Abstract—Most of the existing artificial neural networks (ANNs) fail to learn continually due to catastrophic forgetting, while humans can do the same by maintaining previous tasks’ performances. Although storing all the previous data can alleviate the problem, it takes a large memory, infeasible in real-world utilization. We propose a continual zero-shot learning model that is more suitable in real-case scenarios to address the issue that can learn sequentially and distinguish classes the model has not seen during training. We present a hybrid network that consists of a shared V AE module to hold information of all tasks and task-specific private V AE modules for each task. The model’s size grows with each task to prevent catastrophic forgetting of task-specific skills, and it includes a replay approach to preserve shared skills. We demonstrate our hybrid model is effective on several datasets, i.e., CUB, AWA1, AWA2, and aPY. We show our method is superior to existing algorithms on class sequentially learning with ZSL (Zero-Shot Learning) and GZSL (Generalized Zero-Shot Learning). Our code is available at https://github.com/CZSLwithCV AE/CZSL_CVAE.

I. INTRODUCTION

Although ANNs achieve state-of-the-art performance on many machine learning problems like image classification, object detection, and natural language processing, the models forget the previous knowledge due to catastrophic forgetting when trained for new tasks. We still need to improve our existing algorithms to achieve human-level performance the way humans learn [1] sequentially without forgetting previous tasks throughout their life. Can we build an ANN model that can learn sequentially and simultaneously works for zero-shot learning (distinguish the classes it has not seen during training)? The name of such methods is continual zero-shot learning. All the existing algorithms have limitations; they can either learn continually or works for ZSL (distinguish data of unseen levels).

ZSL (zero-shot learning), where a trained model sees data only from unseen classes during testing, and GZSL (generalized zero-shot learning), where data come from both seen and unseen classes for prediction. Both mentioned tasks are challenging for a model to perform continually. Researchers have addressed both ZSL [2, 3] and continual learning [4, 14, 19] approaches separately. References [5, 6] combines ZSL with continual learning before, though our approach is entirely different from that, but is more realistic and beats most of their results on the same datasets.

One of the approaches for continual learning is learning a single representation with a fixed size. They learn essential weight parameters for each task and avoid alteration to learn new tasks. In contrast, structure-based approaches grow in size with each task. However, these approaches are not feasible for a large number of tasks if each needs a vast memory. Another method relies on experience replay either by storing data [7, 8] from previous tasks or synthesize data [9, 10] for old classes using generative methods. In this paper, we propose a novel adversarial training of variational autoencoders for continual zero-shot learning. Here, a disjoint space composes a task-specific latent space that is learned for each task, and a task-invariant feature space is learned for all tasks. Task-specific in a sense, the 1st private module learns from the 1st task’s real data only, whereas the $t^{th}$ private module gets trained by real data from $t^{th}$ task and replay synthesized data from all previous (t-1) tasks. We tackle both ZSL and continual learning together by using CVAE (conditional variational autoencoders) [11] that transfer knowledge from seen to unseen classes through class embeddings [12] to counter ZSL problems. As visual data is not available during training time, knowledge transfer from seen to unseen classes is formed through side information that makes semantic relationships between classes and class-embeddings. Our approach is motivated by the fact that processing and synthesizing images are time taking for continual learning when the number of classes is high. Therefore, instead of images, we train and test our model with the features of the same images generated using a pre-trained model. Another thing that motivates us is that the human brain structure is complex and contains billions of neurons [13], so we may need to eventually make complicated networks in the future containing a huge number of neurons to learn sequentially. The main contributions of this work are summarized as follows:

- We develop an experience replay-based and structure-based continual zero-shot learning method using CVAE.
- The proposed method is developed for a single head setting that is more convenient to solve real case scenarios.
- We present results for four ZSL benchmark datasets for continual zero-shot learning.
- We propose two types of latent space: one, task-invariant holds information for all tasks, another space is task-
specific. If there are T tasks, our proposed architecture consists of one task-invariant VAE and T task-specific VAEs.

II. RELATED WORK

A. Continual learning

There are three types of continual learning: regularization-based, memory-based, and structure-based methods.

