a challenging task. Second, the model should differentiate between in- and out-of-distribution (OOD) samples. In the case of training on MNIST and evaluating on Fashion-MNIST, the model cannot perform a correct prediction and it should detect the OOD as such. This is also the case when images from a new vehicle interior are provided as input to the model. All training and evaluation scripts can be found in our implementation (link).

TABLE I
Overview of the number of classes and samples for OOD or uncertainty estimation for the different datasets used.

Dataset	Classes	D_{in} and D_{out}	Uncertainty
MNIST	10	2500	10000
Fashion	10	2500	26032
SVHN	10	2500	10000
GTSRB	10	2006	3208
CIFAR10	10	2500	-
Omniglot	660	2636	-
LSUN	10	2500	-
Places365	365	2555	-
SVIRO-U Adults (A)	7	1337	2617
SVIRO-U Seats (S)	8	-	490
SVIRO-U Objects (O)	8	-	1622
SVIRO-U A,S	26	-	896
SVIRO-U A,O	7	-	1421
SVIRO-U A,S,O	30	-	1676
SVIRO Tesla	21	-	2000

A. Evaluation metrics

According to standard evaluation criterions adopted in related works, we evaluate our models using the Area Under the Receiver Operating Characteristic curve (AUROC), Area Under the Precision-Recall curve (AUPR) and the false positive rate at 95% true positive rate (FPR95%). For OOD evaluation we use approximately 50% of the samples from the test set $\mathcal{D}_{in}$ and 50% from the test set from $\mathcal{D}_{out}$. For further details and interpretations of the metrics we refer to [35]–[39].

B. Datasets

We use several commonly used computer vision datasets for training and use the corresponding test data as in-distribution sets $\mathcal{D}_{in}$: MNIST [40], Fashion-MNIST [41], SVHN [42] and GTSRB [43] (which we reduce to use 10 classes only). For out-of-distribution $\mathcal{D}_{out}$ we use a subset of all $\mathcal{D}_{in}$ not coming from the training distribution and the test datasets from Omniglot [44], CIFAR10 [45], LSUN [46] (for which we use the train split) and Places365 [47]. We use approximately the same number of samples from $\mathcal{D}_{in}$ and $\mathcal{D}_{out}$ by sampling each class uniformly. An overview is provided in Table I.

In addition to these commonly used datasets, we release an extension for SVIRO [10] called SVIRO-Uncertainty. For each of the 3 seat positions in the vehicle interior rear bench the model should classify which object is occupying it, with empty being one possible choice. We created two training datasets for the Sharan vehicle, one using adult passengers only (4384 sceneries and 8 classes) and one using adults,

Fig. 2. Examples from the SVIRO-Uncertainty dataset. First row are training samples of adults only. Second row are test samples of unseen adults, but also child-seats and everyday objects which should be classified as empty.

child seats and infant seats (3515 samples and 64 classes - not used for training in this work). We created fine-grained test sets to asses the reliability on several difficulty levels: 1) only unseen adults, 2) only unseen child and infant seats, 3) unseen adults and unseen child and infant seats, 4) unknown random everyday objects (e.g. dog, plants, bags, washing machine, instruments, tv, skateboard, paintings, ...), 5) unseen adults and unknown everyday objects and 6) unseen adults, unseen child and infant seats and unknown everyday objects. The dataset can been downloaded (link). Besides the uncertainty estimation within the same vehicle interior, one can use images from unseen vehicle interiors from SVIRO to further test the models reliability on the same task, but in novel environments, i.e. vehicle interiors. Example images are provided in Fig. 2.

C. Training and evaluation details

We compare our method against MC dropout and an ensemble of models using the same architecture as the autoencoder encoder part, but with an additional classification head. We trained our MCA-AE models for 25000 epochs, but fewer epochs might produce good results as well. We did not perform an ablation study with respect to the number of epochs needed. Further, we did not check whether the training samples are truly fixed point and attractors because of the computational overhead: This could be done by computing the largest eigenvalue of the Jacobian matrix for each training sample and checking whether its greater than 1. The autoencoder model was trained as a denoiser [48] (blur, random noise, brightness and contrast augmentation were used) to facilitate and robustify the recursive autoencoder application. Consequently, to have a fair benchmark, MC dropout and ensemble models used the same augmented images during training. The latter were trained for 1000 epochs. All methods used Adam, a learning rate of $1e^{-4}$ and a batch size of 64. For training on MNIST and Fashion-MNIST we used a latent space of 10, while for all others we used a latent space of 64. We used SSIM [49] for computing the reconstruction loss. We used 250 samples per class for training and treat all datasets as grayscale images. All images were centre cropped and resized to 64 pixels. We used a dropout rate of 0.33 for all methods. Model and training details can be found in the implementation.

