Modular-Relatedness for Continual Learning

Ammar Shaker, Shujian Yu, Francesco Alesiani

November 4, 2021

Abstract

In this paper, we propose a continual learning (CL) technique that is beneficial to sequential task learners by improving their retained accuracy and reducing catastrophic forgetting. The principal target of our approach is the automatic extraction of modular parts of the neural network and then estimating the relatedness between the tasks given these modular components. This technique is applicable to different families of CL methods such as regularization-based (e.g., the Elastic Weight Consolidation) or the rehearsal-based (e.g., the Gradient Episodic Memory) approaches where episodic memory is needed. Empirical results demonstrate remarkable performance gain (in terms of robustness to forgetting) for methods such as EWC and GEM based on our technique, especially when the memory budget is very limited.

Introduction

Despite the success in outperforming humans on tasks such as playing Go with and without human guidance (Silver et al. 2016; Silver et al. 2017), machine learning methods still lack the human ability to learn multiple tasks with less suffering from conflicting skills. Continual learning is a branch of lifelong learning that tackles task accumulation, while minimizing the effect of forgetting previous tasks. In recent years, there have been significant advancements in mitigating catastrophic forgetting through different learning schemes. The regularization-based methods such as EWC (Kirkpatrick et al. 2017) and R-EWC (Liu et al. 2018) penalize model updates that seem to be harmful to previously observed tasks. Rehearsal-based methods such as GEM (Lopez-Paz and Ranzato 2017), AGEM (Chaudhry et al. 2018b), and iCaRL (Rebuffi et al. 2017) aim at weakening the forgetting by replaying real or pseudo examples while learning new tasks.

Despite satisfactory performances achieved by these methods, they usually fail to consider the task-relatedness between the new task and previous ones in the learning objective.

There are only a few exceptions that consider task-relatedness. Dynamically expandable network (DEN) (Yoon et al. 2017) computes the relatedness to decide whether the network’s capacity should be increased in a layer-wise manner. The expert gate (EG) (Aljundi, Chakravarty, and Tuytelaars 2017), on the other hand, trains an autoencoder in each task and estimates the relatedness between the new task $\tau$ and the $t$-th old task with the reconstruction error of the $t$-th autoencoder on the data from the task $\tau$. Both methods follow a train-and-evaluate framework to estimate the task relatedness, which is always computational expensive and sensitive to the selected models. For example, if one resorts to a powerful autoencoder in EG, this autoencoder could likely fit the data from all tasks with just marginal differences.

Task rehearsal method (TRM) (Silver and Mercer 2002) is one of the early works that measure the relatedness in the tasks sequential setting. This relatedness is then used to generate rehearsal examples to achieve transfer learning in a functional manner.

Despite the explicit relatedness computation in these methods, the absence of the network’s modular decomposition causes the modular-relatedness to be ignored.

Motivated by evidence from neuropsychology and neuropsychology that animal and human brains are organized into segregated modules based on their functionality (Azam 2000), a modular neural network is an aggregation of computationally independent sub-networks.

In this work, we present a general approach to mitigate task forgetting by deriving a technique that integrates task relatedness into the learning of modular networks. The detailed contributions of this paper are threefold:

(i) A novel CL framework based on modular-relatedness: It enables the automatic discovering of groups of neurons (in each layer) that are mutually independent or less dependent, and proposes an adaption of the learning process to consider the relevance between tasks given each of these groups.

(ii) Two example realizations of how modular-relatedness could be implemented: As a proof of concept, we propose modular extensions to EWC and GEM.

(iii) We derive a novel theoretical result showing that the symmetric KL divergence takes advantage of the full space of the Fisher information matrix compared to EWC’s diagonal Fisher matrix.

Related Work

Continual Learning

Regularization-based CL methods are a family of methods that add a penalty term to the learning objective such that older knowledge is retained when learning on new tasks. Elastic weight consolidation (EWC) (Kirkpatrick et al. 2017) penalizes the change of each parameter by a weight proportional to the diagonal Fisher information matrix (FIM) (the diagonal of the inverse Hessian of the negative log-likelihood). R-EWC (Liu et al. 2018) suggests rotating the parameter space such that FIM is diagonal. As an alternative to computing FIM, synaptic intelligence...
sequential task learning: Consider a continual learning problem where the set of parameters \( \theta \) needs to be adapted to new tasks \( t \) while maintaining performance on previous tasks. A popular approach is to keep the neural network architecture fixed and only update the weights. However, this can lead to a phenomenon known as the catastrophic forgetting problem, where the performance on old tasks deteriorates as new tasks are learned.

To address this, various methods have been proposed to modularize the neural network. Modularization can be achieved at different levels: sub-networks, layers, or even individual neurons. For example, (Jaderberg et al. 2017) developed a method called Generalized backpropagation that allows for efficient backpropagation across tasks by approximating gradient signals locally.

Neural network parametrization: Consider representing the neural network by the function \( f(x; \theta) : \mathbb{R}^p \rightarrow [0, 1]^{|Y|} \) that computes the score \( f(x; \theta) \) for each category \( c \in Y \) being the correct label for the instance \( x \) through a multi-layered neural network parameterized by \( \theta \in \Theta \). For a \( D \)-layered network, the set of parameters \( \theta = \{ \omega_d^i, b_d^i | d \in 1, \ldots, D - 1 \} \) contains the weights \( \omega_d^i \) of the connections between the units \( u_d^i \) in the \( d \)th layer and the units \( u_{d+1}^j \) in the \((d+1)\)th layer, and the bias terms \( b_d^j \) of the units in the \( d \)th layer. The scoring function resulting from the forward propagation in the \( D \)-layered network takes the form:

\[
f_j(x; \theta) = \phi \left( \sum_i \omega_d^i \cdot o_d^{i-1} + b_d^j \right)
\]

where \( o_d^j \) is the \( j \)th unit’s output at the \( d \) layer, \( o_d^1 \) represent the features of the input data, and \( \phi \) is an activation function. This Eq. 2 is indeed the function \( o_d^j(x) \) that computes the representation of \( x \) given all the units of the previous layers, \( 1, \ldots, d-1 \) and the connections from layer \( d-1 \) to the unit \( u_d^i \). For a given loss function \( L \), multi-task learning methods aim at finding a general parametrization \( \theta \) that minimizes the objective \( \sum_{t \in T} E_{(x_t, y_t) \sim T_t}(\mathcal{L}(f(x_t; \theta), y_t)) \) (Chen and Li 2018), i.e., observing all tasks at the same time and minimizing their joint loss simultaneously. Generally speaking, after learning on \( t - 1 \) tasks, continual learning aims at finding \( \theta_t \) that is the least harmful to the previous tasks:

\[
\arg \min_{\theta} \mathbb{E}_{(x_t, y_t) \sim T_t} \mathcal{L}(f(x_t; \theta), y_t)
\]

s.t. \( \mathbb{E}_{(x_k, y_k) \sim T_k}(\mathcal{L}(f(x_k; \theta_t), y_k)) \leq \mathbb{E}_{(x_k, y_k) \sim T_k}(\mathcal{L}(f(x_k; \theta_{t-1}), y_k)) : k < t \), \hspace{1cm} (3b)

even without having the ability to access \( \{X_k, Y_k\} \) for \( k < t \). Failing to satisfy the conditions in (3) means a deterioration of performance on previous tasks, which is often referred to as catastrophic forgetting.

