IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 28 January 2024, accepted 13 March 2024, date of publication 18 March 2024, date of current version 1 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3379148

RESEARCH ARTICLE

Looking Through the Past: Better Knowledge Retention for Generative Replay in Continual Learning

VALERIYA KHAN[1,2], SEBASTIAN CYGERT[1,3], KAMIL DEJA[1,2], TOMASZ TRZCINSKI[1,2,4,5], (Senior Member, IEEE), AND BARTLOMIEJ TWARDOWSKI[1,6,7]

[1]IDEAS NCBR, 00-801 Warsaw, Poland
[2]Faculty of Electronics and Information Technology Science, Warsaw University of Technology, 00-661 Warsaw, Poland
[3]Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, 80-233 Gdańsk, Poland
[4]Jagiellonian University, 31-007 Kraków, Poland
[5]Tooploox, 00-372 Warsaw, Poland
[6]Computer Vision Center (CVC), 08193 Barcelona, Spain
[7]Department of Computer Science, Universitat Autònoma de Barcelona, 08193 Barcelona, Spain

Corresponding author: Valeriya Khan (valeriya.khan@ideas-ncbr.pl)

This work was supported in part by the National Centre of Science, Poland, under Grant 2020/39/B/ST6/01511, Grant 2021/43/O/ST6/02482, and Grant 2022/45/B/ST6/02817; in part by PL-Grid Infrastructure under Grant PLG/2023/016393, and in part by the Warsaw University of Technology Within the Excellence Initiative: Research University (IDUB) Programme.

ABSTRACT In this work, we improve the generative replay in a continual learning setting to perform well on challenging scenarios. Because of the growing complexity of continual learning tasks, it is becoming more popular, to apply the generative replay technique in the feature space instead of image space. Nevertheless, such an approach does not come without limitations. In particular, we notice the degradation of the continually trained model's performance could be attributed to the fact that the generated features are far from the original ones when mapped to the latent space. Therefore, we propose three modifications that mitigate these issues. More specifically, we incorporate the distillation in latent space between the current and previous models to reduce feature drift. Additionally, a latent matching for the reconstruction and original data is proposed to improve generated features alignment. Further, based on the observation that the reconstructions are better for preserving knowledge, we add the cycling of generations through the previously trained model to make them closer to the original data. Our method outperforms other generative replay methods in various scenarios. Code available at https://github.com/valeriya-khan/looking-through-the-past.

INDEX TERMS Continual learning, generative replay, machine learning.

I. INTRODUCTION

The traditional approach to machine learning involves training models on shuffled training data to ensure independent and identically distributed conditions, enabling the model to learn generalized parameters for the entire data distribution. On the other hand, in continual learning, the models are trained on sequential tasks, with only data from the current task available at any given time. Such a scenario is more realistic in some applications with, for example, privacy concerns, where the old data may become unavailable. However, models trained in such an incremental fashion will face a catastrophic forgetting [24], a significant drop in the accuracy of previously acquired knowledge.

Class Incremental Learning (CIL) is a widely adopted setting where the classifier is trained on new classes incrementally using the sequence of separated data [22]. Different regularization methods can be used to preserve the knowledge [16], [37], however, the performance is significantly lower without utilizing exemplars from the previous tasks. Therefore, generative models [5] have gained

The associate editor coordinating the review of this manuscript and approving it for publication was Wenbing Zhao.

a significant attention as the source of synthetic data that can substitute data from the previous tasks.

Despite the promising setup, it turns out to be very challenging to scale approaches based on generative models in CIL to more demanding datasets than MNIST or CIFAR-10 [32]. Generative replay methods have low performance on datasets with a larger number of classes or with more complex data. This can be attributed to the fact that modeling high-dimensional images is a challenging task during the incremental learning, and the quality of the generations degrades as number of learned tasks increases.

Therefore, more recent methods [18] introduced replay in feature-space of the trained and frozen feature extractor. The data is firstly passed through the feature-extractor, and the resulting features are used as the training data for the generative part. In this case, the distribution of the data has lower dimensionality and is much simpler for learning by the generator.

Brain-Inspired Replay (BIR) [33] is one of the recent works that uses feature-based generative replay. In their work, the authors introduce several modifications to make variational autoencoder (VAE) able to learn and generate longer sequences of more complex data. The highest results reported by the authors are when BIR is combined with Synaptic Intelligence (SI) [37] regularization method, which suggests that BIR alone for a generative features-replay is not enough and maybe other regularization techniques can yield better results. It motivates us to analyze an in-depth VAE-based replay approaches with BIR as its flagship example. We observe, that there remains a significant difference between the features produced by the real data and synthetic data. Our hypothesis is that this difference leads to a significant degradation of the quality of the replay data, and therefore, we propose two modifications that diminish the problem. Firstly, we introduce a new loss term for minimizing the difference between the encoded latent vectors of the original sample and the reconstructed sample. This loss enables the encoder to learn how to reverse the operation of the decoder. Secondly, we propose to refine the quality of rehearsal samples. To that end, we introduce a cycling method where we iterate the generated data through the previously trained model (decoder and encoder), and only after that feed it to the replay buffer for training the new model. As we show in our analysis, this has the effect of reducing a discrepancy between original and generated features for classification (see Figure 1), and as a result, improves the final model accuracy. The proposed changes allowed us to significantly improve the results over our baseline method.

Overall, the contribution of this study is threefold:

- Based on the analysis of existing generative replay methods, we identify the weaknesses of VAE-based approaches such as degradation of generated data and distribution mismatch between the features obtained by original and synthetic data.
- To mitigate the discovered problems, we propose a new generative replay method for class-incremental learning. Our method uses distillation to better match latent vectors of reconstructed and original data. Also, we match the latent representations of current data obtained through previous and current models. Furthermore, we incorporate the cycling of generations to diminish the difference between the original and synthetic data.
- We perform a series of experiments to show that our approach outperforms the baseline method (BIR). In addition, we demonstrate through an ablation study that each improvement we introduce makes an incremental contribution to the overall performance of the model.