For MCA-AE and MC dropout we used 20 inferences and we used an ensemble of 10 models to assess uncertainty and the OOD estimation. We repeated each training for 10 runs for MCA-AE and MC Dropout and for 100 runs to get the ensembles of models. We used 2 recursions for MCA-AE, but this value depends on the dataset used and it is subject to a hyperparameter search. In our case, the models converged fast for test and slow for OOD samples, see Fig. 3. Hence, more iterations did not provide an improvement.

D. Uncertainty estimation and out-of-distribution detection

We report the summary of our results for uncertainty and OOD detection in Table II. An interesting observation is the result that our approach performs significantly better when the visual complexity is increased (GTSRB, SVIRO), while the performance of MC Dropout and ensemble of models decreases on those setups. On the other side, on visually much simpler datasets (MNIST, Fashion-MNIST, SVHN) the performance of MC dropout and ensemble of models performs best. Thus, our method seems to be more beneficial for higher visual complexity, but this behavior should be investigated in detail in future work. Another interesting observation is that our approach provides better OOD estimations for the unseen Tesla vehicle from SVIRO. It can be observed that the different SVIRO-Uncertainty splits are much more challenging and undergo a large performance gap for all methods.

We computed the histograms of the entropies for each D_{in} and D_{out} and report the results in Fig. 4 when trained on GTSRB. The results show that the entropy distribution between D_{in} and several D_{out} are best separated by our approach. The distributions of the different D_{out} are more similar then for the other models. To quantify this, we computed the sum of the Wasserstein distances between D_{in} and all D_{out} (TD, larger is better, as we want them to be different) separately and the sum of the distances between D_{out} CIFAR10 and all other D_{out} (OD, smaller is better, as we want them to be similar). We then computed the mean and standard deviation across 10 runs. The results are reported in Table III and show that our method best separates uncertainty between D_{in} and D_{out}. Further, all D_{out} are most similar between each other.

E. Ablation study

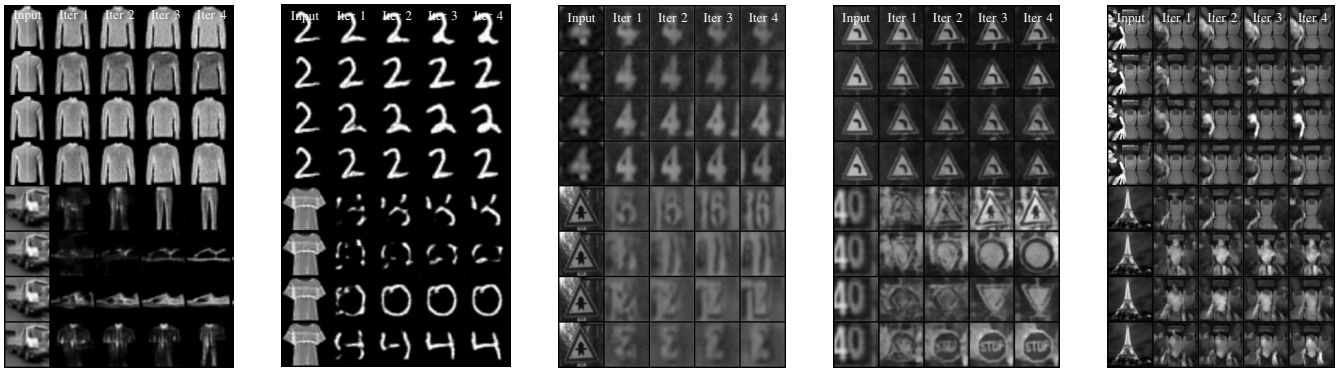
We want to highlight that the performance of our method is improved due to the recursive application of the previously trained autoencoder. To this end we provide additional results where we compare the performance if no recursion is applied. We repeat the evaluation from the previous section and report the performance in Table IV. By comparing the results against Table II, it becomes apparent that the recursive application significantly improves uncertainty and OOD estimation.