Modular Networks

Layer-wise modularization methods aim to assign each unit \( u_d^i \) (in layer \( d \)) to a group \( g_k \). As a result, the groupings \( g_d^1, \ldots, g_d^{K_d} \) of the \( \frac{D}{d} \)th layer’s units are created, where \( K_d \) is the number of groups. From each group \( g_d^k \), the function \( G_d^k \) can be defined as

\[
G_d^k(x; \theta) : \mathbb{R}^p \rightarrow \mathbb{R}^{|g_d^k|}, G_d^k(x; \theta) = [a_d^j(x)| u_d^j \in g_d^k] \cdot (4)
\]
General Approach

In the following, we describe a general method that exploits the relation between two tasks given a subset of parameters (module) of the network. This method lies in the spectrum between looking at every parameter in its singularity (EWC) and taking all parameters at once (GEM). Our method’s essence comes from observing that tasks with similar representations are more prone to overwrite or negatively affect each other. This observation has been recently discovered by Ramasesh et al. (Ramasesh, Dyer, and Raghu 2020).

At first, we introduce a measure of similarity (relatedness) between tasks’ representations. Given the kth module in the dth layer, $g^k_d$, the tasks $T_1$ and $T_t$, and their underlying sampling probability distributions, $P_1$ and $P_t$, the estimate of their divergence conditioned on $g^k_d$ is given by $D(P_1(y|x; g^k_d)||P_t(y|x; g^k_d)) = D(P_1(y|G^k(x)); P_t(y|G^k(x)))$. In this work, we measure the divergence from $P_1(y|G^k(x))$ to $P_t(y|G^k(x))$ with the recently suggested Bregman-correntropy conditional divergence (Yu et al. 2020), which avoids the density estimation of data and is more statistically powerful than the classical KL. For brevity, we give its definition as follows.

**Definition 1.** The asymmetric conditional discrepancy (Yu et al. 2020) between the two conditional probability distributions $P_A(y|x)$ and $P_B(y|x)$ is defined as the quantity:

$$
D_{\phi,B}(P_A(y|x)||P_B(y|x)) = D_{\phi,B}(\sigma_{xy}||\rho_{xy}) - D_{\phi,B}(\sigma_{x}||\rho_{x}),
$$

(5)

where $\sigma_{xy}, \rho_{xy} \in S^{p+1}_+$ denote positive semidefinite matrices characterizing the joint probability distributions $P_A(x,y)$ and $P_B(x,y)$. Similarly, $\sigma_x, \rho_x \in S^+_n$ characterize the marginal distributions $P_A(x)$ and $P_B(x)$. $D_{\phi,B}$ is the Bregman matrix divergence (Kulis, Sustik, and Dhillon 2009) $D_{\phi,B}(\sigma||\rho) = \phi(\sigma) - \phi(\rho) - \text{tr}((\nabla \phi(\rho))^T(\sigma - \rho))$ between the two positive semidefinite matrices $\sigma, \rho \in R^{n \times n}$.

where $\phi : R^{n \times n} \rightarrow R$ is a strictly convex and differentiable function. The Bregman divergence represents a class of divergence functions where the von Neumann $D_{vN}$ and the LogDet $D_{Log}$ divergences can be instantiated based on the choice of $\phi$. One realization of $\sigma$ and $\rho$ could be the covariance matrix or the centered correntropy matrix (Chen et al. 2017). The symmetric conditional discrepancy is defined as:

$$
D_{\phi,B}(P_A(y|x); P_B(y|x)) = \frac{1}{2} (D_{\phi,B}(P_A(y|x)||P_B(y|x)) + D_{\phi,B}(P_B(y|x)||P_A(y|x))).
$$

(6)

For simplicity, we omit the subscripts of the Bregman matrix divergence $D_{\phi,B}$ in the remainder of the paper. The measure of relatedness between tasks $T_1$ and $T_t$ given the module $g^k_d$ would be inversely proportional to the symmetric discrepancy $D(P_1(y|G^k(x)); P_t(y|G^k(x)))$, as defined in (6). The general approach, depicted in Figure 1, starts by learning the model’s parameters on the first task $T_1$ and then finding the modular groups for each network’s layer. The choice of modularization method, in our approach, can be seen as a utility to be employed without fixing the selection of it. Having defined (or induced) modules, we compute for the first task, $T_1$, and each following task, $T_t$, the covariance matrices $\sigma_{G^k(x)y}$ and $\sigma_{G^k(x)}$ of $P_1(G^k(x), y)$ and $P_1(G^k(x))$ for each module $g^k_d$, needless to say that this computation can be efficiently performed in one pass.

The computed covariances allow us to estimate $D(P_1(y|G^k(x)); P_t(y|G^k(x)))$ for each $j < t$ for each $G^k_d$: the estimation can be theoretically re-computed after each sample or each batch (depending on the available

1Setting $\phi(\sigma) = \text{tr}(\sigma \log \sigma - \sigma)$ gives the von Neumann divergence (Nielsen and Chuan 2011): $D_{vN}(\sigma||\rho) = \text{tr}(\sigma \log \sigma - \sigma \log \rho - \rho + \rho \log \sigma)$, for all $\sigma, \rho \in R^{n \times n}$. Setting $\phi(\sigma) = - \log \det \sigma$ gives the LogDet divergence: $D_{Log}(\sigma||\rho) = \sum_{i,j} (\sigma_{ij} - \rho_{ij})^2 - \sum_i \log (\sigma_{ii}) - n.$
computation resource). The estimated divergence is the core element used in each training step, depending on the training procedure and objective. In the next sections, we present two realizations of how the technique outlined here can extend state-of-the-art methods to lessen catastrophic forgetting.

Regularization-Based Continual Learning

Elastic Weight Consolidation

Kirkpatrick et al. argue, in EWC (Kirkpatrick et al. 2017), from a Bayesian point of view that the log-posterior probability of the parametrization $\theta$, after observing two consequent tasks $T_A$ and $T_B$, can be decomposed into the log-likelihood of the task $T_B$ given the current network and the log-prior $\log p(\theta | T_A)$ (which is the same as the log-posterior given the previous task $T_A$), i.e., $\log p(\theta | T_A, T_B) = \log p(T_B | \theta) + \log p(\theta | T_A) - \log p(T_B | T_A)$. Using Laplace approximation, the log-posterior distribution $\log p(\theta | T_A)$ is approximated by a Gaussian distribution with mean $\theta_{A,i}^*$, and the inverse of the Hessian of the negative log-likelihood $-\log p(\theta | T_A)$ gives the variance. This is further simplified by taking the precision matrix as the diagonal Fisher information matrix $F_B$. As a result, the loss function is re-written as $L(\theta) = L_B(\theta) + \sum_{k \in B} r_{k,d}^A \sum_{\theta_{i,k} \in g_{k}^d} (\theta_{i,k} - \theta_{*,A,i})^2$, with $L_B(\theta)$ being the loss for task $T_B$, and $\lambda$ is the importance of the previous task.