II. RELATED WORKS

Continual learning methods can be divided into three categories that we overview in this section.

Regularization methods aim to strike a balance between preserving previously acquired knowledge and providing sufficient flexibility to incorporate new information. To that end, regularisation is applied to slow down the updates on the most important weights. In particular, in Elastic Weights Consolidation (EWC) [13] authors propose to use Fisher Information to select important model's weights, while in Synaptic Intelligence (SI) [37] and Memory Aware Synapses (MAS) [1] additional information is stored together with each parameter. Similarly, in Learning Without Forgetting (LWF) [17] additional distillation loss on current data is used to match the output of the model trained on the previous task, with a new one. In this work, we use distillation techniques to align representations of old and new features similarly to LWF.

Dynamic architecture methods create different versions of the base model for each task. This is usually implemented by creating additional task-specific submodules [29], [35], [36], or by selecting different parts of the base network [4], [20], [21], [23]. Such approaches reduce catastrophic forgetting at the expense of expanding memory requirements.

Rehearsal methods involve storing and replaying past data to prevent catastrophic forgetting. The simplest implementation of this approach employs a memory buffer where a subset of examples from previous tasks can be stored [2], [3], [9], [19], [27]. Such an approach achieves high performance and can significantly reduce catastrophic forgetting.

However, the memory buffer has to store a significant number of examples and, hence, grow with each task. Also in some domains, due to privacy concerns, using historical data is not possible. Therefore, generative models are often used to synthesize past data. The first example of ***generative replay*** for CIL model is [32] where a generative model (e.g., Generative Adversarial Network (GAN) [5]) is used as a source of rehearsal examples. This idea is further extended to other generative methods such as Variational Autoencoders in [12], [34], and [25] or Normalising Flows [28] in [31]. In [15], the authors overview the general performance of generative models as a source of rehearsal examples, showing that even though GANs outperform other solutions, all the methods struggle when evaluated on more complex benchmark scenarios. Therefore, to simplify the problem,

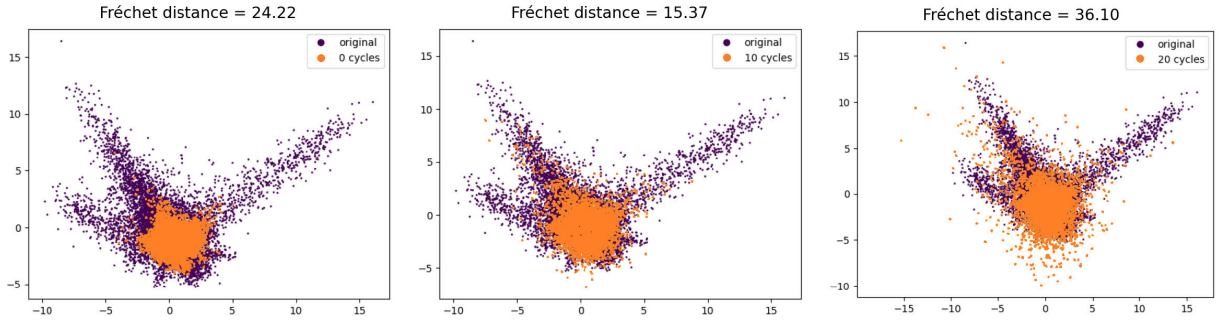


FIGURE 1. Principal Component Analysis (PCA) is performed on the original latent representations and the generated ones after 0, 10, and 20 passes during cycling. The PCA visualizations and Fréchet distances both imply that the cycling procedure decreases the discrepancy between original and generated data when the appropriate number of passes is performed [11].

in Brain-Inspired Replay (BIR) [33] the authors introduce a new idea known as *feature replay* and propose to focus on the replay of internal data representations instead of the original samples. This idea was further explored in [10], with a split between short and long-term memory, and in [18] where authors employ conditional GANs. Our method falls in the generative-feature replay category, as we directly base our approach on the BIR method. This work is an extension of workshop paper originally presented at ICCV [11]. In this version, we add experiments on mini-ImageNet dataset, and detailed evaluation of the quality of rehearsal examples with precision and recall analysis. The results of these experiments are presented in Table 2 and Figure 6. In addition, we present Algorithm1, Figures 2, 3 and 4 for better comprehension of the method.

III. METHOD

A. PROBLEM DEFINITION

This study addresses image classification within a class-incremental setting. We train the model on a sequence of n tasks: $T_1, T_2, \ldots, T_n$ where each task t consists of $\{X^{(t)}, Y^{(t)}\}$ drawn from the distribution $\mathcal{D}^{(t)}$, where X is a set of training samples, Y is a set of corresponding class labels, and $1 \leq t \leq n$. During the training of task t the model has noo access to previous tasks data.

In class-incremental learning, the model has to be trained to predict the labels for all the tasks seen so far.

B. BASELINE MODEL

Brain-Inspired Replay (BIR) method [33] serves as a baseline for our work. The model consists of feature extractor and VAE on top of it that plays a role of the feature generator. The generator part is utilized to create the synthetic data for the replay of old knowledge. It has encoder part q_ϕ and the decoder part p_ψ. The goal of the encoder is to map the sample x to probabilistic latent variable z, and the goal of the decoder is map the latent variable z to reconstruction $\hat{z}$. Typically, the objective of training VAE is to maximize the a variational lower bound on the evidence (ELBO), or alternatively we try to minimize the per-sample loss:

$$\begin{aligned} L^G(x;\phi,\psi) &= E_{z\sim q_\phi(.|x)}[-\log p_\psi(x|z)] \\ &\quad + D_{KL}(q_\phi(.|x)||p(.)) \\ &= L^{recon}(x;\phi,\psi) + L^{latent}(x;\phi), \end{aligned} \tag{1}$$

where $q_\phi(.|x) = \mathcal{N}(\mu^{(x)}, \sigma^{(x)^2} I)$ is the posterior and $p(.) = \mathcal{N}(0, I)$ is prior over the latent variables, and D_{KL} is the Kullback-Leibler divergence.