In Fig 3 we report the reconstructions after 1, 2, 3 and 4 iterative steps. We repeat this for models trained on different D_{in} and show that D_{out} reconstructions converge over time (and much slower) to training samples. We hence believe that considering the trajectory of the latent space representation over several steps can be an additional indicator whether an input sample is in- or out-of-distribution. It becomes also visible that the reconstruction converges robustly to similar classes for D_{in} samples, but to different classes for D_{out}.

TABLE II

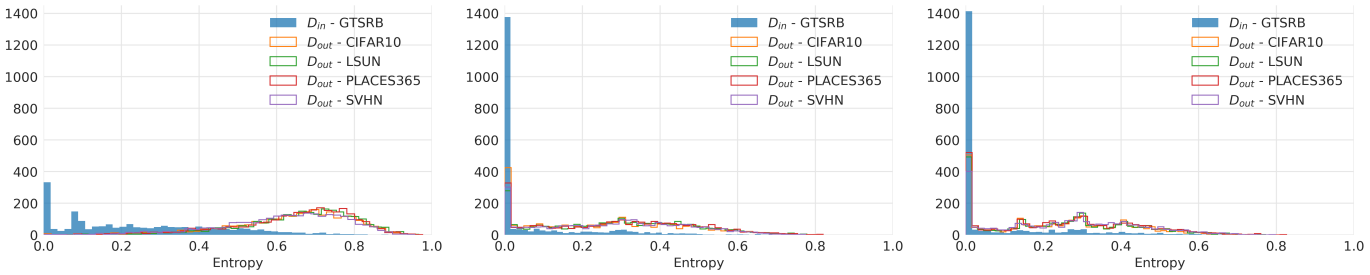
COMPARISON (IN PERCENTAGE) OF OUR METHOD AGAINST MC DROPOUT AND AN ENSEMBLE OF MODELS. WE REPEATED THE EXPERIMENTS FOR 10 RUNS AND REPORT THE MEAN VALUES TOGETHER WITH THEIR STANDARD DEVIATION. IF $\mathcal{D}_{in} = \mathcal{D}_{out}$, THEN WE REPORT THE RESULT ON THE TEST SET OF $\mathcal{D}_{in}$ ONLY. ARROWS INDICATE WHETHER LARGER ↑ OR SMALLER ↓ IS BETTER. BEST RESULTS ARE HIGHLIGHTED IN GREY. THE LAST BLOCK IS A COMPARISON ON THE FINE-GRAINED SPLITS ON THE NEWLY RELEASED SVIRO-UNCERTAINTY. ALL BUT ADULTS SHOULD BE CLASSIFIED AS EMPTY.

