Multiband VAE: Latent Space Alignment for Knowledge Consolidation in Continual Learning

Kamil Deja\textsuperscript{1}, Paweł Wawrzyński\textsuperscript{1}, Wojciech Masarczyk\textsuperscript{1}, Daniel Marczak\textsuperscript{1} and Tomasz Trzcński\textsuperscript{1,2,3}

\textsuperscript{1}Warsaw University of Technology,
\textsuperscript{2}Jagiellonian University,
\textsuperscript{3}Tooploox
kamil.deja.dokt@pw.edu.pl

Abstract

We propose a new method for unsupervised generative continual learning through realignment of Variational Autoencoder’s latent space. Deep generative models suffer from catastrophic forgetting in the same way as other neural structures. Recent generative continual learning works approach this problem and try to learn from new data without forgetting previous knowledge. However, those methods usually focus on artificial scenarios where examples share almost no similarity between subsequent portions of data – an assumption not realistic in the real-life applications of continual learning. In this work, we identify this limitation and posit the goal of generative continual learning as a knowledge accumulation task. We solve it by continuously aligning latent representations of new data that we call \textit{bands} in additional latent space where examples are encoded independently of their source task. In addition, we introduce a method for controlled forgetting of past data that simplifies this process. On top of the standard continual learning benchmarks, we propose a novel challenging knowledge consolidation scenario and show that the proposed approach outperforms state-of-the-art by up to twofold across all experiments and the additional real-life evaluation. To our knowledge, Multiband VAE is the first method to show forward and backward knowledge transfer in generative continual learning.

1 Introduction

Recent advances in generative models [Goodfellow et al., 2014; Kingma and Welling, 2014] led to their unprecedented proliferation across many real-life applications. This includes high energy physics experiments at Large Hadron Collider (LHC) at CERN, where they are employed to speed up the process of particles collisions simulations [Paganini et al., 2018; Deja et al., 2020; Kansal et al., 2021].

Those applications are possible, thanks to the main objective of generative methods, which is the modelling of complex data manifolds with simpler distributions. Unfortu-
we expect no forgetting of previous knowledge, in other sce-
narios, where model is retrained with additional partially sim-
ilar data, we should aim for performance improvement. This
can be observed through forward knowledge transfer – higher
performance on a new task, thanks to already incorporated
knowledge, and backward knowledge transfer – better gener-
ations from previous tasks, when retrained on additional sim-
ilar examples [Lopez-Paz and Ranzato, 2017].

Therefore, to simulate real-life conditions, we prepare a set
of diversified continual learning scenarios with data splits fol-
lowing Dirichlet distribution, inspired by a similar approach
in federated learning [Hsu et al., 2019]. Our experiments indi-
cate that this is indeed a more challenging setup for the ma-
jority of recent state-of-the-art continual generative models,
which lack sufficient knowledge sharing between tasks.

To mitigate this problem, we propose a Multiband VAE.
The core idea behind our method is to split the process of
model retraining into two steps: (1) a local encoding of data
from the new task into a new model’s latent space and (2) a
global rearrangement and consolidation of new and previous
data. In particular, we propose to align local data representa-
tions from consecutive tasks through the additional neural
network. In reference to the way how radio spectrum fre-
quencies are allocated, we name data representations from
different tasks bands. As in telecommunication, our goal is
to limit interference between bands. However, we train our
model to align parts that represent the same or similar data.
To support knowledge consolidation between different bands,
we additionally propose a controlled forgetting mechanism
that enables the substitution of degraded reconstructions of
past samples with new data from the current task.

The main contributions of this work are:

• A novel method for generative continual learning of
  Variational Autoencoder that counteracts catastrophic
  forgetting while being able to align even partially sim-
  ilar tasks at the same time.

• A simple method for controlled forgetting of past exam-
  ples whenever a new similar data is presented.

• A novel knowledge consolidation training scenario that
  underlines limitations of recent state-of-the-art methods.

2 Related Works

Most of the works incorporating generative models in contin-
ual learning relate to generative rehearsal. In this technique,
the base model is trained with a mixture of new data examples
from the current task and recreation of previous samples gen-
erated by a generative model. This idea was first introduced
by [Shin et al., 2017], with Generative Adversarial Networks
(GAN) trained with the self rehearsal method so-called Gen-
erative Replay (GR). [Lesort et al., 2019] overview different
generative models trained with the GR method. Our Multi-
band VAE is a direct extension to this technique.