Even though EWC is always categorized as a regularization-based method, it does depend on a replay buffer for computing FIM on previous tasks. This fact makes it partially qualify as a member of the other family of methods discussed below.

Modular EWC

We propose a modularization-based objective that considers the divergence between the tasks (more precisely, between the modular latent representations of the tasks) given the modular slicing of the network. In this objective, the changes to parameters belonging to the same module are regularized together into account (i) their relatedness to the different tasks (through the divergence estimation), and (ii) the parameter’s interdependence through the modularization step. Taking these two aspects into consideration, we redefine EWC’s objective:

$$L(\theta) = L_B(\theta) + \sum_{T_A \in \mathcal{T} \setminus \{T_B\}} \sum_{k \in T_B} r_{k,d}^A \sum_{\theta_{i,k} \in g_{k}^d} (\theta_{i,k} - \theta_{*,A,i})^2,$$

$$r_{k,d}^A = \frac{1}{2} \frac{\lambda}{Z^2} \exp(-D(p_{T_A}(\theta | x; g_{k}^d)) : p_{T_B}(\theta | x; g_{k}^d)), \tag{8}$$

where the first sum in (7) iterates over the tasks $T_A \in \mathcal{T} \setminus \{T_B\}$, the second sum is over the groups $g_{k}^d$, and the third sum iterates over the parameters concerning units in the group $g_{k}^d$. Equation (8) computes the relatedness $r_{k,d}^A$ between the representations of $T_A$ and $T_B$ given the group $g_{k}^d$; this normalized relatedness takes the form of the softmax of the negative divergence with $Z$ being the normalization term. In this EWC extension, we use $d \in \{2, \ldots, D - 1\}$, which leaves the parameters and bias of the first layer without assigned groups; therefore, for these parameters, we keep the Fisher index as a weight for the change penalty.

The motivation behind employing modularization here is that regularization becomes over groups of parameters and, hence, taking their correlations into account, unlike EWC, which takes only the diagonal of the Fisher information matrix that otherwise would be computationally expensive. Moreover, the relatedness (the negative discrepancy defined in (5)) can be efficiently computed without requiring to estimate the probability distribution in high-dimensional spaces.

Special Relation to EWC and Fisher Information

Theorem 1 shows that the J-divergence (Jeffreys 1946) is identical to the integral of the Fisher information of the exponential twist density $p_t(y|x)$ (Bucklew 1990) of the conditional distributions $p_{0_t}(y|x)$ and $p_{0_1}(y|x)$.

Based on this equivalence (proportional by a factor of $1/2$) between the J-divergence, $D_J(p_{0_t}(y|x), p_{0_1}(y|x))$, and the symmetric conditional discrepancy (when the function $\ell$ is the negative entropy), we reach the following result: The measure $\Delta_\theta$ in a regularization-based method such as EWC is equivalent to integrating over the Fisher information along the geodesic between the previous task distribution $p_{0_t}(y|x)$ and the current task distribution $p_{0_1}(y|x)$. Whereas, EWC takes only the first-order approximation of the distribution’s distance (the diagonal of Fisher matrix); this approximation is only valid in $\theta_0$’s neighborhood, i.e., $\Delta_\theta = \theta_1 - \theta_0 \rightarrow 0$.

**Theorem 2.** The J-divergence (Jeffreys 1946), $D_J(p_{0_t}(y|x), p_{0_1}(y|x))$, between $p_{0_t}(y|x)$ and $p_{0_1}(y|x)$, is the integral of the Fisher information $F_{\theta_t}$ defined on their exponential twist density $p_t(y|x)$ (Bucklew 1990), i.e.,

$$D_J(p_{0_t}(y|x), p_{0_1}(y|x)) = \int_0^1 F_{\theta_t} dt, \tag{9}$$

where $D_J(p_{0_t}(y|x), p_{0_1}(y|x)) = \int_{T_{K1}}(p_{0_t}(y|x)||p_{0_1}(y|x)) + \int_{T_{K2}}(p_{0_1}(y|x)||p_{0_t}(y|x)).$

$$F_{\theta_t} = \int_y \left(\frac{d \ln p_t(y|x)}{dt}\right)^2 p_t(y|x) dy, \ p_t(y|x) = p_{\theta_0}^{-1}(y|x)p_{0_t}(y|x)Z_t^{-1} \text{ and } Z_t = \int p_{\theta_0}^{-1}(y|x)p_{0_t}(y|x)dy.$$

The proof and a more comprehensive theoretical comparison with EWC can be seen in the appendix.

Rehearsal-Based Continual Learning

Gradient Episodic Memory

GEM (Lopez-Paz and Ranzato 2017) is a rehearsal-based continual learning method with an episodic memory $M$ storing a subset of the observed examples. For a total number of $N$ tasks, for each task $T_i$, the set of examples $M_i$ is preserved where $|M_i| = |M|/N$. GEM’s main aspect is constraining the loss on the episodic memory to decrease while updating the network’s parameters on the new task $T_i$. This is achieved by adding the decrease of the loss, $l(f(;\theta), M_k) = \frac{1}{|M|} \sum_{(x,y) \in M_k} l(f(x;\theta), y)$, on $M$ as a constraint in the search for parameters after observing the
example \((x, y)\) from the current task \(T_t\):
\[
\arg \min_\theta l(f(x; \theta), y) \tag{10a}
\]
s.t. \(l(f(\cdot; \theta), M_k) \leq l(f^{t-1}(\cdot; \theta), M_k) : k < t \tag{10b}\)
where \(f^{t-1}(\cdot; \theta)\) is the found parameterization after learning the previous \(t - 1\) tasks. Solving problem \((10)\) can be done efficiently by inferring an increase in the loss from the angle between the gradients of the loss before and after the update, which we refer to as \(q_k\) and \(q\), respectively. If all these constraints (a constraint for each previous task \(T_k\)) are satisfied, then the episodic memories’ losses should not increase. However, when one of these constraints is violated, the authors propose to project the gradient \(q\) to the closest gradient \(\tilde{q}\) in squared \(L_2\) norm, i.e., solving the following problem:
\[
\arg \min_\tilde{q} \frac{1}{2} \|q - \tilde{q}\|_2^2 \tag{11a}
\]
subject to \(\langle \tilde{q}, q_k \rangle \geq 0 \text{ for } k < t \) . \tag{11b}
Problem \((11)\) has the primal quadratic program:
\[
\arg \min_z \frac{1}{2} z^\top z - q^\top z + \frac{1}{2} q^\top q \tag{12a}
\]
subject to \(Rz \geq 0 \) , \tag{12b}
where \(R\) is the matrix of the negative gradients on all previous \(t - 1\) tasks computed on the episodic memories \(M_k\), \(R = -(q_1; \ldots; q_{t-1})\). Instead of solving the primal problem \((12)\) whose number of variables could be in millions (the number of the network’s parameters \(|\theta| = |\Theta|\)), the following dual problem is defined
\[
\arg \min_{R} \frac{1}{2} v^\top RR^\top v + q^\top R^\top v \tag{13a}
\]
subject to \(v \geq 0 \) . \tag{13b}
Upon finding \(v\), the projected gradient is computed as \(\tilde{q} = R^\top v + q\).