	MCA-AE (Ours)			MC Dropout			Ensemble of 10 models		
$\mathcal{D}_{in} \rightarrow \mathcal{D}_{out}$	AUROC ↑	AUPR ↑	FPR95 % ↓	AUROC ↑	AUPR ↑	FPR95 % ↓	AUROC ↑	AUPR ↑	FPR95 % ↓
MNIST									
→MNIST	79.9 ± 1.6	94.9 ± 0.5	74.2 ± 4.8	90.1 ± 0.6	99.6 ± 0.1	28.0 ± 2.4	85.8 ± 1.4	99.0 ± 0.1	41.5 ± 3.0
→CIFAR10	88.2 ± 2.3	87.4 ± 2.2	44.1 ± 8.3	91.2 ± 1.3	92.2 ± 1.1	39.8 ± 5.1	91.5 ± 1.1	92.4 ± 0.9	34.0 ± 4.7
→Fashion	74.5 ± 3.2	72.7 ± 3.3	72.3 ± 4.4	90.0 ± 1.6	91.1 ± 1.3	40.5 ± 5.9	89.5 ± 1.1	90.6 ± 0.9	37.0 ± 3.2
→Omniglot	64.4 ± 5.0	70.4 ± 5.7	99.4 ± 0.7	93.4 ± 2.8	94.2 ± 2.5	35.4 ± 12.2	95.5 ± 1.0	96.0 ± 0.8	22.0 ± 6.0
→SVHN	92.2 ± 2.0	91.3 ± 1.5	28.2 ± 12.4	94.2 ± 1.7	94.9 ± 1.4	30.5 ± 9.4	94.9 ± 0.9	95.4 ± 0.7	22.4 ± 5.1
Fashion									
→Fashion	81.0 ± 1.0	94.4 ± 0.3	80.2 ± 2.4	82.5 ± 0.4	96.4 ± 0.1	64.4 ± 4.3	81.7 ± 0.7	96.4 ± 0.1	64.7 ± 2.2
→CIFAR10	93.9 ± 1.8	95.8 ± 1.1	47.0 ± 20.0	88.7 ± 1.9	89.6 ± 1.8	45.7 ± 5.8	91.6 ± 0.9	92.1 ± 0.8	34.3 ± 3.1
→MNIST	87.8 ± 4.0	88.2 ± 3.4	48.7 ± 15.6	85.4 ± 1.8	86.7 ± 1.5	53.5 ± 5.1	90.2 ± 0.5	90.6 ± 0.5	35.7 ± 2.4
→Omniglot	86.8 ± 3.8	91.2 ± 2.5	87.3 ± 9.7	93.6 ± 2.0	94.1 ± 1.8	32.6 ± 10.1	97.9 ± 0.4	98.1 ± 0.3	9.3 ± 2.3
→SVHN	93.7 ± 2.0	95.6 ± 1.3	48.9 ± 17.1	90.8 ± 1.0	91.7 ± 0.9	40.7 ± 3.6	94.8 ± 0.5	95.1 ± 0.4	23.0 ± 2.6
SVHN									
→SVHN	77.6 ± 0.8	80.8 ± 1.0	79.5 ± 2.1	84.0 ± 0.6	93.1 ± 0.4	69.3 ± 2.4	83.7 ± 0.5	92.9 ± 0.3	68.7 ± 2.0
→CIFAR10	77.5 ± 1.2	80.4 ± 1.1	83.6 ± 3.0	74.9 ± 0.9	78.0 ± 0.8	85.8 ± 1.8	77.6 ± 0.7	80.5 ± 0.6	83.3 ± 1.4
→GTSRB	75.4 ± 2.2	80.5 ± 1.9	80.7 ± 5.4	74.0 ± 1.1	80.1 ± 1.0	84.9 ± 2.9	75.3 ± 0.7	81.2 ± 0.7	84.0 ± 3.0
→LSUN	78.4 ± 0.9	81.5 ± 0.8	82.7 ± 4.8	77.0 ± 0.7	79.8 ± 0.7	81.9 ± 2.1	79.2 ± 0.7	81.9 ± 0.7	80.1 ± 1.9
→Places365	78.5 ± 0.8	81.0 ± 0.7	82.6 ± 3.7	77.1 ± 0.6	79.4 ± 0.6	80.9 ± 2.5	79.2 ± 0.5	81.5 ± 0.4	79.5 ± 1.9
GTSRB									