For prior distribution equal to $N(0, I)$, the KL divergence can be calculated as follows:

$$L^{latent}(x;\phi) = \frac{1}{2}\sum_{j=1}^{D}(1 + \log(\sigma_j^{(x)^2}) - \mu_j^{(x)^2} - \sigma_j^{(x)^2}), \tag{2}$$

where D is a latent dimension. The reconstruction loss in this work is given by:

$$\begin{aligned} L^{recon}(x;\phi,\psi) = E_{\epsilon\sim\mathcal{N}(0,I)}\Bigg[\sum_{p=1}^{N} x_p \log(\hat{x}_p) \\ + (1 - x_p)\log(1 - \hat{x}_p)\Bigg], \end{aligned} \tag{3}$$

where N is the size of the input, x_p is the p^{th} entry of the original input x, and $\hat{x}_p$ is the p^{th} entry of reconstruction $\hat{x}$.

In order to generate samples from specifically chosen classes, the prior can be changed from the standard normal distribution to the Gaussian mixture with each class modeled as a separate distribution:

$$p_{\mathcal{X}}(.) = \sum_{c=1}^{N_{\text{classes}}} p(\mathcal{Y} = c) p_{\mathcal{X}}(.|c), \tag{4}$$

where $p_{\mathcal{X}}(.|c) = \mathcal{N}(\mu^c, \sigma^c I)$ for $c = 1, \ldots, N_{\text{classes}}$, μ^c and σ^c are trainable means and standard deviation for class c, $\mathcal{X}$ is a set of means and standard deviations for all classes N_{classes} and $p(\mathcal{Y} = c)$ is the class prior.

For the current task with hard targets (labels), the L^{latent} has the following form:

$$\begin{aligned} L^{latent}(x,y;\phi,\mathcal{X}) = \frac{1}{2}\sum_{j=1}^{D}\Bigg(1 + \log(\sigma_j^{(x)^2}) \\ - \log(\sigma_j^{y^2}) - \frac{(\mu_j^{(x)} - \mu_j^y)^2 + \sigma_j^{(x)^2}}{\sigma_j^{y^2}}\Bigg), \end{aligned} \tag{5}$$

where μ_j^y is the j^{th} element of μ^y and σ_j^y is the j^{th} element of σ^y. For the replay, this loss is estimated for

soft-target $\tilde{y}$ as:

$$L^{latent}(x, y; \phi, \mathcal{X}) = \frac{1}{2}\sum_{j=1}^{D}\left(1 + \log(2\pi) + \log(\sigma_j^{(x)^2})\right) + E_{\epsilon\sim\mathcal{N}(0,I)}\left[\log\left(\sum_{j=1}^{D}\tilde{y}_j\mathcal{N}(\mu^{(x)} + \sigma^{(x)} \odot \epsilon|\mu^j, \sigma^{j^2}I)\right)\right], \tag{6}$$

where $\tilde{y}_j$ is the j^{th} entry of $\tilde{y}$, and estimation of expectation is performed by a single Monte Carlo sample for each input.

Classification loss is calculated for the current task as following:

$$L^C(x, y; \theta) = -\log p_\theta(\mathcal{Y} = y|x), \tag{7}$$

where p_θ is the conditional probability distribution defined by the model parameters.

In the replay part of BIR method classification loss is substituted by the distillation loss. Typically, the objective of knowledge distillation is to transfer knowledge from the teacher model to student model. Knowledge distillation is performed by minimizing the distance between the resulting vectors of the softmax function in teacher and student models. One of the problem of this approach is that the predicted probability of the true class is usually close to 1. Hence, the probability vector is close to the one-hot ground-truth label vector, and does not provide additional information. To mitigate this problem, the *softmax with temperature* is incorporated [8]. The distillation loss is calculated by:

$$L^D(x, \tilde{y}; \theta) = -T^2\sum_{c=1}^{N_{\text{classes}}}\tilde{y}_c \log p_\theta^T(\mathcal{Y} = x|x), \tag{8}$$

where T is the softmax temperature.

C. IMPROVED FEATURE REPLAY

This section describes our proposed modifications to the BIR method that serves as the baseline. These changes are aimed to mitigate the problems with VAE-based feature replay: (1) misalignment between original and reconstructed data, (2) latent drift due to continual learning training, (3) high difference between generations and original samples.

1) LATENT MATCHING FOR RECONSTRUCTIONS AND ORIGINAL DATA

The first modification we propose aims to improve VAE model performance in continual retraining. To that end, we propose a latent matching regularization that enforces encoder to reverse the decoding operation performed by the decoder. More specifically, we pass the sample x through the encoder model to get the latent representation z_o. After that, we reconstruct the original sample by passing this latent vector through the decoder and obtain $\hat{x}$. Then, the reconstruction is passed through the encoder model, and the latent representation z_r is received.

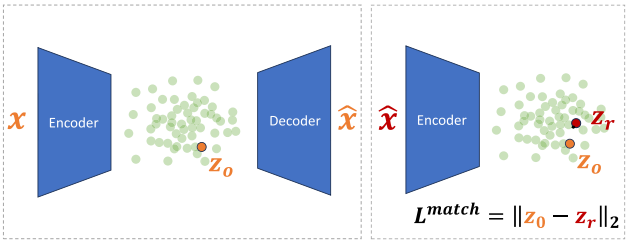


FIGURE 2. Visualisation of the latent matching loss. We minimize the difference between latent vectors of the original samples and their reconstructions.

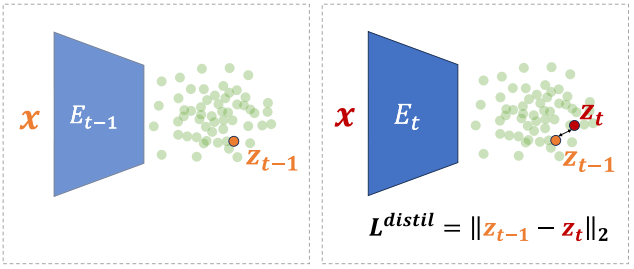


FIGURE 3. Visualisation of the latent distillation loss that reduces the feature drift between tasks.