**Modular GEM**

Our proposed rethinking of GEM consists of two main aspects: (i) the modular partitioning of the units of each of the network’s layers, and (ii) the discrepancy estimation of each task’s estimation projected in a group. The first aspect concerns the creation of the groups \(g_1^d, \ldots, g_{K_d}^d\) for each layer \(d \in \{2, \ldots, L - 1\}\), and the second aspect leads to the computation of the discrepancy \(v_k^d = D(P_t(y|G_k^d(x; \theta)) : P_k(y|G_k^d(x; \theta)))\) between task \(T_t\) and each previous task \(T_{t'}(k < t)\) given the group \(g_k^d\), see the definition of the discrepancy in Eq \((6)\).

The first part, grouping, allows us to slice the gradients \(q\) of problem \((11)\) into \(q_1\) for the first layer’s parameters, and \(q_i^d\) for the gradients for each group \(g_i^d\) in each layer \(d \in \{2, \ldots, L\}\), since each group \(g_i^d\) concerns the set of parameters \(\theta_i^d\) \(= \{\omega_{ij}, b_{j+1}^d, u_i^d \in g_i^d \text{ and for each } j\}\). Similarly, the gradient projection \(\tilde{q}_i\), that is searched for, becomes \(\tilde{q}_1\) and \(\tilde{q}_i^d\) for each group \(g_i^d\). This formulation allows us to change the constraints such that the inner product is computed on the group-wise gradients and not on all parameters at once. Therefore, we formulate the new problem
\[
\arg \min_{q} \frac{1}{2} \|q - \tilde{q}\|_2^2 \tag{14a}
\]
subject to \(\langle \tilde{q}_i^d, (q_i^d)_k \rangle \geq \langle h_i^d, (q_i^d)_k \rangle \text{ for each } (g_i^d)_k, k < t \) \tag{14b}
subject to \(\langle \tilde{q}_i, (q_i)_k \rangle \geq 0 \text{ for } k < t \) , \tag{14c}
where \((h_i^d)_k\) is proportional to the inverse of \(exp(-r_i^d)_k\) and normalized over the seen tasks \(T_k (k < t)\). In other words, for a group that establishes a strong relation between the current and the previous task, the angle between its gradients \(q_i^d\) and \((q_i^d)_k\) should be smaller than that when such a relation is absent. The primal problem of the quadratic program solving \((14)\) becomes: \(\arg \min_z \frac{1}{2} z^\top z - q^\top z + \frac{1}{2} q^\top q\) subject to \(Rz \geq 0 \). Finally, we solve the dual problem
\[
\arg \min_{v} \frac{1}{2} v^\top RR^\top v + q^\top R^\top v \tag{15a}
\]
subject to \(v \geq 0 \) . \tag{15b}
As in \((13)\), the new projected gradient is given by computed as \(\tilde{q} = R^\top v + q\).

**Three Ways of Modularization**

In this work, we suggest three ways to approach modularization at the neuron-level with different intuitions.

(i) Information-Theoretic modularization: Motivated by the general goal of neuron-level modularization, we propose identifying groups of neurons that minimize the pairwise group dependence. One way to measure group dependence is to use the recently proposed matrix-based Rényi’s \(\alpha\)-order mutual information \(\text{Giraldo, Rao, and Principe 2014}\) to measure the pairwise independence between two groups of neurons.

(ii) Log-likelihood modularization: This type of modularization searches for the grouping that maximizes the likelihood of the neurons’ group assignment. Watanabe et. al \((\text{Chihiro Watanabe and Kashino 2018})\) propose a modular decomposition of trained neural networks into a set of independent sub-networks.

(iii) Random modularization: One naive way to achieve modularization is by creating a random grouping where the \(j\)-th unit in the layer \(d\), \(u_i^d\), is assigned to the group \(g_i^d\) randomly.

The formal definitions and description of these methods are in the supplementary material.

**Experiments**

**Setting: Online Continual Learning**

The following empirical evaluations follow the online setting described in \((\text{Riemer et al. 2018})\), where each sample of each task is observed in a single pass sequence.

As for the neural network architecture, we use a single head fully-connected neural network with two hidden layers, each with 100 neurons, a \(28 \times 28\) input layer, and an output layer with a single head with 10 units. The architecture is similar to the one used in \((\text{Lopez-Paz and Ranzato 2017})\). The hidden layers employ the ReLu activation function, and SGD is used to minimize the softmax cross-entropy on the online training data.
Datasets and Performance Measures We use for the evaluation the following data sets: (i) MNIST Permutations (mnistP) (Kirkpatrick et al. 2017), (ii) MNIST Rotations (mnistR) (Lopez-Paz and Ranzato 2017), (iii) Permuted Fashion-MNIST (fashionP) (Xiao, Rasul, and Vollgraf 2017), and (iv) Permuted notMNIST (notmnistP) (Buitelaar 2011). All these data sets contain images of the same size.

To measure the learning-ability and resistance to forgetting, we compute the often used performance measures: (i) Learning accuracy (LA) the average accuracy on each task after learning it. (ii) Retained accuracy (RA) is the averaged performance on all tasks after observing the last one. (iii) Backward transfer (BT) represents the loss in performance due to forgetting, i.e., the difference between LA and RA (Chaudhry et al. 2018a).

Comparison Protocol In our framework, we claim to improve the performance of continual learning methods by showing how modular relatedness to previous tasks can be exploited to prevent catastrophic forgetting. To verify this claim, we evaluate our extensions of methods taken from two different families of continual learning. The first extension ModEWC modifies EWC as a regularization-based method. The second method is ModGEM, which alters GEM as a representative of rehearsal-based methods. We compare the performance of our two proposed methods against that of their original methods. To ensure a fair comparison, we start with a grid-based hyperparameter search for each method on each data set; the search protocol and the found parameters are reported in the appendix.

In all the following experiments, we employ a stream of ten tasks, where a sequence of only 1000 samples is observed from each task. Every time an evaluation is performed on a task, it is done on its test data of 10,000 samples.

As for the modularization method, we implement the community detection-based modularization (Chhihiro Watanabe and Kashino 2018) using ten iterations and then choosing the detected modules with the highest log-likelihood. We also fix the number of groups to be $K_d = 20$ for ModEWC and $K_d = 5$ for ModGEM. We use this setting in the forthcoming experiments unless stated differently.

Comparing Modular Relatedness versus EWC and GEM In this experiment, we compare at first ModEWC versus EWC, under the aforementioned online setting with the restricted memory budget of ten samples per task. Table 1 shows that our proposed method improves retained accuracy by 20% on the fashionP dataset, and around 6% and 4% on notmnistP and mnistP, respectively. ModEWC also performs better than EWC on mnistR without a significant difference. In terms of learning accuracy, both methods perform comparatively similar on notmnistP and mnistR, whereas, ModEWC shows substantial improvement of the learning accuracy on fashionP and mnistP. This is a clear sign of a missing forward transfer that EWC fails to achieve under the circumstances of limited memory compared to ModEWC. The gain in both LA and RA that our modification causes to EWC is accompanied by a better backward transfer (BT) on all data sets.