→GTSRB	85.1 ± 0.9	95.6 ± 0.5	69.3 ± 3.3	89.3 ± 2.4	98.8 ± 0.3	50.9 ± 6.0	84.6 ± 1.7	97.4 ± 0.3	62.1 ± 3.2
→CIFAR10	91.4 ± 0.6	90.3 ± 0.8	42.0 ± 3.3	81.2 ± 0.9	81.4 ± 0.9	69.5 ± 3.7	76.3 ± 0.5	77.7 ± 0.5	83.4 ± 1.3
→LSUN	93.0 ± 0.7	92.2 ± 0.7	36.5 ± 4.4	83.4 ± 0.8	83.3 ± 0.7	65.3 ± 3.9	77.7 ± 0.8	78.7 ± 0.6	81.3 ± 1.6
→Places365	92.3 ± 0.7	91.3 ± 0.7	38.8 ± 3.4	82.8 ± 0.7	82.4 ± 0.6	65.1 ± 3.6	77.5 ± 0.6	78.2 ± 0.6	80.6 ± 1.7
→SVHN	91.3 ± 0.7	90.7 ± 0.8	44.5 ± 3.7	85.6 ± 1.5	85.5 ± 1.5	60.7 ± 5.1	79.4 ± 0.6	80.3 ± 0.5	80.1 ± 1.7
SVIRO-U									
→CIFAR10	95.4 ± 0.6	93.3 ± 1.0	26.9 ± 3.4	74.6 ± 3.5	73.6 ± 2.0	60.4 ± 7.2	77.7 ± 1.5	75.0 ± 1.1	57.1 ± 3.2
→GTSRB	95.8 ± 1.0	94.9 ± 1.1	25.1 ± 6.9	69.9 ± 2.7	74.2 ± 1.2	68.8 ± 4.7	74.7 ± 2.5	76.2 ± 1.5	63.8 ± 1.3
→LSUN	94.8 ± 0.5	92.7 ± 0.7	31.5 ± 2.7	67.6 ± 2.0	70.1 ± 1.0	72.3 ± 4.2	72.0 ± 1.1	71.6 ± 0.6	64.4 ± 2.3
→Places365	95.4 ± 0.5	93.3 ± 0.7	27.3 ± 2.8	73.2 ± 2.6	72.5 ± 1.3	63.5 ± 6.8	77.4 ± 1.0	74.5 ± 0.7	57.0 ± 2.5
→SVHN	92.4 ± 1.6	88.6 ± 2.3	40.1 ± 7.6	81.0 ± 3.4	77.8 ± 2.4	49.5 ± 8.9	81.0 ± 1.3	77.3 ± 1.1	51.6 ± 4.1
→Adults (A)	87.8 ± 1.3	99.1 ± 0.3	62.9 ± 3.9	95.2 ± 1.7	99.9 ± 0.1	8.9 ± 3.1	91.1 ± 1.9	99.8 ± 0.1	28.8 ± 8.8
→Seats (S)	54.0 ± 7.5	8.9 ± 4.2	88.8 ± 10.8	17.5 ± 13.1	0.4 ± 0.2	95.7 ± 5.1	28.1 ± 6.8	2.6 ± 1.6	98.0 ± 2.5
→Objects (O)	68.9 ± 3.1	83.7 ± 2.4	84.1 ± 5.5	64.7 ± 3.2	85.3 ± 3.3	85.3 ± 4.6	57.4 ± 2.3	80.6 ± 1.4	86.5 ± 3.3
→A,S	58.8 ± 2.6	48.6 ± 6.4	93.2 ± 1.1	36.3 ± 2.3	16.5 ± 2.8	97.4 ± 1.4	39.5 ± 1.7	23.6 ± 1.8	96.9 ± 1.1
→A,O	78.8 ± 1.5	93.0 ± 0.5	76.1 ± 3.4	70.1 ± 1.6	93.5 ± 0.9	77.4 ± 2.8	71.1 ± 0.7	92.3 ± 0.6	77.8 ± 2.4
→A,S,O	62.2 ± 2.0	56.4 ± 4.7	88.7 ± 3.1	42.1 ± 2.7	18.6 ± 2.7	96.4 ± 0.9	45.8 ± 1.9	33.0 ± 1.7	95.5 ± 0.9
→Tesla (OOD)	88.6 ± 2.0	97.4 ± 0.5	58.0 ± 6.1	52.1 ± 2.9	90.9 ± 0.5	94.1 ± 3.7	28.6 ± 28.6	45.8 ± 45.8	44.4 ± 44.4