Regularization methods

Here, the learning capacity is fixed [14, 15], and continual learning is performed by penalizing a network’s parameters. Researchers use a new regularizer for a new method. In [15], the essential parameters are computed online. They keep track of how the loss function changes due to a specific parameter change and accumulate this information during training. There should be a weight importance process for parameters selection to prioritize parameters usage—the way elastic weight consolidation (EWC) [14] gives importance to parameters based on the Fisher information matrix. The usages of these methods are limited because they cannot perform well for a large number of tasks.

Memory-based methods

Methods in this category try to prevent forgetting by either storing [17, 18] or synthesizing data from previous classes. The first one needs memory for rehearsal, whereas the latter is a generative model like GAN [20] or VAE [11], or both synthesize data of previous tasks to perform pseudo-rehearsal. The number of examplers stored decreases with the increase in classes if the memory budget is limited. Researchers have recently proposed using a tiny memory [17] to store a few examples per class for old tasks.

Structure-based methods

The third approach to mitigate forgetting is structure-based methods [21]. The size of a network grows with each task to prevent catastrophic forgetting. Previous tasks’ performance is maintained by freezing the learned module while accommodating new tasks by augmenting the network with new modules. The computational cost for this method is inevitable if the number of tasks is high.

B. Zero-shot learning and Generalized zero-shot learning

Recently, zero-shot learning (ZSL) has attracted a lot of attention because the model can distinguish unseen classes during testing. ZSL models are able to do so by transferring knowledge from seen to unseen levels through a semantic relationship between classes and their attributes. We can transform a ZSL problem into a supervised machine learning problem using generative models like GAN or VAE, or both. Once a generative model gets trained, it can synthesize data for unseen classes, and the data is useful for training a classifier like a conventional supervised problem. Another modification of ZSL is GZSL, a more practical approach, where data come from both seen and unseen classes.

C. Adversarial learning

Adversarial learning has usages in many domains such as generative models [20], object composition [22], representation learning [23], domain adaptation [24], active learning [25] etc. In adversarial training, a model learns the parameters through the minimax game, where a module wants to maximize the cost function, and another wants to minimize the same. This paper shows shared play the minimax game with discriminator, where shared tries to minimize the loss function, and the discriminator wants to maximize.
III. Adversarial Training of Conditional Variational Autoencoders for Continual Zero-Shot Learning

We study the problem of learning a sequence of T data distributions denoted as $D_{tr} = \{D_{tr}^1, D_{tr}^2, ..., D_{tr}^T\}$, where $D_{tr}^t = \{(X_i^t, Y_i^t, T_i^t)\}_{i=1}^{n_t}$ is the data distribution for the task t with $n_t$ sample tuples of input($X_i^t \in X$), target label($Y_i^t \in Y$), and task label($T_i^t \in T$). $D_{tr}^t$ contains seen class information. Apart from this, class embeddings of unseen classes($U_{uc} = \{(a_i^t)_{i=1}^{n_{uc}}\}$) are also available, where $n_{uc}$ is the number of unseen labels. The goal is to learn a sequential function, $f : (z \sim \mathcal{N}(0, 1), Y) \rightarrow X_s$, for each task, where $X_s$ is synthesized data generated from the shared module. The synthetic data can be used to train a supervised classifier. We aim to learn another function, $f_{\theta} : X \rightarrow \text{tar}$, after training each task, that can map input (from seen or unseen or both classes) into it’s target output without affecting the previous model’s performance on prior works. We try to achieve our goal by training two separate modules: shared and private, to enhance a better knowledge transfer from seen to unseen classes and better forget avoidance of prior knowledge. The model prevents catastrophic forgetting in shared and private spaces separately and begins learning $f_{\theta}^t$ where $\theta \in (\theta_S, \theta_P)$ as mapping function from $(X_s^t, Y_s^t)$ to $\text{tar}^t$. We use some n samples per class to be synthesized prior to $t^{th}$ task and accumulate the generated data to the current task($t^{th}$) to train the model.