In particular, we calculate the regularisation on mean and variations outputted by the encoder. To that end, we utilize the mean squared error (MSE) loss for measuring the difference between obtained latent representations. Therefore, we introduced latent match loss which is defined as the following:

$$L^{\text{latent match}}(z_o; \phi, \psi) = \frac{1}{2}(z_r - z_o)^2 \tag{9}$$

The visualisation of our latent match loss is presented in Figure 2.

2) LATENT DISTILLATION

As mentioned in Section III-B, the BIR method does not have any mechanism for prevention of feature drift, i.e. the distribution change in feature space during training on new data. To prevent that, we add a latent distillation loss which is performed similarly as in [18]. In order to calculate the loss during the task t, we pass the sample through the previous model encoder E_{t-1} and current model encoder E_t, and obtain latent representations z_{t-1} and z_t respectively. The latent distillation loss is calculated as the MSE between the latent representations of previous and current model, and is calculated by:

$$L^{\text{latent distill}}(z_{t-1}; \phi_{t-1,t}) = \frac{1}{2}(z_t - z_{t-1})^2 \tag{10}$$

The latent distillation loss serves as the purpose of the regularization term that controls forgetting, similarly to the SI regularization in the BIR method. Nevertheless our latent distillation achieves better performance. Figure 3 presents latent distillation loss.

3) CYCLING

Our hypothesis is that even with the first two added modifications, there is still a large discrepancy between the

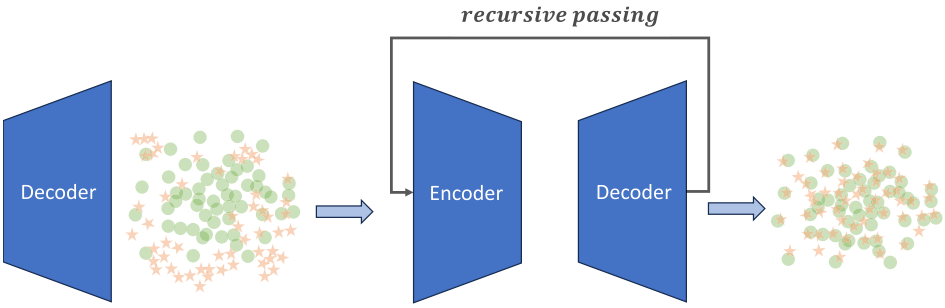


FIGURE 4. Visualisation of the cycling procedure. Each time we generate a batch of rehearsal samples (orange stars), we pass the generated outputs several times through the Variational Autoencoder in the recursive passing procedure. As a consequence, the final generations exhibit a considerably improved alignment with the reconstructions of the original training data (green dots).

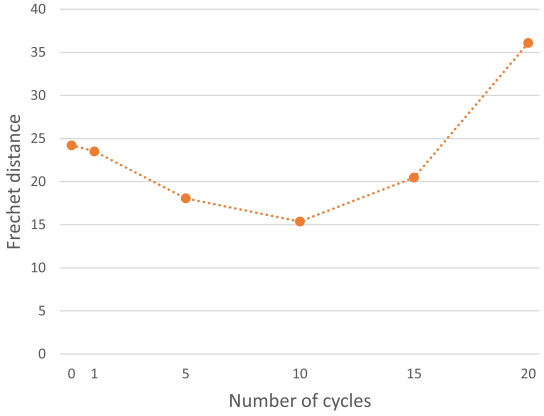


FIGURE 5. Fréchet distance between the distributions of original and generated latent vectors depending on the number of cycles. Zero cycles mean the model without cycling procedure. As the number of cycles increases (till some point), the distribution of generated representations better aligns with the original one.

generated and original features. To minimise this effect, we propose a cycling mechanism that is inspired by the idea presented by Gopalakrishnan et al. in [6]. In this work, authors propose to recursively pass images from the buffer through the pre-trained autonecoder in order to better align them to the data from a new task. Here, we use the similar mechanism with our Variational Autoencoder to align generations of data from the previous task with data reconstructions.

The visualisation of our cycling mechanism is presented in Figure 4.

In order to check the hypothesis, we calculate the Fréchet distance [7], which is used to measure the similarity of two Gaussian distributions. Typically, it is utilized to estimate the quality of generated images (known as Fréchet inception distance). In this case, we use it to measure the quality of the generated latent representations. Figure 5 presents the decrease in the Fréchet distance between the distributions of generated and original latent vectors with the increase of passes through the previous model. Therefore, we add it to the training procedure.

Empirical evaluation of the cycling and number of used rounds is presented with other experiments in Section V-B.

D. FINAL TRAINING OBJECTIVE

To summarize, we present our modified VAE-based replay method with all the improvements incorporated into the training routine via a single objective for class-incremental setting. This objective can be divided into two main parts: L^{current} and L^{replay}. Current task loss L^{current} is calculated as follows:

$$L^{\text{current}} = L^{G} + L^{C} + L^{\text{latent match}} \tag{11}$$

Replay loss L^{replay} for the previous tasks is given by:

$$L^{\text{replay}} = L^{G} + L^{D} + L^{\text{latent distill}} \tag{12}$$

Finally, the total objective is calculated as summation of these two losses:

$$L^{\text{total}} = L^{\text{current}} + L^{\text{replay}} \tag{13}$$

The final loss is utilized for training of the VAE and classifier using the current task data and the generative replay data passed through the previous model the defined number of times. The resulting loss is a combination of components without any coefficients to balance the off. That can be further investigated. The ablation study is provided in Section V-C. The steps of the overall training procedure can be found in the Algorithm 1.

IV. EXPERIMENTAL SETUP

A. DATASET

We evaluate the models on two commonly used benchmarks that are challenging for the generative replay setup CIFAR-100 dataset [14] and mini-ImageNet. CIFAR-100 consists of 100 object classes in 45,000 images for training, 5,000 for validation, and 10,000 for test. All images are in the size of 32 × 32 pixels. The mini-ImageNet contains 50,000 training images, and 10,000 testing images evenly distributed across 100 classes. All images have the size 84 × 84.