(a) D_{in}: Fashion-MNIST (b) D_{in}: MNIST (c) D_{in}: SVHN (d) D_{in}: GTSRB (e) D_{in}: SVIRO-U

Fig. 3. Multiple recursive reconstructions (from left to right) of identical samples (first column) from D_{in} and D_{out} by our novel MCA-AE model. Notice the evolution in the reconstruction results over each iterative step for the OOD samples. D_{in} converge more robustly compared to D_{out} reconstructions.



(a) MCA-AE (Ours) (b) MC Dropout (c) Ensemble of 10 models

Fig. 4. Comparison of entropy histograms between D_{in} (GTSRB, filled bars with blue) and several D_{out} (not filled bars and coloured according to dataset used) for different model architectures (a), (b) and (c) . MCA-AE provides the best separation between D_{in} and D_{out} across the entire datasets.

TABLE III
WE COMPUTED THE SUM OF THE WASSERSTEIN DISTANCES BETWEEN D_{in} AND ALL D_{out} (TD ↑) SEPARATELY AND THE SUM OF THE DISTANCES BETWEEN D_{out} CIFAR10 AND ALL OTHER D_{out} (OD ↓) OVER 10 RUNS. WE REPORT MEAN AND STANDARD DEVIATION.

		MCA-AE (Ours)	MC Dropout	Ensemble
OD	↓	0.049 ± 0.007	0.080 ± 0.016	0.050 ± 0.007
TD	↑	1.551 ± 0.044	0.854 ± 0.028	0.686 ± 0.017

TABLE IV
OOD AND UNCERTAINTY ESTIMATION WHEN NO RECURSION IS APPLIED. IN MOST CASES THE RESULTS ARE WORSE COMPARED TO 2 RECURSIONS - SEE TABLE II. IN CASE THEY ARE BETTER, WE MARK THEM GREY.

$\mathcal{D}_{in} \rightarrow \mathcal{D}_{out}$	AUROC ↑	AUPR ↑	FPR95 % ↓
MNIST → MNIST	73.5 ± 1.6	91.3 ± 0.8	82.6 ± 2.7
MNIST → CIFAR10	80.4 ± 3.9	77.8 ± 4.5	58.9 ± 8.6
MNIST → Fashion	61.5 ± 5.8	59.3 ± 5.4	82.7 ± 4.2
MNIST → Omniglot	32.6 ± 10.5	44.0 ± 6.5	99.9 ± 0.1
MNIST → SVHN	87.2 ± 3.4	83.3 ± 4.9	38.8 ± 15.5
Fashion → Fashion	77.2 ± 0.9	91.6 ± 0.5	82.0 ± 1.7
Fashion → CIFAR10	88.2 ± 5.0	89.6 ± 6.2	63.0 ± 15.9
Fashion → MNIST	88.2 ± 3.3	89.3 ± 3.0	55.7 ± 8.9
Fashion → Omniglot	60.9 ± 25.4	71.3 ± 19.5	98.5 ± 2.8
Fashion → SVHN	87.2 ± 6.7	88.2 ± 8.8	61.0 ± 11.6
SVHN → SVHN	67.2 ± 1.2	54.6 ± 2.3	87.9 ± 2.1
SVHN → CIFAR10	56.0 ± 1.0	57.2 ± 1.1	94.8 ± 0.9
SVHN → GTSRB	53.6 ± 3.0	60.4 ± 2.7	94.9 ± 1.9
SVHN → LSUN	57.5 ± 1.7	59.2 ± 1.7	94.4 ± 1.4
SVHN → Places365	57.9 ± 1.3	58.6 ± 1.4	93.9 ± 1.5
GTSRB → GTSRB	85.7 ± 1.3	95.9 ± 0.6	67.3 ± 2.9
GTSRB → CIFAR10	82.2 ± 2.3	81.0 ± 2.5	69.9 ± 6.5
GTSRB → LSUN	83.2 ± 2.2	82.0 ± 2.3	68.6 ± 6.1
GTSRB → Places365	82.8 ± 2.1	81.3 ± 2.3	68.2 ± 5.9
GTSRB → SVHN	79.8 ± 2.8	78.7 ± 3.1	76.4 ± 5.5
SVIRO-U → CIFAR10	73.4 ± 2.8	60.9 ± 3.3	76.4 ± 4.7
SVIRO-U → GTSRB	70.5 ± 7.8	63.6 ± 7.2	82.4 ± 6.6
SVIRO-U → LSUN	70.8 ± 2.7	58.2 ± 2.6	81.2 ± 3.6
SVIRO-U → Places365	73.5 ± 2.9	60.4 ± 3.2	76.4 ± 4.5
SVIRO-U → SVHN	79.9 ± 3.5	66.7 ± 5.7	59.8 ± 4.6
SVIRO-U → Adults (A)	86.7 ± 2.2	98.6 ± 0.5	66.6 ± 8.8
SVIRO-U → Seats (S)	19.5 ± 13.1	1.2 ± 1.0	74.6 ± 37.6
SVIRO-U → Objects (O)	58.6 ± 4.9	56.6 ± 6.2	88.2 ± 6.2
SVIRO-U → A,S	43.4 ± 2.9	11.0 ± 2.0	95.1 ± 2.3
SVIRO-U → A,O	65.9 ± 1.8	75.1 ± 1.6	88.5 ± 1.4
SVIRO-U → A,S,O	48.4 ± 3.2	16.9 ± 2.1	91.6 ± 2.4
SVIRO-U → Tesla (OOD)	54.2 ± 10.8	86.2 ± 4.1	94.8 ± 3.7