Continual learning of generative models  [Nguyen et al.,
2018] adapt regularization-based methods such as Elastic
Weight Consolidation (EWC) [Kirkpatrick et al., 2017], and
Synaptic Intelligence (SI) [Zenke et al., 2017] to the contin-
ual learning in generative models regularizing the adjustment
of the most significant weights. The authors also introduce
Variational Continual Learning (VCL), with adjustments in
parts of the model architecture for each task.

In HyperCL, [von Oswald et al., 2019] propose entirely
different approach, where a hypernetwork generates the
weights of the continually trained model. This yields state-
of-the-art results in discriminative models task-incremental
training but is also applicable to the generative models. In or-
der to differentiate tasks, [Rao et al., 2019] propose CURL
that learns task-specific representation and deals with task
ambiguity by performing task inference within the genera-
tive model. This approach directly addresses the problem
of forgetting by maintaining a buffer for original instances
of poorly-approximated samples and expanding the model
with a new component whenever the buffer is filled. In
BooVae, [Egorov et al., 2021] propose an approach for con-
tinual learning of VAE with an additive aggregated posterior
expansion. Several works train GANs in the continual learn-
ing scenarios either with memory replay [Wu et al., 2018],
with the extension to VAEGAN in Lifelong-VAEGAN by [Ye
and Bors, 2020].

Continual learning with disentanglement  In VASE by
[Achille et al., 2018], authors propose a method for continual
learning of shared disentangled data representation. While
encoding images with a standard VAE, VASE also seeks
shared generative factors. A similar concept of mixed-type
latent space was introduced in Lifelong VAE [Ramapuram et
al., 2020], where it is composed of discrete and continuous
values. In this work we also use a disentanglement method
with binary latent space.

3 Method

In this section, we introduce Multiband VAE – a method for
consolidating knowledge in a continually learned generative
model. We propose to split generative replay training into
two parts: (1) a local training that allows us to build a new
data representations band in the latent space of VAE, and (2)
global training where we attach a newly trained band to the
already trained global model. As a part of the global train-
ing, we propose a controlled forgetting mechanism where we
replace selected reconstructions from previous tasks with cur-
rently available data.

3.1 Knowledge Acquisition – Local Training

In the local training, we learn a new data representations band
by training a VAE using only currently available data.

Let $x_j^i$ denote the $j$-th sample of $i$-th task. Then, for given
sample $x_j^i$, and latent variable $\lambda_j^i$ we use a decoder $p_\phi$, which
is trained to maximize posterior probability $p(x_j^i|\lambda_j^i)$. To get
the latent variable $\lambda_j^i$, we use encoder $q_\theta$ parametrized with
weights vector $\phi$ that approximates probability $q(\lambda_j^i|x_j^i)$.

To simplify the notation, let us focus on specific task $i$ and
drop the index. As in standard VAE, we follow optimization
introduced by [Kingma and Welling, 2014] that maximizes
the variational lower bound of log likelihood:

$$\max_{\theta,\phi} \mathbb{E}_{q(\lambda|x)} [\log p(x|\lambda)] - D_{KL}(q(\lambda|x)||N(\hat{0}, I))).$$ (1)
where $\theta$ and $\phi$ are weights of encoder and decoder respectively. In the first task, this is the only part of the training, after which local decoder is remembered as a global one. In other cases we drop local decoder.

### 3.2 Shared Knowledge Consolidation

In the second - global part of the training, we align the newly trained band with already encoded knowledge. The simplest method to circumvent interference between bands is to partition the latent space of VAE and place new data representation in a separate area of latent space. However, such an approach limits information sharing across separate tasks and hinders forward and backward knowledge transfer. Therefore, in Multiband VAE we propose to align different latent spaces through an additional neural network that we call translator. Translator maps individual latent spaces which are conditioned with task id into the common global one where examples are stored independently of their source task, as presented in Fig. 2.

To that end, we define a translator network $t_\rho(\lambda^i, i)$ that learns a common alignment of separate latent spaces $\lambda^i$ conditioned with task id $i$ to a single latent variable $Z$, where all examples are represented independently of their source task. Finally, we propose a global decoder $p_\omega(x|Z)$ that based on distribution approximated with latent variables $Z$ learns to approximate original data distribution $x$.