B. IMPLEMENTATION DETAILS

As a framework for our experiment, we use PyTorch [26]. We use ResNet-32 model for feature extraction. We pretrain feature extractor on the 50 classes contained in the first task, and freeze it afterwards. The same procedure is used for mini-ImageNet with the substitution of ResNet-32 to

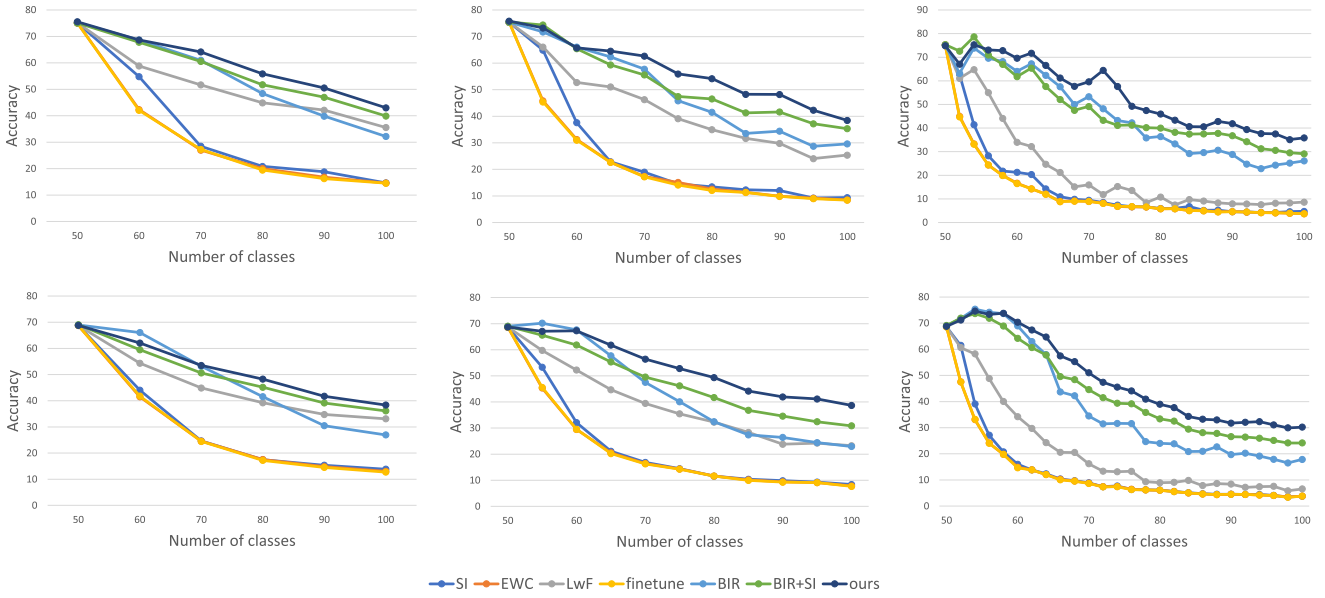


FIGURE 6. Comparison of average accuracies on CIFAR-100 (top) and mini-ImageNet (bottom) after each task for 6, 11, and 26 tasks (from left to right) with the first task containing 50 classes.

Algorithm 1 Class-Incremental Learning With Improved Generative Feature Replay

Input: Data $D_1, D_2, \ldots, D_T$, where $D_t = \{F(X_t), Y_t\}$, where F is a pretrained feature extractor
Require: Initialized encoder Enc_0, initialized decoder Dec_0, initialized classifier θ_0, number of cycles N_{cycles}
for $t = 1, \ldots, T$ **do**
  if $t = 1$ **then**
    Step 1: Train Enc_{new}, Dec_{new} and θ on data D_1 by minimizing L^{current}
  else
    Step 2: Save previously trained generator $Dec_{old} = Dec_{new}$, $Enc_{old} = Enc_{new}$
    Step 3: Generate data $\hat{D}_{1:t-1} = Dec_{old}(y_{t'}, z)$, where $y_{t'}$ is all classes seen-so-far
    Step 4:
    for $k < N_{cycles}$ **do**
      $\hat{D}_{1:t-1} = Dec_{old}(Enc_{old}(\hat{D}_{1:t-1}))$
    end for
    Step 5: Train Enc_{new}, Dec_{new} and θ on current data D_t by minimizing L^{current} and on generated data $\hat{D}_{t-1}$ by minimizing L^{replay}
  end if
end for

ResNet18. During the pretraining, we utilize the strong data augmentations from the PyCIL framework [38] to improve the feature extraction model. During the class-incremental training of generator and classifier, we use weaker data augmentations to minimize the distortions to the original data. More specifically, we firstly pad images by 4 with 0 values, and after that we crop the image at random location to the size 32 × 32 for CIFAR-100 and 84×84 for mini-ImageNet. Lastly, random horizontal flips are applied. We train the encoder part on top of the feature extractor for 10000 iterations for the first task and for 5000 iterations for the rest of the tasks. Adam optimizer from PyTorch [26] is used for the experiments with the learning rate equal to 1e-4.

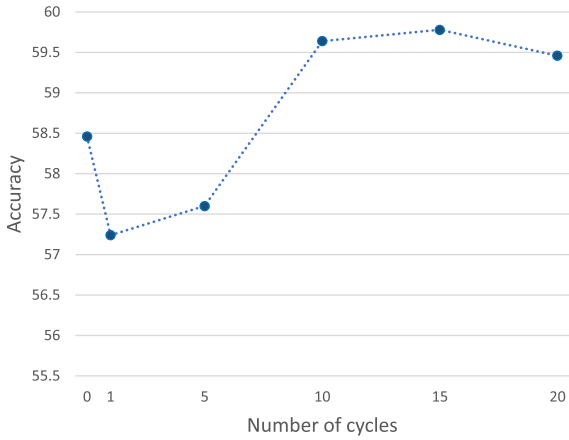


FIGURE 7. Average incremental accuracy as a function of a number of cycles for $T = 6$.