$$D_{tr}^t \leftarrow D_{tr}^t \cup D_{gen}^{t:(t-1)}$$

The cross-entropy loss function for the $f_{\theta}$ mapping corresponds to:

$$L_{task} = \mathbb{E}_{(X^t, Y^t, \text{tar}^t) \sim D^t} \left[\sum_{c=1}^{C} \mathbb{I}_{c(\text{tar}^t)} \log(\sigma(f_{\theta}^t(X^t, Y^t)))\right]$$

Where $\sigma$ is the softmax function, in learning a sequence of tasks, an ideal $f_{\theta}$ maps the input features $X^t$ to two independent feature spaces: $X_s^t$ a shared features space among all tasks and $X_p^t$ remains private for each task. Both $X_s^t$ and $X_p^t$ get concatenated and fed to a task-specific multi-layer perceptron network to get desired output labels.

We introduce a mapping named shared ($S_{\theta_S} : X \rightarrow X_s$) and train it to generate features by feeding noise into the shared module’s decoder to fool a discriminator D. In contrast, the $D(S_{\theta_D} : X_s \rightarrow T)$ try to assign the synthesized features to their corresponding task labels($T \in \{0, 1, 2, ..., T\}$). The decoder and the discriminator can do so when the D gets trained to maximize the probability of assigning correct task labels to the features generated from the shared module. Simultaneously, the shared tries to minimize the same probability.

The corresponding cross-entropy adversarial loss for the minimax game:

$$L_{adv}(D, S, D_{tr}^t) = \min_D \max_S \sum_{t=0}^{T} \mathbb{I}_{t=t'} \log(D(S(X^t, Y^t)))$$

The extra-label zero is there for fake data generated from the Gaussian distribution with mean = 0 and std = 1. In most cases, we use adversarial training in a generative adversarial network that tries to learn the input data distribution in order to synthesize more data from the same distribution. Here we do the same by utilizing generative models task-invariant shared(VAE), and task-specific private(VAE); both try to learn input data distribution.

To facilitate adversarial training for S, we use the Gradient Reversal Layer[28] that directly tries to maximize the discriminator’s loss. The layer acts like an identity function during forward-propagation but multiplies the loss with a negative one during backpropagation in order to maximize the cost function for the discriminator. The adversarial training between the discriminator and the shared is complete when the discriminator can no longer predict the correct task label for features generated from the shared module. The private module, however, merely learns any task-invariant features.

Variational autoencoders

Autoencoders can effectively learn feature space and representation[15, 22]. A variational Autoencoder(VAE) is a generative model that follows an encoder-latent vector-decoder architecture of classical autoencoder, which places a prior distribution on the feature space and uses an expected lower bound to optimize the learned posterior. Conditional VAE is an extension of the VAE, where data are fed to network with class properties such as labels, attributes, etc. The VAE is a fundamental building block of our approach. Variational distribution aims to find a true conditional probability distribution over the latent variables z through minimizing their distance using a variational lower bound limit. The loss function for a VAE is:

$$L_{VAE} = \mathbb{E}_{q_{\phi}(z|x)} \left[\log(p_{\theta}(x|z)) - D_{KL}(q_{\phi}(z|x) \parallel p_{\theta}(z))\right]$$

Where the first term is the reconstruction loss, and the second one is the KL divergence between $q(z|x)$ and $p(z)$. The encoder predicts $\mu$ and $\sum$ such that $q_{\phi}(z|x) = \mathcal{N}(\mu, \sum)$, from which a latent vector is synthesized via reparametrization process.

The final objective function of the model for the $t^{th}$ task is:

$$L^{(t)} = \lambda_1 L_{adv} + \lambda_2 L_{task} + \lambda_3 L_{VAE}^{(t)} + \lambda_4 L_{VAE}$$

Where, $\lambda_1, \lambda_2, \lambda_3$, and $\lambda_4$ are regularizers to control the effect of each component. The full algorithm of the model is given in Algorithm 1.
A. Avoid forgetting

Catastrophic forgetting occurs because of the imbalance between old and new classes that results in a bias of the network towards the newest ones. One insight of our approach is to decouple the single representation learned for all tasks continually into two parts: private and shared. Though knowledge is transferred for ZSL and GZSL mostly from the shared module from seen to unseen classes. The critical approach is experience replay that gets concatenated to the current task’s data during training of the model with the same task to avoid forgetting sequentially.