To counteract forgetting, when training translator and global decoder we use auto-rehearsal as in standard generative replay, with a copy of the translator and decoder frozen at the beginning of the task. As training pairs, we use combination of original images $x$ with their encodings from local encoder $\lambda$, and for previous tasks, random values $\lambda$ with generations $x$ reconstructed with a frozen translator and global decoder. Fig. 3 presents the overview of this procedure.

We start translator training with a frozen global decoder, to find the best fitting part of latent space $Z$ for a new band of data without disturbing previous generations. For that end we minimize the reconstruction loss:

$$\min_\rho \frac{1}{k} \sum_{i=1}^{k} ||x^i - p_\omega(t_\rho(\lambda^i, i))||_2^2,$$

where $k$ is the number of all tasks.

Then, we optimize parameters of translator and global decoder jointly, minimizing the reconstruction error between outputs from the global decoder and training examples

$$\min_{\rho, \omega} \frac{1}{k} \sum_{i=1}^{k} ||x^i - p_\omega(t_\rho(\lambda^i, i))||_2^2,$$

To generate new example $t$ with Multiband VAE, we randomly sample task id $i \sim U(\{1, \ldots, k\})$, where $k$ is the number of all tasks and latent representation $\lambda_i \sim N(\bar{0}, I)$. These values are mapped with translator network to latent variable $z_t$, which is the input to global decoder to generate $x_t$. Therefore, translator and global decoder are the only models that are stored in-between tasks.

### 3.3 Controlled Forgetting

In a real-life scenario, it is common to encounter similar data examples in many tasks. In such a case, we would like our continuously trained model to refresh the memory of examples instead of combining vague, distorted memories with new instances. Therefore, we propose a mechanism for controlled forgetting of past reconstructions during the translator and global decoder joint training. To that end, when creating new training pairs, we compare representations of previous data reconstructions generated as new targets with representations of data samples from the current task in the common latent space $Z$. If these representations are similar enough, we substitute previous data reconstruction with the current data sample as presented in Fig. 4.

More specifically, when training on task $i$, we first create a subset $Z^i = t_\rho(q_\phi(x^i), i)$ with representations of all currently available data in joint latent space $Z$. Now, for each data sample $x_j$ generated as a rehearsal target from previous task $l < i$ and random variable $\lambda^j_l$, we compare its latent representation $z_j = t_\rho(\lambda^j_l, j)$ with all elements of set $Z^i$.

$$\text{sim}(z_j) := \max_{z_q \in Z^i} \cos(z_j, z_q).$$

To counteract forgetting, when training translator and global decoder, we perform a comparison of representations of previous data reconstructions generated as new targets with representations of data samples from the current task in the common latent space $Z$. If these representations are similar enough, we substitute previous data reconstruction with the current data sample as presented in Fig. 4.

More specifically, when training on task $i$, we first create a subset $Z^i = t_\rho(q_\phi(x^i), i)$ with representations of all currently available data in joint latent space $Z$. Now, for each data sample $x_j$ generated as a rehearsal target from previous task $l < i$ and random variable $\lambda^j_l$, we compare its latent representation $z_j = t_\rho(\lambda^j_l, j)$ with all elements of set $Z^i$.

$$\text{sim}(z_j) := \max_{z_q \in Z^i} \cos(z_j, z_q).$$

To counteract forgetting, when training translator and global decoder, we perform a comparison of representations of previous data reconstructions generated as new targets with representations of data samples from the current task in the common latent space $Z$. If these representations are similar enough, we substitute previous data reconstruction with the current data sample as presented in Fig. 4.
Figure 5: Visualization of latent space $\mathcal{Z}$ and generations from VAE in standard Generative Replay and our multiband training for the three tasks (different colors) in a case of entirely different new data distribution, and partially same classes. GR does not instantly separate data from different tasks, which results in the deformation of previously encoded examples. Contrary, our Multiband VAE can separate representations from different classes while properly aligning examples from the same new class if present.

If $\text{sim}(z_j) \geq \gamma$ we substitute target sampled reconstruction $x'_j$ with respective original image from $x^i$. Intuitively, $\gamma$ controls how much do we want to forget from task to task, with $\gamma = 0.9$ being a default value for which we observe a stable performance across all benchmarks.