C. EVALUATION

For evaluation, we use the average overall accuracy metric as in [33]. It is the average accuracy of the model on the test data of all tasks up to the current one. In addition, to evaluate the overall performance, we calculate average incremental accuracy over all tasks. It is obtained by taking the average of accuracies after each task. Each experiment is performed over 3 random seeds and the mean is reported.

V. RESULT AND ANALYSIS

A. MAIN RESULTS

For the experiments on CIFAR-100 and mini-ImageNet, 50 classes are contained in the first task following [33],

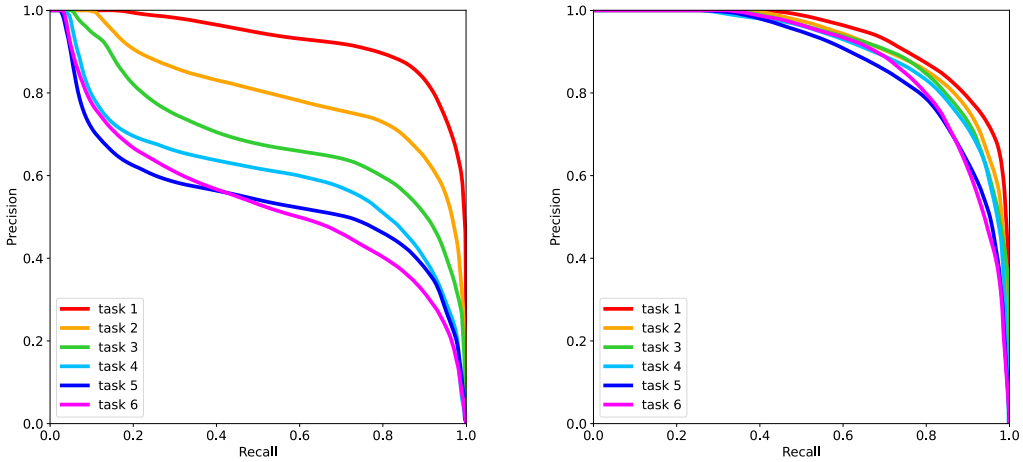


FIGURE 8. Comparison of the Precision/Recall curves for features generated after each task with either the standard BIR method (left) or our improved version (right). Our method is able to retain much better precision-recall tradeoff of the generated samples. Experiment is performed for CIFAR-100 and T = 6.

TABLE 1. The average incremental accuracies on CIFAR-100 with the first task containing 50 classes and the rest 50 classes split into 5, 10, and 25 tasks equally.

CIL Method	T=6	T=11	T=26
Finetune	32.41±0.07	23.42±0.09	13.26±0.16
SI	35.32±0.35	26.13±0.74	15.6±0.27
EWC	32.64±0.05	23.53±0.74	13.33±0.08
LwF	51.38±0.16	43.57±0.26	22.63±0.08
BIR	54.52±0.29	51.16±0.57	44.95±0.59
BIR+SI	57.18±0.23	52.4±0.29	47.71±0.98
Ours	**59.05±0.42**	**57.97±0.99**	**53.75±0.32**
Joint		64.7	

TABLE 2. The average incremental accuracies on mini-ImageNet with the first task containing 50 classes and the rest 50 classes split into 5, 10, and 25 tasks equally.

CIL Method	T=6	T=11	T=26
Finetune	29.9±0.08	22±0.05	13.04±0.03
SI	30.68±0.34	23.27±0.27	14.08±0.22
EWC	30.03±0.03	22.07±0.06	13.09±0.02
LwF	45.84±0.3	39.28±0.29	21.47±0.22
BIR	47.86±0.22	44.15±0.58	38.93±0.83
BIR+SI	49.59±0.85	47.52±0.4	43.78±0.55
Ours	**52.45±1.22**	**52.79±2.1**	**48.94±0.71**
Joint		64.2	

and the rest 50 classes are divided evenly to 5, 10, and 25 tasks. The average incremental accuracies for CIFAR-100 are shown in Table 1, and the accuracies after each task for T = 5, 10, 25 are shown in the form of plots in Figure 6 (top). Our method shows better result in comparison with the baseline and regularization methods.

The second best method is BIR+SI, but, it is consistently worse than the proposed approach.

Similar results are presented for mini-ImageNet dataset, which consists of bigger images than CIFAR-100. Table 2 present average incremental accuracy for this dataset. Here, as well for CIFAR-100, our method outperforms the other in a meaning of average incremental accuracy. However, the difference between ours and BIR+SI is more significant with the increasing number of tasks, where for T = 26 we reach 48.94 and BIR+SI 43.78. The other regularization-based methods baselines for this scenario fall far behind. In Figure 6 (bottom) we see accuracies after each task. For mini-ImageNet BIR results in a better average accuracy in the second task for T = 6 and T = 11. This can be attributed to better plasticity (no SI). However, with a longer training and with more task, our method outperforms others.

For both datasets, SI alone presents the results comparable to finetuning. While simple application of LwF works good for smaller number of bigger tasks, T = 6 and T = 11, but for longer sessions T = 26 the performance significantly drops. Here, better adjustment of regularization hyper-parameters can play more important role. Our proposed method does not suffer from this issue.

B. NUMBER OF CYCLES

We analyze the influence of number of passes during cycling procedure on the average incremental accuracy for 6 tasks. According to the results presented in Figure 7, there is a drop of performance for small number of passes, but increasing the number improves the accuracy significantly. We suggest to search an optimal value for the number of passes depending on the dataset and split scenario used.

C. ABLATION STUDY

Through adding the proposed modifications one by one to the baseline method, we perform an ablation study for the proposed method. The obtained results can be seen in Table 3. The ablation study suggests that each of our modifications significantly contributes to the total performance of the model, and overall increase to average incremental accuracy is 5.56% over baseline.

TABLE 3. Ablation study of our method for class incremental learning setting with T = 6 and CIFAR-100. Average incremental accuracy is reported for ResNet32.