B. Datasets

We evaluate our model on four benchmark datasets used for ZSL: Attribute Pascal and Yahoo(apY)[2], Animals With Attributes(AWA1, AWA2)[2], and Caltech-UCSD-Birds 200-2011(CUB)[26]. Statistics of the datasets are presented in Table I.

C. Continual Zero-shot learning(CZSL) setting

The dataset we use follows the setting used in[5]. It explains whether a class is seen or unseen is decided based on the number of tasks a model has been trained so far. If a model goes trained continually up to the tth task, the classes are assumed to be seen till the tth task, and the rest of the whole dataset’s classes are accepted unseen for the model while training.

D. evaluation matrices

We evaluate the resulting model on all previous tasks similar to[16, 18] after training for each new task. We use ACC as the average test classification accuracy across all classes for GZSL, seen classes, and unseen classes for GSL to measure our model’s performance. To measure forgetting, we calculate backward transfer, BWT that says how much learning new tasks has helped improve performance on previous tasks. We calculate forgetting measure for seen classes only.

\[ BWT = \frac{1}{T-1} \sum_{t=1}^{T-1} [R_{t,t}^{seen} - R_{t,t}^{seen}] \]  

\[ mSA = \frac{1}{T} \sum_{t=1}^{T} R_{t,t}^{seen} \]  

\[ mUA = \frac{1}{T-1} \sum_{t=1}^{T-1} R_{t,t}^{unseen} \]  

mSA is the mean seen classification accuracy across all tasks.

\[ mUA = \frac{1}{T} \sum_{t=1}^{T} R_{t,t}^{unseen} \]  

mUA is the measure of zero-shot learning performance for the model.

\[ mOA = \frac{1}{T} \sum_{t=1}^{T} R_{t,t}^{overall} \]  

mOA is the measure of generalized zero-shot learning performance.

\[ mH = \frac{1}{T-1} \sum_{t=1}^{T-1} \left[ \frac{2 \cdot R_{t,t}^{seen} \cdot R_{t,t}^{unseen}}{R_{t,t}^{seen} + R_{t,t}^{unseen}} \right] \]  

mH is the hermonic mean classification accuracy.

Algorithm 1 Continual Zero-shot Learning

Input: \((X', Y', tar) \sim D^{all})

Parameters: \(\theta_s, \theta_p, \theta_D, \theta_c\)

Output: \(X_S, X_P\)

1: \(D^{gen} \leftarrow \{\}\)
2: for \(t \leftarrow 1\) to \(T\) do
3:    for \(e \leftarrow 1\) to epochs do
4:        for \(k \leftarrow 1\) to \(S_{steps}\) do
5:            Compute \(L_{task}\) using \((X'^t, Y'^t, tar^t) \in D^t\)
6:            Compute \(L_{adv}\) using \((X'^t, Y'^t, t) \in D^t\)
7:            Compute \(L_S^{VAE}\) for shared module using \((X'^t, Y'^t) \in D^t\)
8:            Compute \(L_P^{VAE}\) for private module using \((X'^t, Y'^t) \in D^t\)
9:            \(L(t) = \lambda_1 L_{adv} + \alpha_2 L_{task} + \lambda_3 L_S^{VAE} + \lambda_4 L_P^{VAE}\)
10: \(\theta'_S \leftarrow \theta_S - \alpha_S \nabla L(t)\)
11: \(\theta'_P \leftarrow \theta_P - \alpha_P \nabla L(t)\)
12: end for
13: for \(j \leftarrow 1\) to \(D_{steps}\) do
14:    Compute \(L_{adv}\) for \(D\) using \((S(x)^t, tar^t)\) and \((z' \sim N(\mu = 0, \sum = 1), tar = 0)\)
15: \(\theta'_D \leftarrow \theta_D - \alpha_D \nabla L(t)\)
16: end for
17: end for
18: Generate data from the shared module for seen and unseen classes to train a separate classifier.
19: \(D \leftarrow D_{seen} \cup D_{unseen}\)
20: for \(C_e \leftarrow 1\) to \(C_{epochs}\) do
21:    Compute \(L_{class}\) using \((X', tar) \in D\)
22: \(\theta'_c \leftarrow \theta_c - \alpha_c \nabla L_{class}\)
23: end for
24: Test the classifier for seen data.
25: Test the classifier for unseen data(ZSL).
26: Test the classifier for all seen and unseen data(GZSL).
27: for \(c \leftarrow 1\) to \(C\) do
28:    \(C\) is the replay classes.
29:    for \(i \leftarrow 1\) to \(n\) do
30:        \(n\) is the number of samples to be generated per class for the experience replay.
31: \((X_t, Y_t, tar_t) \sim D^{gen}\)
32: end for
33: end for
34: \(D^{t+1} \leftarrow D^{t+1} \cup D^{gen}\)
35: end for