V. DISCUSSION AND LIMITATIONS

From a mathematical point of view dynamical systems are defined by natural phenomena or mechanical systems one wants to investigate and understand. Hence, designing or influencing the dynamical system of interest is usually not a possibility. An interesting observation is that the latter phenomenon is not the case for the recursive application of an autoencoder which is then interpreted as a dynamical system. Since we train the autoencoder in the first step, the resulting dynamical behavior and its attractors can be influenced by our previously defined autoencoder training procedure. We believe that it is an interesting direction for future work to analyze this interrelationship. Further, the effect of the number of epochs needed to obtain good results should be investigated. The basins of attraction can be studied after the autoencoder model is trained, such that potentially this information could be used to further improve robustness, interpretability and uncertainty estimation. We believe that the trajectory of the latent space representation over several iterations can give hints about the model robustness. Finally, while we fix the dropout mask for one recursion and each iterative step (but using a different one for each new recursion), it would also be possible to sample a new function f for each iterative step within a recursion.

VI. CONCLUSION

Our results on several datasets show that the recursive application of autoencoder models, viewed as dynamical systems, together with an MC dropout approach provides good uncertainty and out-of-distributions estimations. Our model design choices improve the performance, particularly for computer vision datasets of higher visual complexity. Our ablation study highlights that the success is mainly due to the recursion and the entropy histograms underline the improved separability compared to MC dropout and an ensemble of models.

ACKNOWLEDGEMENT

The first author is supported by the Luxembourg National Research Fund (FNR) under grant number 13043281. The second author is supported by DECODE (01IW21001).

REFERENCES

[1] Y. Gal, R. Islam, and Z. Ghahramani, "Deep bayesian active learning with image data," in *International Conference on Machine Learning (ICML)*, 2017.

[2] C. Käding, E. Rodner, A. Freytag, and J. Denzler, "Fine-tuning deep neural networks in continuous learning scenarios," in *Asian Conference on Computer Vision (ACCV)*, 2016.

[3] A. Damianou and N. D. Lawrence, "Deep gaussian processes," in *Artificial intelligence and statistics (AISTATS)*, 2013.

[4] M. A. Kupinski, J. W. Hoppin, E. Clarkson, and H. H. Barrett, "Ideal-observer computation in medical imaging with use of markov-chain monte carlo techniques," *Journal of the Optical Society of America A: Optics and Image Science, and Vision*, 2003.

[5] C. Louizos and M. Welling, "Multiplicative normalizing flows for variational bayesian neural networks," in *International Conference on Machine Learning (ICML)*, 2017.

[6] A. Amini, W. Schwarting, A. Soleimany, and D. Rus, "Deep evidential regression," *Advances in Neural Information Processing Systems (NeurIPS*, 2020.

[7] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *International Conference on machine learning (ICML)*, 2016.

[8] A. Radhakrishnan, M. Belkin, and C. Uhler, "Overparameterized neural networks implement associative memory," *Proceedings of the National Academy of Sciences (PNAS)*, 2020.

[9] S. H. Strogatz, *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry and Engineering*. Westview Press, 2000.

[10] S. Dias Da Cruz, O. Wasenmüller, H.-P. Beise, T. Stifter, and D. Stricker, "Sviro: Synthetic vehicle interior rear seat occupancy dataset and benchmark," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2020.

[11] H. Ramsauer, B. Schäfl, J. Lehner, P. Seidl, M. Widrich, L. Gruber, M. Holzleitner, T. Adler, D. Kreil, M. K. Kopp *et al.*, "Hopfield networks is all you need," in *International Conference on Learning Representations*, 2020.

[12] D. Krotov and J. J. Hopfield, "Dense associative memory for pattern recognition," *Advances in neural information processing systems (NeurIPS*, 2016.

[13] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the national academy of sciences (PNAS)*, 1982.

[14] T. Salvatori, Y. Song, Y. Hong, L. Sha, S. Frieder, Z. Xu, R. Bogacz, and T. Lukasiewicz, "Associative memories via predictive coding," *Advances in Neural Information Processing Systems (NeurIPS*, 2021.

[15] Y. Jiang and C. Pehlevan, "Associative memory in iterated overparameterized sigmoid autoencoders," in *International Conference on Machine Learning (ICML)*, 2020.

[16] A. Radhakrishnan, K. Yang, M. Belkin, and C. Uhler, "Memorization in overparameterized autoencoders," in *Deep Phenomena Workshop, International Conference on Machine Learning (ICML)*, 2019.