Approach	Latent match	Latent distillation	10 cycles	Acc.(%)
baseline method - BIR				54.22
w/ latent match	✓			56.21
w/ latent distillation	✓	✓		58.46
w/ 10 cycles	✓	✓	✓	59.78

D. ANALYSIS OF PRECISION AND RECALL

Finally, we perform the analysis of our models performance in terms of the quality of generations. To that end, we refer to the distribution precision and recall of the distributions as proposed by [30]. As authors indicate, those metrics disentangle FID score into two aspects: the quality of generated results (Precision) and their diversity (Recall). We calculate those two metrics on the features level and compare the resulting scores between standard BIR method and our improved approach. As presented in Figure 8, our improvements allow the model to retain both higher precision and recall of the regenerated samples.

VI. CONCLUSION AND FUTURE WORK

In conclusion, we propose the modifications to improve the VAE-based generative replay in the class-incremental setting. We observe the disparity between the latent representations of the original and generated data. Therefore, we incorporate the latent match loss that address this problem. To mitigate shift in the feature space during training on new data, we add latent distillation loss. Finally, we propose the cycling of the generated features though the previous model to decrease the distance between the distributions of original and generated samples. This allowed us to scale the generative approaches to more complex datasets, such as mini-ImageNet. The performed ablation study illustrates that the increase of performance due to each component.

In future, we plan to scale our method to perform well on more challenging scenarios such as ImageNet dataset and longer sequences of tasks.

This stands out as a notable limitation in numerous generative replay methods which are unsuitable for larger datasets, whereas our approach holds a significant advantage in this regard.

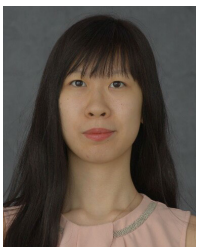
A. IMPACT STATEMENT

By using the generative approach for continual learning, our method does not require storing exemplars of past data, therefore it addresses concerns about private or sensitive data, which are applicable in some scenarios. However, generative models can retain the biases present in the training data, and we strongly advise a careful examination of their performance to ensure unbiased outcomes.

REFERENCES

[1] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars, "Memory aware synapses: Learning what (not) to forget," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 139–154.
[2] E. Belouadah and A. Popescu, "IL2M: Class incremental learning with dual memory," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 583–592.
[3] A. Chaudhry, M. Rohrbach, M. Elhoseiny, T. Ajanthan, P. K. Dokania, P. H. S. Torr, and M. Ranzato, "Continual learning with tiny episodic memories," in *Proc. ICML*, 2019, pp. 1–13.
[4] S. Golkar, M. Kagan, and K. Cho, "Continual learning via neural pruning," in *Proc. Neuro AI. Workshop NIPS*, 2019, pp. 1–12.
[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1–9.
[6] S. Gopalakrishnan, P. R. Singh, H. Fayek, S. Ramasamy, and A. Ambikapathi, "Knowledge capture and replay for continual learning," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 337–345.
[7] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local Nash equilibrium," in *Proc. NIPS*, 2017, pp. 1–12.
[8] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *Proc. NIPS*, 2015, pp. 1–9.
[9] D. Isele and A. Cosgun, "Selective experience replay for lifelong learning," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 1–8.
[10] R. Kemker and C. Kanan, "FearNet: Brain-inspired model for incremental learning," 2017, *arXiv:1711.10563*.
[11] V. Khan, S. Cygert, B. Twardowski, and T. Trzcinski, "Looking through the past: Better knowledge retention for generative replay in continual learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2023, pp. 3496–3500.
[12] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. ICLR*, 2014, pp. 1–14.
[13] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, and A. Grabska-Barwinska, "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 13, pp. 3521–3526, Mar. 2017.
[14] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Tech. Rep., 2009.
[15] T. Lesort, H. Caselles-Dupré, M. Garcia-Ortiz, A. Stoian, and D. Filliat, "Generative models from the perspective of continual learning," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.
[16] Z. Li and D. Hoiem, "Learning without forgetting," in *Computer Vision—ECCV* (Lecture Notes in Computer Science), vol. 9908, Amsterdam, The Netherlands. Cham, Switzerland: Springer, 2016, pp. 614–629.
[17] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2935–2947, Dec. 2018.
[18] X. Liu, C. Wu, M. Menta, L. Herranz, B. Raducanu, A. D. Bagdanov, S. Jui, and J. V. de Weijer, "Generative feature replay for class-incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2020, pp. 226–227.
[19] S. Lee, M. Weerakoon, J. Choi, M. Zhang, D. Wang, and M. Jeon, "CarM: Hierarchical episodic memory for continual learning," in *Proc. 59th ACM/IEEE Design Autom. Conf.*, Jul. 2022, pp. 6467–6476.
[20] A. Mallya and S. Lazebnik, "PackNet: Adding multiple tasks to a single network by iterative pruning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7765–7773.
[21] A. Mallya, D. Davis, and S. Lazebnik, "Piggyback: Adapting a single network to multiple tasks by learning to mask weights," in *Proc. ECCV*, 2018, pp. 67–82.
[22] M. Masana, X. Liu, B. Twardowski, M. Menta, A. D. Bagdanov, and J. van de Weijer, "Class-incremental learning: Survey and performance evaluation on image classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 5, pp. 5513–5533, May 2023, doi: 10.1109/TPAMI.2022.3213473.
[23] N. Y. Masse, G. D. Grant, and D. J. Freedman, "Alleviating catastrophic forgetting using context-dependent gating and synaptic stabilization," *Proc. Nat. Acad. Sci. USA*, vol. 115, no. 44, pp. 10467–10475, Oct. 2018.
[24] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of Learning and Motivation*, vol. 24. Amsterdam, The Netherlands: Elsevier, 1989, pp. 109–165.
[25] M. Mundt, S. Majumder, I. Pliushch, Y. W. Hong, and V. Ramesh, "Unified probabilistic deep continual learning through generative replay and open set recognition," 2020, arXiv:1905.12019v4.