4 Experiments

To visualize the difference between Generative Replay and Multiband VAE, in Fig. 5 we present a toy-example with the MNIST dataset limited to 3 tasks with data examples from 3 classes. When presented with data from a new distribution (different class in task 2), our method places a new band of data in a separate part of a common latent space $\mathcal{Z}$. On the other hand, the standard generative replay model learns to transform some of the previous data examples into currently available samples before it can distinguish them, even with additional conditioning on task identity. At the same time, when presented data with partially same classes as in task 3, our translator is able to properly align bands of data representations so that similar data examples (in this case ones) are located in the same area of latent space $\mathcal{Z}$ independently of the source task, without interfering with zeros and twos.

4.1 Evaluation Setup

For fair comparison, in all evaluated methods we use a Variational Autoencoder architecture similar to the one introduced by [Nguyen et al., 2018], with nine dense layers. However, our Multiband VAE is not restricted to any particular architecture, so we also include experiments with a convolutional version. The exact architecture and training hyperparameters are enlisted in the appendix and code repository. We do not condition our generative model with class identity since it greatly simplifies the problem of knowledge consolidation and applies to all evaluated methods. However, similarly to [Ramapuram et al., 2020], we use additional binary latent space trained with Gumbel softmax [Jang et al., 2016].

1https://github.com/KamilDeja/multiband_vae

Figure 6: Class splits for different continual learning scenarios. In class incremental split each task consists of separate classes. For $\alpha = 1$ Dirichlet distribution, we have highly imbalanced splits with randomly occurring dominance of one or two classes. For higher values of parameter $\alpha$, classes are split almost equally.

4.2 Evaluation

To assess the quality of our method, we conduct a series of experiments on benchmarks commonly used in continual learning (MNIST, Omniglot [Lake et al., 2015]) and generative modeling – FashionMNIST [Xiao et al., 2017]. Since the performance of VAE on diverse datasets like CIFAR is limited, in order to evaluate how our method scales to more complex data, we include tests on CelebA [Liu et al., 2015]. For each dataset, we prepare a set of training scenarios designed to evaluate various aspects of continual learning. This is the only time we access data classes, since our solution is fully unsupervised.

To assess whether the model suffers from catastrophic forgetting, we run class incremental scenarios introduced by [Van de Ven and Tolias, 2019]. However, CI simplifies the problem of learning data distribution in the generative model’s latent space since the identity of the task conditions final generations. Therefore, we also introduce more complex data splits with no assumption of independent task distributions. To that end, we split examples from the same classes into tasks, according to the probability $q \sim \text{Dir}(\alpha p)$ sampled from the Dirichlet distribution, where $p$ is a prior class distribution over all classes, and $\alpha$ is a concentration parameter that controls similarity of the tasks, as presented in Fig. 6. In particular, we exploit the Dirichlet $\alpha = 1$ scenario, where the model has to learn the differences between tasks while
To measure the quality of generations from different methods, we use the Fréchet Inception Distance (FID) [Heusel al., 2017]. As proposed by [Bílkowski et al., 2018], for simpler datasets, we calculate FID based on the LeNet classifier pre-trained on the whole target dataset. Additionally, we report the precision and recall of the distributions as proposed by [Sajjadi et al., 2018]. As authors indicate, those metrics disentangle FID score into two aspects: the quality of generated results (Precision) and their diversity (Recall).

For each experiment, we report the FID, Precision, and Recall averaged over the final scores for each task separately. For methods that do not condition generations on the task index (CuRL and LifelongVAE), we calculate measures in comparison to the whole test set. The results of our experiments are presented in Tab. 1 and Tab. 2, where we show scores averaged over three runs with different random seeds.

To compare different continual-learning generative methods in a real-life scenario we also use real data from detector responses in the LHC experiment. Calorimeter response simulation is one of the most profound applications of generative models where those techniques are already employed in practice [Paganini et al., 2018]. In our studies, we use a dataset of real simulations from Zero Degree Calorimeter in the ALICE experiment at CERN introduced by [Deja et al., 2020], where a model is to learn outputs of $44 	imes 44$ resolution energy depositions in calorimeter. Following [Deja et al., 2020], instead of using FID, for evaluation, we benefit from the nature of the data and compare the distribution of real and generated channels – the sum of selected pixels that well describe the physical properties of simulated output. We report the Wasserstein distance between original and generated channels distribution poses a greater challenge than the class incremental scenario. However, our Multiband VAE can precisely consolidate knowledge from such complex setups, while still preventing forget-

### Table 1: Average FID and distribution Precision (Prec) and Recall (Rec) or Wasserstein distance between original and generated simulation channels, after the final task in different data incremental scenarios. Our method with vanilla architecture outperforms competing solution.