Table 1: Performance comparison between ModEWC and EWC. The numbers in parentheses are the standard errors (SE) of the means in the former row.

| Data   | EWC          | ModEWC        |
|--------|--------------|---------------|
|        | RA | LA  | BT     | RA | LA  | BT     |
| notmnistP | 68.65 | 80.98 | -12.33 | (0.28) | (0.13) | (0.21) |
| fashionP | 42.24 | 56.24 | -14.00 | (2.14) | (1.4) | (0.81) |
| mnistR  | 62.10 | 85.56 | -23.46 | (0.26) | (0.07) | (0.27) |
| mnistP  | 66.11 | 76.95 | -10.85 | (1.9) | (0.65) | (1.25) |

Table 2: Performance comparison between ModGEM and GEM. The numbers in parentheses are the standard errors (SE) of the means in the former row.

| Data   | GEM          | ModGEM        |
|--------|--------------|---------------|
|        | RA | LA  | BT     | RA | LA  | BT     |
| notmnistP | 64.20 | 78.60 | -14.36 | (0.4) | (0.1) | (0.4) |
| fashionP | 56.86 | 67.40 | -10.54 | (0.18) | (0.1) | (0.17) |
| mnistR  | 75.75 | 85.05 | -9.30  | (0.2) | (0.07) | (0.21) |
| mnistP  | 64.40 | 80.38 | -15.97 | (0.25) | (0.07) | (0.26) |

In the second experiment, we compare ModGEM versus GEM using the same setting used in the previous experiment, online and a memory budget of ten samples per task. Table also shows that ModGEM outperforms GEM on each of notmnistP, mnistP and fashionP with margins of 4%, 4%, and 2% retained accuracy, respectively. The only exception here is mnistR, where GEM is only 1.6% better than ModGEM. Both methods have relatively the same learning accuracy, which results in a better backward performance achieved by ModGEM.

Modular-Relatedness under Different Memory Constraints In this experiment, we evaluate the effect of different memory budgets on RA of EWC, GEM, and our proposed methods. The experiments vary the memory size from the set $\{5, 10, 15, 20\}$. Figure depicts the RA curves versus the memory budget (x-axis). The upper row of figures shows how the curves of ModEWC, most of the time, dominate those of EWC and with a large margin. There is, sometimes, the trend for EWC to improve when more memory is granted, and its curve does meet with that of ModEWC on the mnistP data. This result indeed confirms our intuition that the modular-relatedness plays the role of an augmented memory when memory budget is scarce, moreover, ModEWC seems to offer an empirical upper bound of what EWC can achieve, as confirmed on notmnistP, fashion and PmnistP. Interestingly, no clear pattern can be deduced from Figure on mnistR since the difference between the two
Figure 2: Retained accuracy performance curves for ModEWC, ModGEM, and their original methods EWC and GEM. The curves are computed when the memory budget is taken from the set \{5, 10, 15, 20\}.

Table 3: Retained accuracy for ModEWC when a different number of groups is used. Five iterations are used. The numbers in parentheses are the standard error.

| Data    | 5     | 10    | 15    | 20    |
|---------|-------|-------|-------|-------|
| notmnistP | 71.0(0.65) | 71.27(0.65) | 71.01(0.61) | 71.45(0.66) |
| fashionP | 63.16(0.55) | 63.2(0.45) | 63.62(0.48) | 64.31(0.57) |
| mnistR   | 61.43(0.36) | 61.02(0.5)  | 61.63(0.35) | 61.85(0.32) |
| mnistP   | 71.91(0.65) | 72.34(0.45) | 72.08(0.61) | 71.78(0.62) |

curves does not exceed 1%.

The figures in the lower row of Figure 2 show similar results when comparing ModGEM with GEM. Again and on all data sets, except for mnistR, ModGEM’s performance presents an upper bound of what GEM can achieve when the budget for the memory increases.

Sensitivity Analysis on the Number of Groups In this experiment, we present a sensitivity analysis on the number of groups generated in ModEWC by trying the different numbers of groups, i.e., \(K_d \in \{5, 10, 15, 20\}\) for all \(d\). Table 3 depicts an almost monotone increasing performance with a larger number of groups. However, the slope of this trend is very small, which can be interpreted as insensitivity of the proposed method towards \(K_d\).

On the Effect of the Modularization Method To study modularization’s effects, we perform a comparison of how our EWC’s extension acts when the initial induced modules are created randomly instead of applying the log-likelihood method (Chihiro Watanabe and Kashino 2018). Moreover, to gain more insights, we design experiments on fashionP, where all three modularization methods are employed, and the observed likelihood and independence are plotted for the hidden layers. The results and the discussion of this experiment are in the supplementary material.

Conclusion

In this work, we present a technique that exploits tasks’ modular relatedness based on the discrepancy of their latent representations. This method’s essence is the automatic discovery of groups of neurons in each layer, and the adaptation of the learning process to account for the tasks’ relatedness given these modules. This technique has been applied to extend two of the state-of-the-art methods, EWC and GEM; the technique is also equipped with respectively three ways of modularization. A significant performance gain has been observed on the learning and retained accuracy using these extensions compared to EWC and GEM. We abstain from claiming that our method is beneficial to every work descending from EWC or GEM. However, this might be suspected to do the inherited property of not ignoring the modularity. Future work is twofold. First, we aim at studying the applicability of our technique in a broader spectrum of continual learning families. Second, we are investigating the use of this technique in meta-learning, where task relatedness plays an essential role in choosing the right prior. Here modular-relatedness would help in choosing the right prior over the modular sub-networks.
References

Aljundi, R.; Chakravarty, P.; and Tuytelaars, T. 2017. Expert gate: Lifelong learning with a network of experts. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3366–3375.

Azam, F. 2000. Biologically inspired modular neural networks. Ph.D. thesis, Virginia Tech.

Bhatia, R. 2006. Infinitely divisible matrices. The American Mathematical Monthly 113(3): 221–235.

Bucklew, J. A. 1990. Large deviation techniques in decision, simulation, and estimation. Wiley New York.

Bulatov, Y. 2011. Machine Learning, etc: notMNIST dataset. URL http://yaroslavvb.blogspot.com/2011/09/notmnist-dataset.html.

Chaudhry, A.; Dokania, P. K.; Ajanthan, T.; and Torr, P. H. 2018a. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In Proceedings of the European Conference on Computer Vision (ECCV), 532–547.

Chaudhry, A.; Ranzato, M.; Rohrbach, M.; and Elhoseiny, M. 2018b. Efficient lifelong learning with a-gem. arXiv preprint arXiv:1812.00420.

Chen, B.; Xing, L.; Xu, B.; Zhao, H.; Zheng, N.; and Principe, J. C. 2017. Kernel risk-sensitive loss: definition, properties and application to robust adaptive filtering. IEEE Transactions on Signal Processing 65(11): 2888–2901.