[26] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," Tech. Rep., 2017.
[27] A. Prabhu, P. H. Torr, and P. K. Dokania, "GDUMB: A simple approach that questions our progress in continual learning," in *Computer Vision—ECCV*, Glasgow, U.K. Cham, Switzerland: Springer, 2020, pp. 524–540.
[28] D. J. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *Proc. 32nd Int. Conf. Mach. Learn. (PMLR)*, 2015, pp. 1530–1538.
[29] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," 2016, *arXiv:1606.04671*.
[30] M. S. Sajjadi, O. Bachem, M. Lucic, O. Bousquet, and S. Gelly, "Assessing generative models via precision and recall," 2018, *arXiv:1806.00035*.
[31] S. Scardapane and A. Uncini, "Pseudo-rehearsal for continual learning with normalizing flows," in *Proc. ICML*, 2020, pp. 1–9.
[32] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," in *Proc. Adv. Neural Inf. Process. Syst.*, Long Beach, CA, USA, 2017, pp. 2990–2999.
[33] G. M. van de Ven, H. T. Siegelmann, and A. S. Tolias, "Brain-inspired replay for continual learning with artificial neural networks," *Nature Commun.*, vol. 11, no. 1, p. 4069, Aug. 2020.
[34] G. M. van de Ven and A. S. Tolias, "Generative replay with feedback connections as a general strategy for continual learning," 2018, *arXiv:1809.10635*.
[35] J. Yoon, E. Yang, J. Lee, and S. J. Hwang, "Lifelong learning with dynamically expandable networks," in *Proc. ICLR*, 2018, pp. 1–11.
[36] J. Xu and Z. Zhu, "Reinforced continual learning," in *Proc. NIPS*, 2018, pp. 1–10.
[37] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3987–3995.
[38] D.-W. Zhou, F.-Y. Wang, H.-J. Ye, and D.-C. Zhan, "PyCIL: A Python toolbox for class-incremental learning," *Sci. China Inf. Sci.*, vol. 66, no. 9, Sep. 2023, Art. no. 197101.



VALERIYA KHAN received the master's degree in cloud computing. She is currently pursuing the Ph.D. degree with Warsaw University of Technology and IDEAS NCBR, where she focuses on the topic of continual learning of artificial neural networks. Prior to working at IDEAS, she was with Samsung's team developing computer vision algorithms for gaze tracking.

SEBASTIAN CYGERT received the Ph.D. degree from Gdańsk University of Technology. He is currently a Postdoctoral Researcher with IDEAS NCBR and an Assistant Professor with Gdańsk University of Technology. Previously, he was employed as an Applied Scientist with Amazon and contributed to projects, such as the visual perception system for the autonomous robot Amazon Scout. In addition, he is collaborating with the Medical University of Gdańsk on a project aimed at early cancer diagnosis through the use of liquid biopsies. His research interest includes the real-world generalization and efficient computation of machine learning algorithms.

KAMIL DEJA received the Ph.D. degree from Warsaw University of Technology. He is currently a Postdoctoral Researcher with IDEAS NCBR and Warsaw University of Technology. He has previously interned at Vrije Universiteit Amsterdam and twice at Amazon Alexa. His research interest includes generative modeling with applications to continual learning. His research work has been published in prestigious conferences, such as NeurIPS, IJCAI, and Interspeech. In recognition of his accomplishments, he received the FNP Start Scholarship, in 2023, awarded to the top 100 young researchers in Poland.

TOMASZ TRZCINSKI (Senior Member, IEEE) received the M.Sc. degree from Universitat Politècnica de Catalunya, the M.Sc. degree from Politecnico di Torino, in 2010, the Ph.D. degree from École Polytechnique Fédérale de Lausanne, in 2014, and the D.Sc. degree from Warsaw University of Technology, in 2020. He was an Associate Professor with Jagiellonian University, Kraków, from 2020 to 2023; a Visiting Scholar with Stanford University, in 2017; and Nanyang Technological University, in 2019. Previously, he was with Google, in 2013; Qualcomm, in 2012; and Telefónica, in 2010. He is currently an Associate Professor with Warsaw University of Technology, where he leads the Computer Vision Laboratory. He is also a Chief Scientist at Tooploox and the Co-Founder of Comixify, a technology startup focused on using machine learning algorithms for video editing. He is the Computer Vision Group Leader with IDEAS NCBR, a publicly-funded Polish Center for AI. He is a member of the ELLIS Society, a member of the ALICE Collaboration at CERN, and an Expert of the National Science Centre and Foundation for Polish Science. He serves as a Reviewer in major computer science conferences, such as CVPR, ICCV, ECCV, NeurIPS, and ICML, and journals, such as IEEE Transactions on Pattern Analysis and Machine Intelligence, *International Journal of Computer Vision*, and *Computer Vision and Image Understanding*. He is an Associate Editor of IEEE Access and *Electronics* (MDPI).

BARTLOMIEJ TWARDOWSKI received the Ph.D. degree, in 2018, with a focus on recommender systems and neural networks. He was an Assistant Professor with the AI Group, Warsaw University of Technology, for 1.5 years, before deciding to join the Computer Vision Center, Universitat Autònoma de Barcelona (UAB), for a postdoctoral program. He is currently the Research Team Leader with the IDEAS NCBR Research Institute and a Researcher with the Computer Vision Center, UAB. He has been actively involved in various research projects related to DL/NLP/ML (ranging from €40k to €1.4 million). He has wide industry experience (more than 15 years), including international companies, such as Zalando, Adform, Huawei, and Naspers Group (Allegro), as well as helping startups with research projects (Sotrender, Scattered). Throughout his career, he has had the opportunity to publish papers in prestigious conferences, such as CVPR, in 2020, two papers; NeurIPS, in 2020; ICCV, in 2021 and 2023; ICLR, in 2023; and ECIR, in 2021 and 2023. His research interests include lifelong machine learning in computer vision, efficient neural network training, transferability, domain adaptation, information retrieval, and recommender systems. He is a Ramón y Cajal Fellow. In addition, he has served as a Reviewer for multiple AI/ML conferences, such as AAAI, CVPR, ECCV, ICCV, ICML, and NeurIPS.