<table>
<thead>
<tr>
<th>Num. tasks</th>
<th>Measure</th>
<th>Split-Omniglot</th>
<th>Split-Omniglot</th>
<th>Omniglot</th>
<th>FashionMNIST-→MNIST</th>
<th>CERN Class Inc.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SI</td>
<td>5</td>
<td>20</td>
<td>20</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>EWC</td>
<td>48</td>
<td>68</td>
<td>156</td>
<td>146</td>
<td>157</td>
</tr>
<tr>
<td></td>
<td>Generative replay</td>
<td>45</td>
<td>88</td>
<td>106</td>
<td>119</td>
<td>133</td>
</tr>
<tr>
<td></td>
<td>VCL</td>
<td>48</td>
<td>87</td>
<td>98</td>
<td>84</td>
<td>140</td>
</tr>
<tr>
<td></td>
<td>HyperCL</td>
<td>22</td>
<td>95</td>
<td>79</td>
<td>93</td>
<td>137</td>
</tr>
<tr>
<td></td>
<td>CURL</td>
<td>31</td>
<td>90</td>
<td>112</td>
<td>91</td>
<td>137</td>
</tr>
<tr>
<td></td>
<td>Lifelong-VAE</td>
<td>49</td>
<td>87</td>
<td>79</td>
<td>93</td>
<td>137</td>
</tr>
<tr>
<td></td>
<td>Lifelong-VAE</td>
<td>31</td>
<td>90</td>
<td>71</td>
<td>85</td>
<td>127</td>
</tr>
<tr>
<td>Multiband VAE</td>
<td>21</td>
<td>97</td>
<td>33</td>
<td>95</td>
<td>41</td>
<td>51</td>
</tr>
<tr>
<td>Multiband VAE (conv)</td>
<td>12</td>
<td>98</td>
<td>24</td>
<td>95</td>
<td>24</td>
<td>49</td>
</tr>
</tbody>
</table>

### Table 2: Average Fréchet Inception Distance (FID) and distribution Precision (Prec) and Recall (Rec) after the final task in different data incremental scenarios. In more challenging datasets Multiband VAE outperforms competing solutions.

<table>
<thead>
<tr>
<th>Num. tasks</th>
<th>Measure</th>
<th>Split-Omniglot</th>
<th>Split-Omniglot</th>
<th>Omniglot</th>
<th>FashionMNIST-→MNIST</th>
<th>CERN Class Inc.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SI</td>
<td>5</td>
<td>20</td>
<td>20</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>EWC</td>
<td>48</td>
<td>68</td>
<td>156</td>
<td>146</td>
<td>157</td>
</tr>
<tr>
<td></td>
<td>Generative replay</td>
<td>45</td>
<td>88</td>
<td>106</td>
<td>119</td>
<td>133</td>
</tr>
<tr>
<td></td>
<td>VCL</td>
<td>48</td>
<td>87</td>
<td>98</td>
<td>84</td>
<td>140</td>
</tr>
<tr>
<td></td>
<td>HyperCL</td>
<td>22</td>
<td>95</td>
<td>79</td>
<td>93</td>
<td>137</td>
</tr>
<tr>
<td></td>
<td>CURL</td>
<td>31</td>
<td>90</td>
<td>112</td>
<td>91</td>
<td>137</td>
</tr>
<tr>
<td></td>
<td>Lifelong-VAE</td>
<td>49</td>
<td>87</td>
<td>79</td>
<td>93</td>
<td>137</td>
</tr>
<tr>
<td></td>
<td>Lifelong-VAE</td>
<td>31</td>
<td>90</td>
<td>71</td>
<td>85</td>
<td>127</td>
</tr>
<tr>
<td>Multiband VAE</td>
<td>21</td>
<td>97</td>
<td>33</td>
<td>95</td>
<td>41</td>
<td>51</td>
</tr>
<tr>
<td>Multiband VAE (conv)</td>
<td>12</td>
<td>98</td>
<td>24</td>
<td>95</td>
<td>24</td>
<td>49</td>
</tr>
</tbody>
</table>
In this work, we propose a new method for unsupervised continual learning of generative models. We observe that the currently employed class-incremental scenario simplifies the continual learning of generative models. Therefore, we propose a novel, more realistic scenario, with which we experimentally highlight the limitations of state-of-the-art methods. Finally, we introduce a new method for continual learning of generative models based on the constant consolidation of VAE’s latent space. To our knowledge, this is the first work that experimentally shows that with continually growing data with even partially similar distribution, we can observe both forward and backward performance improvement. Our experiments on various benchmarks and with real-life data show the superiority of Multiband VAE over related methods, with upper-bound performance in some training scenarios.