Chen, Z.; and Liu, B. 2018. Lifelong machine learning. Synthesis Lectures on Artificial Intelligence and Machine Learning 12(3): 1–207.

Chihiro Watanabe, K. H.; and Kashino, K. 2018. Modular representation of layered neural networks. Neural Networks 97: 62–73.

Clune, J.; Mouret, J.-B.; and Lipson, H. 2013. The evolutionary origins of modularity. Proceedings of the Royal Society b: Biological sciences 280(1755): 20122863.

Dabak, A. G.; and Johnson, D. H. 2002. Relations between Kullback-Leibler distance and Fisher information.

De Lange, M.; Aljundi, R.; Masana, M.; Parisot, S.; Jia, X.; Leonardis, A.; Slabaugh, G.; and Tuytelaars, T. 2019. Continual learning: A comparative study on how to defy forgetting in classification tasks. arXiv preprint arXiv:1909.08383.

Giraldo, L. G. S.; Rao, M.; and Principe, J. C. 2014. Measures of entropy from data using infinitely divisible kernels. IEEE Transactions on Information Theory 61(1): 535–548.

Jaderberg, M.; Czarnecki, W. M.; Osindero, S.; Vinyals, O.; Graves, A.; Silver, D.; and Kavukcuoglu, K. 2017. Decoupled neural interfaces using synthetic gradients. In International Conference on Machine Learning, 1627–1635. PMLR.

Jeffreys, H. 1946. An invariant form for the prior probability in estimation problems. Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences 186(1007): 453–461.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. 2017. Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences 114(13): 3521–3526.

Kulis, B.; Sustik, M. A.; and Dhillon, I. S. 2009. Low-rank kernel learning with Bregman matrix divergences. JMLR 10(Feb): 341–376.

Li, Z.; and Hoiem, D. 2017. Learning without forgetting. IEEE transactions on pattern analysis and machine intelligence 40(12): 2935–2947.

Liu, X.; Masana, M.; Herranz, L.; Van de Weijer, J.; Lopez, A. M.; and Bagdanov, A. D. 2018. Rotate your networks: Better weight consolidation and less catastrophic forgetting. In 2018 24th International Conference on Pattern Recognition (ICPR), 2262–2268. IEEE.

Lopez-Paz, D.; and Ranzato, M. 2017. Gradient episodic memory for continual learning. In Advances in Neural Information Processing Systems, 6467–6476.

Meunier, D.; Lambiote, R.; and Bullmore, E. T. 2010. Modular and hierarchically modular organization of brain networks. Frontiers in neuroscience 4: 200.

Nielsen, M. A.; and Chuang, I. L. 2011. Quantum Computation and Quantum Information. Cambridge University Press, 10th edition.

Parisi, G. I.; Kemker, R.; Part, J. L.; Kanan, C.; and Wernter, S. 2019. Continual lifelong learning with neural networks: A review. Neural Networks.

Ramasesh, V. V.; Dyer, E.; and Raghu, M. 2020. Anatomy of Catastrophic Forgetting: Hidden Representations and Task Semantics. arXiv preprint arXiv:2007.07400.

Rebuffi, S.-A.; Kolesnikov, A.; Sperl, G.; and Lampert, C. H. 2017. icarl: Incremental classifier and representation learning. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2001–2010.

Riemer, M.; Cases, I.; Ajemian, R.; Liu, M.; Rish, I.; Tu, Y.; and Tesauro, G. 2018. Learning to learn without forgetting by maximizing transfer and minimizing interference. arXiv preprint arXiv:1810.11910.

Robins, A. 1995. Catastrophic forgetting, rehearsal and pseudorehearsal. Connection Science 7(2): 123–146.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of Go with deep neural networks and tree search. nature 529(7587): 484–489.

Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. 2017. Mastering the game of go without human knowledge. nature 550(7676): 354–359.

Silver, D. L.; and Mercer, R. E. 2002. The task rehearsal method of life-long learning: Overcoming impoverished data. In Conference of the Canadian Society for Computational Studies of Intelligence, 90–101. Springer.
Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms.

Yoon, J.; Yang, E.; Lee, J.; and Hwang, S. J. 2017. Life-long learning with dynamically expandable networks. arXiv preprint arXiv:1708.01547.

Yu, S.; Shaker, A.; Alesiani, F.; and Principe, J. C. 2020. Measuring the Discrepancy between Conditional Distributions: Methods, Properties and Applications. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, 2777–2784.

Zenke, F.; Poole, B.; and Ganguli, S. 2017. Continual learning through synaptic intelligence. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, 3987–3995. JMLR. org.
Special Relation to EWC and Fisher Information

Chaudhry et. al (Chaudhry et al. 2018a) show that the KL-divergence \( D_{KL}(p_0(y|x)||p_{0+\Delta \theta}(y|x)) \) between conditional likelihoods of two neural networks parametrized by \( \theta \) and \( \theta + \Delta \theta \) can be approximated as \( D_{KL}(p_0(y|x)||p_{0+\Delta \theta}(y|x)) \approx \frac{1}{2} \Delta \theta^T F_0 \Delta \theta \) where \( F_0 \) is the Fisher information matrix at \( \theta \), assuming that \( \Delta \theta \to 0 \), see the proof in Appendix A1 of (Chaudhry et al. 2018a). Since it is infeasible to compute \( F_0 \) when the number of parameters is in the order of millions, parameters are assumed to be independent and only the diagonal of \( F_0 \) is computed, as a result, the divergence becomes \( D_{KL}(p_0(y|x)||p_{0+\Delta \theta}(y|x)) \approx \sum_i \frac{1}{2} F_0(\theta_i - \theta_i^*)^2 \) which collides with the regularization term of EWC, i.e., the second term in

\[
\mathcal{L}(\theta) = \mathcal{L}_B(\theta) + \sum_i \frac{\lambda}{2} F_0(\theta_i - \theta_i^*)^2.
\]

In the following, we show a similar analysis that connects ModEWC with EWC from an information-theoretic point of view.

EWC uses the Fisher information to weight parameters based on the importance of the previous parameters. Whereas in this work, we consider the joint influence of previous and the current task’s data distributions (in terms of their direct measure of relatedness) on the learned parameters, thus, measuring the effort of moving the current induced distribution from the previous task to the current. We first state the following theorem and its proof.