Where \(R_{t,i}\) is the test classification accuracy on task \(i\) after sequentially finishing learning the \(jth\) task.
| Dataset | Semantic Dim | #Images | #SC | #UC |
|---------|--------------|---------|-----|-----|
| CUB     | 312          | 11788   | 150 | 50  |
| aPY     | 64           | 15339   | 20  | 12  |
| AWA1    | 85           | 30475   | 40  | 10  |
| AWA2    | 85           | 37322   | 40  | 10  |

**TABLE I:** Datasets and their statistics, Where SC and UC are seen and unseen classes respectively.

| Methods                  | mSA | mUA(ZSL) | mH  | mOA(GZSL) |
|--------------------------|-----|----------|-----|-----------|
| AGEM+CZSL                | -   | -        | -   | -         |
| Seq-CVAE                 | 24.66 | 8.57    | 12.18 | -         |
| CZSL-CV+mof             | 43.73 | 10.26   | 16.34 | -         |
| CZSL-CV+rb              | 42.97 | 13.07   | 19.53 | -         |
| CZSL-CV+res             | 44.89 | 13.45   | 20.15 | -         |
| ours(without adv)       | 34.25 | 12.42   | 17.41 | 22.40     |
| ours(with adv)           | 34.47 | 12.00   | 17.15 | 21.72     |

**TABLE II:** Results for the CUB dataset, where mSA: Mean Seen Accuracy, mUA: Mean Unseen Accuracy, mH: Hermonic Mean Accuracy, mOA = Mean Overall Accuracy. The best results in the table are presented in boldface.

| Methods                  | mSA | mUA(ZSL) | mH  | mOA(GZSL) |
|--------------------------|-----|----------|-----|-----------|
| AGEM+CZSL                | -   | -        | -   | -         |
| Seq-CVAE                 | 51.57 | 11.38   | 18.33 | -         |
| CZSL-CV+mof             | 64.91 | 10.79   | 18.27 | -         |
| CZSL-CV+rb              | 64.45 | 11.00   | 18.60 | -         |
| CZSL-CV+res             | 64.88 | 15.24   | 23.90 | -         |
| ours(without adv)       | 58.14 | 15.91   | 23.05 | 38.20     |
| ours(with adv)           | 55.46 | 11.2    | 18.63 | 35.97     |

**TABLE III:** Results for the aPY dataset, where mSA: Mean Seen Accuracy, mUA: Mean Unseen Accuracy, mH: Hermonic Mean Accuracy, mOA = Mean Overall Accuracy. The best results in the table are presented in boldface.

IV. EXPERIMENTS

In this section, we discuss baselines, training, and results.

**A. Baselines**

The research on continual zero-shot learning (CZSL) has been less explored. References[5, 6] has investigated the work before on a single-head setting that we represent in this paper. Reference [6] used the following baselines, so we do the same in this paper.