5 Conclusion

In this work, we propose a new method for unsupervised continual learning of generative models. We observe that the currently employed class-incremental scenario simplifies the continual learning of generative models. Therefore, we propose a novel, more realistic scenario, with which we experimentally highlight the limitations of state-of-the-art methods. Finally, we introduce a new method for continual learning of generative models based on the constant consolidation of VAE’s latent space. To our knowledge, this is the first work that experimentally shows that with continually growing data with even partially similar distribution, we can observe both forward and backward performance improvement. Our experiments on various benchmarks and with real-life data show the superiority of Multiband VAE over related methods, with upper-bound performance in some training scenarios.

Table 3: Average FID, distribution Precision, and Recall after the final task on the CelebA dataset. Our Multiband VAE consolidates knowledge from separate tasks even in the class incremental scenario, clearly outperforming other solutions. With more even splits our method converges to the upper bound which is a model trained with full data availability.

<table>
<thead>
<tr>
<th>CelebA split</th>
<th>Class Incremental</th>
<th>Dirichlet $\alpha = 1$</th>
<th>Dirichlet $\alpha = 100$</th>
<th>Single split</th>
</tr>
</thead>
<tbody>
<tr>
<td>Num. tasks</td>
<td>5</td>
<td>10</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>Measure</td>
<td>FID↓ Prec↑ Rec↑</td>
<td>FID↓ Prec↑ Rec↑</td>
<td>FID↓ Prec↑ Rec↑</td>
<td>FID↓ Prec↑ Rec↑</td>
</tr>
<tr>
<td>Separate models</td>
<td>103 31 21</td>
<td>105 24.5 7.6</td>
<td>109 28.4 10.6</td>
<td>88 35 30</td>
</tr>
<tr>
<td>Generative Replay</td>
<td>105 23.4 14.9</td>
<td>109 14.6 7.4</td>
<td>102 17.2 11.6</td>
<td></td>
</tr>
<tr>
<td>Multiband VAE</td>
<td>95 28.5 23.2</td>
<td>93 33 22</td>
<td>89 36.2 28</td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Ablation study on the MNIST dataset with Dirichlet $\alpha = 1$ distribution. Average FID after the last task.

<table>
<thead>
<tr>
<th>Modification</th>
<th>FID↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generative replay</td>
<td>254</td>
</tr>
<tr>
<td>+ Two step training</td>
<td>64</td>
</tr>
<tr>
<td>+ Translator</td>
<td>53</td>
</tr>
<tr>
<td>+ Binary latent space</td>
<td>44</td>
</tr>
<tr>
<td>+ Controlled forgetting</td>
<td>41</td>
</tr>
<tr>
<td>+ Convolutional model</td>
<td>30</td>
</tr>
</tbody>
</table>

Figure 7: Comparison of Wasserstein distance ↓ between original simulation channels and generations from VAE trained with standard GR and our multiband training. Multiband VAE well consolidates knowledge with forward transfer (each row starts with better score) and backward knowledge transfer (improvement for some rows when retrained with more data). At the same time standard GR struggles to retain quality of generations on old tasks.

Ablation study The main contribution of this work is a multiband training procedure, yet we also introduce several mechanisms that improve knowledge consolidation. Tab. 4 shows how those components contribute to the final score.
Acknowledgments

This research was funded by Foundation for Polish Science (grant no POIR.04.04.00-00-14DE/18-00 carried out within the Team-Net program co-financed by the European Union under the European Regional Development Fund) and National Science Centre, Poland (grant no 2018/31/N/ST6/02374 and 2020/39/B/ST6/01511).

References