**Theorem 1.** The J-divergence (Jeffreys 1946),
\[
D_J(p_0(y|x), p_0(y|x)) = \int_0^1 F_{pi} dt,
\]
where
\[
D_J(p_0(y|x), p_0(y|x)) = D_{KL}(p_0(y|x)||p_{0i}(y|x)) + D_{KL}(p_0(y|x)||p_{0i}(y|x)) - t D_{KL}(p_0(y|x)||p_{0i}(y|x)).
\]

Proof of Theorem [7] Consider the two conditional distributions \( p_{0i}, p_{0i} \), their exponential twist density \( p_t(y|x) \) (Bucklew 1990) is defined as
\[
p_t(y|x) = \frac{p_{0i}(y|x)^{1-t} p_{0i}(y|x)^{t}}{Z_t} \quad (18)
\]
where \( Z_t \) is the normalization function of \( t \). The parameter \( t \) moves the probability density function \( p_t \) along the manifold of densities between \( p_{0i}, p_{0i} \). Following (Dabak and Johnson 2002), we establish the connection between our symmetric discrepancy measure (6) and the Fisher information matrix \( F_{pi} \) of \( p_t(y|x) \),

\[
F_{pi} = \int_y \left( \frac{d \ln p_t(y|x)}{dt} \right) p_t(y|x)dy = \int_y \ln \frac{p_{0i}(y|x)}{p_{0i}(y|x)} p_t(y|x)dy - \left( \int_y \ln \frac{p_{0i}(y|x)}{p_{0i}(y|x)} p_t(y|x)dy \right)^2 \quad (19)
\]

since
\[
\frac{d \ln p_t(y|x)}{dt} = \ln \frac{p_{0i}(y|x)}{p_{0i}(y|x)} - \left( \int_y \ln \frac{p_{0i}(y|x)}{p_{0i}(y|x)} p_t(y|x)dy \right).
\]

Kullback-Leibler divergence \( D_{KL} \) and its first derivative, between \( p_t(y|x) \) and \( p_{0i}(y|x) \), can be written as

\[
D_{KL}(p_t(y|x)||p_{0i}(y|x)) = \int_y \ln \frac{p_{0i}(y|x)}{p_{0i}(y|x)} p_t(y|x)dy \quad (20)
\]

\[
= \int_y t \ln \frac{p_{0i}(y|x)}{p_{0i}(y|x)} p_t(y|x)dy - \ln Z_t \quad (21)
\]

\[
dD_{KL}(p_t(y|x)||p_{0i}(y|x)) = \int_y \ln \frac{p_{0i}(y|x)}{p_{0i}(y|x)} dp_t(y|x) dy + \frac{\ln Z_t}{dt} \quad (22)
\]

Comparing (21) and (22), we notice that
\[
\int_y \ln \frac{p_{0i}(y|x)}{p_{0i}(y|x)} dp_t(y|x) = t F_{pi},
\]

knowing that \( \frac{d \ln Z_t}{dt} = \int_y \ln \frac{p_{0i}(y|x)}{p_{0i}(y|x)} p_t(y|x)dy \).
If we use the alternative parametrization \(u = 1 - t\) we can similarly derive
\[
\frac{dD_{KL}(p_\theta(y|x)||p_\theta(y|x))}{du} = u F_{p_\theta},
\]
where \(p_\theta(y|x) = p_{\theta_1}^{u}(y|x)p_{\theta_1}^{1-u}(y|x)\) and \(Z_u = \int p_{\theta_1}^{u}(y|x)p_{\theta_1}^{1-u}(y|x)dy\). If we consider the J-divergence (Jeffreys 1946) on \(p_{\theta_0}(y|x), p_{\theta_1}(y|x)\)
\[
D_{KL}(p_{\theta_1}(y|x)||p_{\theta_0}(y|x)) + D_{KL}(p_{\theta_0}(y|x)||p_{\theta_1}(y|x))
\]
\[
= \int_1^0 t F_{p_\theta} dt + \int_0^1 u F_{p_\theta} du
\]
\[
= \int_1^0 t F_{p_\theta} dt + \int_0^1 (1-t)F_{p_\theta} dt = \int_0^1 F_{p_\theta} dt
\]
Thus the J-divergence is the geodesic integral of the Fisher information.

Notice the equivalence (proportional by a factor of \(1/2\)) between the J-divergence, \(D_{J}(p_{\theta_1}(y|x), p_{\theta_0}(y|x))\), and the symmetric conditional discrepancy \(\delta\), when the function \(\ell\) is the negative entropy.

We can now observe that using the diagonal Fisher matrix is equivalent to applying the first-order approximation on the previous and current parameters’ geodesic distance. Thus, employing the measure \(\delta\) in a regularization-based method such as EWC (Lee et al. 2017) is equivalent to integrating over the Fisher information along the geodesic between the previous task distribution \(p_{\theta_0}(y|x)\) and the current task distribution \(p_{\theta_1}(y|x)\), instead of taking only the first-order approximation induced by the diagonal of Fisher matrix at the previous task parameters, which is only valid in a neighbourhood of \(\Delta \theta = \theta_1 - \theta_0 \to 0\), as EWC does.

We also notice that the previous result is valid not only for a single parameter but also for a vector of parameters. Therefore, ModEWC does not approximate the integral only a parameter before and after an update \(\theta\) and \(\theta + \Delta \theta\), as EWC does.

### Hyperparameter Search

To ensure a fair comparison, we start with a grid-based hyperparameter search for each of the methods on each of the datasets using a sample of 5 tasks and 300 samples per task. For EWC, we tune the two parameters, the learning rate \(lr \in \{0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1.0\}\) and the memory strength \(ms \in \{1, 3, 10, 30, 100, 300, 1000, 3000, 10000, 30000\}\). For GEM, we search for the parameters, the learning rate \(lr \in \{0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1.0\}\) and the margin \(mg \in \{0, 0.1, 0.5, 1.0\}\). The found parameters are reported in the following:

- **EWC found hyperparameters**:
  - \(lr \in \{0.001, 0.003\} \text{ (notmnistP)}, 0.01 \text{ (mnistR, mnistP, fashionP)}, 0.03, 0.1, 0.3, 1.0 \}\)
  - \(ms \in \{1 \text{ (notmnistP)}, 3 \text{ (mnistR)}, 10, 30, 100 \text{ (mnistP, fashionP)}, 300, 1000, 3000, 10000, 30000\}\)
  - **GEM found hyperparameters**

- **lr \in \{0.001, 0.003, 0.01 \text{ (notmnistP)}, 0.03, 0.1, 0.3, 1.0 \}\)
- **ms \in \{0 \text{ (notmnistP)}, 0.0 \text{ (mnistR)}, 0.1, 0.3, 0.5, 1.0 \}\)

Without any further tuning, we adopt the same found parameter to our proposed modification, except for the memory strength, in ModEWC, that we force to be less than 10. As for the relatedness measure, we employ (6) when \(D_{F,\ell}\) is the von Neumann divergence by setting \(\varphi(\sigma) = \text{tr}(\sigma \log \sigma - \sigma)\). \(\sigma\) is the covariance matrix of each task’s latent representation at each group.

### Three Ways of Modularization

In this work, we suggest three ways to approach modularization in a neuron level with different intuitions.

- **Information-Theoretic modularization**: The information-theoretic modularization is directly motivated by the general goal of neuron-level modularization, in which we aim to identify groups of neurons such that each neuron largely depends on neurons in the same group, but is less relevant to neurons from other groups. In information theory, the mutual dependence or independence can be simply measured with mutual information (denoted by \(I\)). Formally, suppose we partition neurons at layer \(d\) into \(K_d\) groups, \(g_1^d, \ldots, g_{K_d}^d\), and in each group \(g_i^d\) there are \(n_i\) neurons such that \(\sum_{i=1}^{K_d} n_i = N_d\). Let us denote \(u_{i,k}\) as the \(i\)-th neuron in the \(k\)-th group \(g_i^d\), then our objective is to minimize the sum of pairwise mutual information between the representations formed by any two groups of neurons \((g_i^d, g_j^d) : J = \sum_{i \neq j} I(u_{i,1}, u_{i,2}, \ldots, u_{i,n_i} : u_{j,1}, u_{j,2}, \ldots, u_{j,n_j})\).