- **AGEM + CZSL[5]:** It is an average gradient episodic memory-based continual zero-shot learning. The authors of [5] have mentioned the harmonic mean of the CUB dataset only.
- **SEQ + CVAE[6]:** The authors train CVAE sequentially without considering any continual learning strategy. After

**B. Other Methods**

We also compare our results with CZSL-CV+mof[6], CZSL-CV+rb[6], and CZSL-CV+res[6].

**C. Training**

We use Pytorch as our framework. We train our model with a hundred epochs and a classifier for thirty epochs for each task on all datasets except CUB. We use the same number of epochs for the CUB dataset till task fifteen and then reduce to fifty for the model and ten for the classifier till task eighteen and again decrease to twenty for the model and five for the classifier. The Adam[27] optimizer is used in all experiments, and the learning rate for the classifier and others is 0.0001 and 0.001, respectively. We use weight decay 0.0001 as a regularizer for the classifier. We use 500 hidden units for both shared and private modules. Latent dimension is 50, and batch size for both model and classifier is 64. We take $\lambda_1 = \lambda_2 = \lambda_3 = 1$, and $\lambda_4 = 0.5$. 

SEQ+CVAE is trained on the current task, synthetic are generated using noise and class embeddings for all classes to train a separate classifier.
TABLE IV: Results for the AWA1 dataset, where mSA: Mean Seen Accuracy, mUA: Mean Unseen Accuracy, mH: Harmonic Mean Accuracy, mOA = Mean Overall Accuracy. The best results in the table are presented in boldface.

| Methods            | mSA  | mUA (ZSL) | mH  | mOA (GZSL) |
|--------------------|------|-----------|-----|------------|
| AGEM+CZSL          | -    | -         | -   | -          |
| Seq-CVAE           | 59.27| 18.24     | 27.14| -          |
| CZSL-CV+mof        | 79.77| 20.12     | 30.46| -          |
| CZSL-CV+rb         | 77.85| 21.90     | 33.64| -          |
| CZSL-CV+res        | 78.56| 23.65     | 35.51| -          |
| ours(without adv)  | 67.98| 20.58     | 30.83| 43.08      |
| ours(with adv)     | 71.00| 24.26     | 35.75| 50.9       |

TABLE V: Results for the AWA2 dataset, where mSA: Mean Seen Accuracy, mUA: Mean Unseen Accuracy, mH: Harmonic Mean Accuracy, mOA = Mean Overall Accuracy. The best results in the table are presented in boldface.

| Methods            | mSA  | mUA (ZSL) | mH  | mOA (GZSL) |
|--------------------|------|-----------|-----|------------|
| AGEM+CZSL          | -    | -         | -   | -          |
| Seq-CVAE           | 61.42| 19.34     | 28.67| -          |
| CZSL-CV+mof        | 79.11| 24.41     | 36.60| -          |
| CZSL-CV+rb         | 80.92| 24.82     | 37.32| -          |
| CZSL-CV+res        | 80.97| 25.75     | 38.34| -          |
| ours(without adv)  | 70.05| 22.85     | 32.98| 44.97      |
| ours(with adv)     | 70.16| 25.93     | 37.19| 51.55      |

Fig. 7: Results without adversarial training for the AWA1 dataset.

Fig. 8: Results with adversarial training for the AWA1 dataset.

Fig. 9: Results without adversarial training for the AWA2 dataset.

Fig. 10: Results with adversarial training for the AWA2 dataset.

V. CONCLUSION

In this work, we proposed a novel hybrid algorithm. The novelty of our work is that we use adversarial learning. Here the model needs experience replay and grows for task incremental learning. The private module barely shares knowledge from seen to unseen classes that can be future work to optimize the private module for ZSL. Another future work might be to develop task-free continual zero-shot learning. How can we build a continual zero-shot learning model for object detection? What should be the optimum latent dimension, hidden-layer size?
VI. REFERENCES

[1] W. C. Abraham and A. Robins. Memory retention—the synaptic stability versus plasticity dilemma. Trends in neurosciences, 28(2):73–78, 2005.

[2] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In 2009 IEEE Conference on Computer Vision and Pattern Recognition, pages 1778–1785. IEEE, 2009.