[17] A. H. Hadjahmadi and M. M. Homayounpour, "Robust feature extraction and uncertainty estimation based on attractor dynamics in cyclic deep denoising autoencoders," *Neural Computing and Applications*, 2019.

[18] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya *et al.*, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *Information Fusion*, 2021.

[19] V. Böhm and U. Seljak, "Probabilistic auto-encoder," *arXiv preprint arXiv:2006.05479*, 2020.

[20] A. Grover and S. Ermon, "Uncertainty autoencoders: Learning compressed representations via variational information maximization," in *The 22nd International Conference on Artificial Intelligence and Statistics (AISTATS*, 2019.

[21] X. Ran, M. Xu, L. Mei, Q. Xu, and Q. Liu, "Detecting out-of-distribution samples via variational auto-encoder with reliable uncertainty estimation," *Neural Networks*, 2021.

[22] Z. Xiao, Q. Yan, and Y. Amit, "Likelihood regret: An out-of-distribution detection score for variational auto-encoder," *Advances in Neural Information Processing Systems*, 2020.

[23] A. M. Vartouni, S. S. Kashi, and M. Teshnehlab, "An anomaly detection method to detect web attacks using stacked auto-encoder," in *2018 6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)*, 2018.

[24] H. Xu, W. Chen, N. Zhao, Z. Li, J. Bu, Z. Li, Y. Liu, Y. Zhao, D. Pei, Y. Feng *et al.*, "Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications," in *Proceedings of the 2018 World Wide Web Conference (WWW)*, 2018.

[25] D. Y. Oh and I. D. Yun, "Residual error based anomaly detection using auto-encoder in smd machine sound," *Sensors*, 2018.

[26] M. Sensoy, L. Kaplan, and M. Kandemir, "Evidential deep learning to quantify classification uncertainty," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[27] D. J. MacKay, "Probable networks and plausible predictions-a review of practical bayesian methods for supervised neural networks," *Network: computation in neural systems*, 1995.

[28] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *International Conference on Machine Learning (ICML*, 2015.

[29] T. Chen, E. Fox, and C. Guestrin, "Stochastic gradient hamiltonian monte carlo," in *International conference on machine learning (ICML*, 2014.

[30] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer school on machine learning*. Springer, 2003, pp. 63–71.

[31] R. Noori, H.-D. Yeh, M. Abbasi, F. T. Kachoosangi, and S. Moazami, "Uncertainty analysis of support vector machine for online prediction of five-day biochemical oxygen demand," *Journal of Hydrology*, vol. 527, pp. 833–843, 2015.

[32] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[33] A. Vyas, N. Jammalamadaka, X. Zhu, D. Das, B. Kaul, and T. L. Willke, "Out-of-distribution detection using an ensemble of self supervised leave-out classifiers," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[34] M.-H. Laves, S. Ihler, K.-P. Kortmann, and T. Ortmaier, "Calibration of model uncertainty for dropout variational inference," *arXiv preprint arXiv:2006.11584*, 2020.

[35] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," *Proceedings of International Conference on Learning Representations (ICLR)*, 2017.

[36] D. Hendrycks, M. Mazeika, and T. Dietterich, "Deep anomaly detection with outlier exposure," in *International Conference on Learning Representations (ICLR)*, 2018.

[37] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd international conference on Machine learning (ICML)*, 2006.

[38] C. Manning and H. Schutze, *Foundations of statistical natural language processing*. MIT press, 1999.

[39] S. Liu, R. Garrepalli, T. Dietterich, A. Fern, and D. Hendrycks, "Open category detection with pac guarantees," in *International Conference on Machine Learning (ICML)*, 2018.

[40] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 1998.

[41] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[42] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

[43] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel, "Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark," in *International Joint Conference on Neural Networks (IJCNN)*, 2013.

[44] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, "Human-level concept learning through probabilistic program induction," *Science*, 2015.

[45] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[46] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, "Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop," *arXiv preprint arXiv:1506.03365*, 2015.

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

[48] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Advances in neural information processing systems (NeurIPS)*, 2012.
[49] P. Bergmann, S. Löwe, M. Fauser, D. Sattlegger, and C. Steger, "Improving unsupervised defect segmentation by applying structural similarity to autoencoders," *arXiv preprint arXiv:1807.02011*, 2018.