In this work, we use the recently proposed matrix-based Rényi’s \(\alpha\)-order mutual information (Giraldo, Rao, and Principe 2014) to measure the pairwise independence between two groups of neurons, as it avoids the estimation of the underlying data distributions and scales well to high-dimensional space. For simplicity, we directly give the following definition.

**Definition 2.** Let \(\kappa : \mathbb{R} \times \mathbb{R} \to \mathbb{R}\) be a real valued positive definite kernel that is also infinitely divisible (Bhattacharya 2006). Given \(\{x_i\}_{i=1}^N \subset \chi\), each \(x_i\) can be a real-valued scalar or vector, and the Gram matrix \(K\) obtained from evaluating a positive definite kernel \(\kappa\) on all pairs of exemplars, that is \(K = \kappa(x_i, x_j)\), a matrix-based analogue to Rényi’s \(\alpha\)-entropy for a normalized positive definite matrix \(A\) of size \(n \times n\), such that \(\text{tr}(A) = 1\), can be given by the following functional:

\[
H_{\alpha}(A) = \frac{1}{1 - \alpha} \log (\text{tr}(A^\alpha)) = \frac{1}{1 - \alpha} \log \left(\sum_{i=1}^n \lambda_i(A)^\alpha\right),
\]

where \(A\) is the normalized version of \(K\), i.e., \(A = K/\text{tr}(K)\), and \(\lambda_i(A)\) denotes the \(i\)-th eigenvalue of \(A\).

**Definition 3.** Given \(n\) pairs of samples \(\{x_i, y_i\}_{i=1}^n\), each sample contains two different types of measurements \(x \in \mathbb{R}^d, y \in \mathbb{R}^e\), we refer to the above formulation as the **information-theoretic modularization**.
\( \chi \) and \( y \in \Gamma \) obtained from the same realization, and the positive definite kernels \( \kappa_1: \chi \times \chi \to \mathbb{R} \) and \( \kappa_2: \Gamma \times \Gamma \to \mathbb{R} \), a matrix-based analogue to Rényi’s \( \alpha \)-order joint-entropy can be defined as:

\[
H_\alpha(A, B) = H_\alpha \left( \frac{A \circ B}{tr(A \circ B)} \right),
\]

where \( A_{ij} = \kappa_1(x_i, x_j) \), \( B_{ij} = \kappa_2(y_i, y_j) \) and \( A \circ B \) denotes the Hadamard product between the matrices \( A \) and \( B \).

Given Eqs. (31) and (32), the matrix-based Rényi’s \( \alpha \)-order MI \( I_\alpha(A;B) \) in analogy of Shannon’s MI is given by:

\[
I_\alpha(A;B) = H_\alpha(A) + H_\alpha(B) - H_\alpha(A, B).
\]

Throughout this paper, we use the radial basis function (RBF) kernel \( \kappa(x_i, x_j) = \exp(-\frac{\|x_i-x_j\|^2}{2\sigma^2}) \) to obtain the Gram matrices.

- **Log-likelihood modularization**: This type of modularization searches for the grouping that maximizes the likelihood of the neurons’ group assignment. Watanabe et al. (Chihiro Watanabe and Kashino 2018) propose a modular decomposition of trained neural networks into a set of independent sub-networks. This decomposition considers each unit’s assignment \( u_{id} \) (in layer \( d \)) to a group \( g_k \) as a latent variable, and defines the probability for each group to connect to each neuron of the previous and the following layers as a parameter. These parameters and the group assignment are found by maximizing the likelihood of observing the groups given the connections to the previous and following layers, \( d-1 \) and \( d+1 \), respectively. To this end, the expectation-maximization algorithm is iteratively employed to find the groups.

- **Random modularization**: One naive way to achieve modularization is through the creation of a random grouping where the \( j \)-th unit in the layer \( d \), \( u_{jd} \), is assigned to the group \( g_k^d \) after sampling the indicator vector \( e_i \sim \text{Multinomial}(1, 1/K_d, \ldots, 1/K_d) \) \( (e_i \in \mathbb{R}^{K_d} \) with only the \( i \)th element set to 1 and others to 0). Multinomial \((1, 1/K_d, \ldots, 1/K_d) \) is the multinomial distribution with the number of trials equal to one and a uniform probability of events \( 1/K_d \). Here, the condition \( |g_k^d| > 1 \) for all \( i \in \{1, \ldots, K_d\} \) is needed to guarantee that no group of a single unit exists.

### On the Effect of the Modularization Method

To study the modularization’s effect, we perform a comparison of how our EWC’s extension acts when the initial induced modules are created randomly instead of applying the log-likelihood modularization method (Chihiro Watanabe and Kashino 2018). It can be concluded from Table 4 that the log-likelihood method often performs better than the random grouping except on mnistP. Nonetheless, the random grouping does not seem to be badly performing, as one would intuitively suspect. To gain more insights on this issue, we design a minimal experiment on the fashionP data

| Data       | notmnistP | fashionP | mnistR | mnistP |
|------------|-----------|----------|--------|--------|
| ModEWC     | 71.45     | 64.31    | 61.85  | 71.78  |
|            | (0.66)    | (0.566)  | (0.32) | (0.624) |
| RandEWC    | 71.45     | 63.92    | 61.34  | 72.31  |
|            | (0.59)    | (0.55)   | (0.41) | (0.47)  |

where we compute the likelihoods and the mutual information (on each layer) for each of the methods (i) (Mod.IT) the information-theoretic modularization method using matrix-based Rényi’s \( \alpha \)-order mutual information criteria, as described earlier, (ii) (Mod.LLH) the log-likelihood modularization as proposed by Watanabe et al. (Chihiro Watanabe and Kashino 2018), and (iii) (Rand) the random grouping.

Each of these three methods is experimented twice, once by computing the grouping only after the first task, and once by recomputing the grouping (modules) after each task (recomp).

Figure [a] shows that all methods have decreasing log-likelihoods on the first hidden layer, and Figure [b] shows that only Mod.LLH in its both variants and the Recomp.-Mod.IT have sustained their likelihoods on the second hidden layer. On the other hand, a different observation can be made from Figures [c] and [d]; here, the independence measure for all suggested modularization methods seems to be improving, on all hidden layers, the more tasks we observe. Based on this single evidence, it is too early to make a concrete conclusion without further investigation. Nonetheless, there is the sign that our extension to EWC seems to create an imposed modularization, i.e., regardless of the method used to create the initial groups, these groups become valid modules based on mutual independence of the latent representations they produce.
Figure 3: The computed likelihood and mutual information of different grouping initialization methods for ModEWC.