[3] Rafael Felix, Vijay BG Kumar, Ian Reid, and Gustavo Carneiro. Multi-modal cycle-consistent generalized zeroshot learning. In ECCV, pages 21–37, 2018.

[4] Tyler L Hayes, Nathan D Cahill, and Christopher Kanan. Memory efficient experience replay for streaming learning. In 2019 International Conference on Robotics and Automation (ICRA), pages 9769–9776. IEEE, 2019.

[5] Ivan Skorokhodov and Mohamed Elhoseiny. Normalization matters in zero-shot learning. arXiv preprint arXiv:2006.11328, 2020.

[6] Gautam C, Parameswaran S, Mishra A, Sundaram S. Generalized Continual Zero-Shot Learning. arXiv preprint arXiv:2011.08508. 2020 Nov 17.

[7] Tyler L Hayes, Nathan D Cahill, and Christopher Kanan. Memory efficient experience replay for streaming learning. In 2019 International Conference on Robotics and Automation (ICRA), pages 9769–9776. IEEE, 2019.

[8] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thaialaysingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc’Aurelio Ranzato. On tiny episodic memories in continual learning. arXiv preprint arXiv:1902.10486, 2019.

[9] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In Advances in Neural Information Processing Systems, pages 2990–2999, 2017.

[10] Xiaolei Liu, Chenshen Wu, Mikel Menta, Luis Herranz, Bogdan Raducanu, Andrew D Bagdanov, Shangling Jui, and Joost van de Weijer. Generative feature replay for classincremental learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 226–227, 2020.

[11] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013.

[12] Vinay Kumar Verma and Piyush Rai. A simple exponential family framework for zero-shot learning. In Joint European conference on machine learning and knowledge discovery in databases, pages 792–808. Springer, 2017.

[13] Herculano-Houzel S. The human brain in numbers: a linearly scaled-up primate brain. Front Hum Neurosci. 2009 Nov 9;3:31. doi: 10.3389/neuro.09.031.2009. PMID: 19915731; PMCID: PMC2776484, 2009.

[14] Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences p. 201611835, 2017.

[15] Zenke, F., Poole, B., Ganguli, S.: Continual learning through synaptic intelligence. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 3987-3995. PMLR, 2017.

[16] Ebrahimi, S., Elhoseiny, M., Darrell, T., Rohrbach, M.: Uncertainty-guided continual learning with bayesian neural networks. In: International Conference on Learning Representations, 2020.

[17] Chaudhry, A., Ranzato, M., Rohrbach, M., Elhoseiny, M.: Efficient lifelong learning with A-GEM. In: International Conference on Learning Representations, 2019.

[18] D Lopez-Paz, MA Ranzato. Gradient episodic memory for continual learning. In: Advances in Neural Information Processing Systems. pp. 6467-6476, 2017.

[19] Riemer, M., Cases, I., Ajemian, R., Liu, M., Rish, L., Tu, Y., , Tesauro, G.: Learning to learn without forgetting by maximizing transfer and minimizing interference. In: International Conference on Learning Representations, 2019.

[20] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, DavidWarde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Advances in Neural Information Processing Systems, pages 2672–2680, 2014.

[21] Rusu, A.A., Rabinowitz, N.C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., Hadsell, R.: Progressive neural networks. arXiv preprint arXiv:1606.04671, 2016.

[22] Azadi, S., Pathak, D., Ebrahimi, S., Darrell, T.: Compositional gan: Learning image-conditional binary composition. International Journal of Computer Vision pp. 1-16, 2020.

[23] Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., Frey, B.: Adversarial autoencoders. arXiv preprint arXiv:1511.05644, 2015.

[24] Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7167-7176, 2017.

[25] Sinha, S., Ebrahimi, S., Darrell, T.: Variational adversarial active learning. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5972-5981, 2019.

[26] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset, 2011.

[27] Kingma, Diederik P and Ba, Jimmy Lei. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[28] Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. The Journal of Machine Learning Research 17(1), 2096-2030, 2016.