PROJECTED LATENT DISTILLATION FOR DATA-AGNOSTIC CONSOLIDATION IN MULTI-AGENT CONTINUAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Many real-world applications are characterized by non-stationary distributions. In this setting, independent expert models trained on subsets of the data can benefit from each other and improve their generalization and forward transfer by sharing knowledge. In this paper, we formalize this problem as a multi-agent continual learning scenario, where agents are trained independently but they can communicate by sharing the model parameters after each learning experience. We split the learning problem into two phases: *adaptation* and *consolidation*. Adaptation is a learning phase that optimizes the current task, while consolidation prevents forgetting by combining expert models together, enabling knowledge sharing. We propose Data-Agnostic Consolidation (DAC), a novel double knowledge distillation method. The method performs distillation in the latent space via a novel Projected Latent Distillation (PLD) loss. Experimental results show state-of-the-art accuracy on SplitCIFAR100 even when a single out-of-distribution image is used as the only source of data during consolidation.

1 INTRODUCTION

Real world data is characterized by non-stationary distributions. In this setting, continual learning (Lesort et al., 2020) is necessary to mitigate catastrophic forgetting(French, 1999) of past experiences. In many applications there may even be multiple independent sources of data that cannot be integrated in a single dataset due to privacy constraints. Let us consider a reference scenario where a fleet of robots is deployed in different locations, forming a decentralized network of edge devices. Each robot is an independent agent learning from a distinct environment, possibly with limited connectivity with the others. In this application, sharing knowledge between the agents can help to improve generalization and forward transfer. At the same time, we may not be allowed to collect and share the raw data due to privacy constraints, such as data about agents' interactions with real users.

In this paper, we formalize this problem as a *multi-agent continual learning* scenario. The key novelty compared to popular continual learning scenarios (van de Ven & Tolias, 2019) is the necessity to share the knowledge between agents. Currently, the only method to exploit pretrained models is to use them to initialize a continual learning model (Hayes & Kanan, 2020; Maltoni & Lomonaco, 2019), which means that it is not possible to integrate external knowledge once training is started. Other frameworks, such as federated learning (Li et al., 2020), are designed to train a single model in a distributed way, which is a different problem from sharing knowledge between independent agents. Federated learning requires tight synchronization between the clients and a centralized server that controls the training process. As a result, it is not possible to integrate the knowledge of independent agents using popular federated learning methods or to allow each agent to train independently and in a fully decoupled fashion.

The main challenge of this scenario is the problem of knowledge consolidation in the absence of the original data. We therefore propose a novel method, called *Data-Agnostic Consolidation (DAC)*, that allows to distill knowledge from independent agents with a data-free double knowledge distillation. The key idea of the method is that each continual learning step can be split into two separate phases, *adaptation* and *consolidation*. We show that DAC learns successfully even when the original data

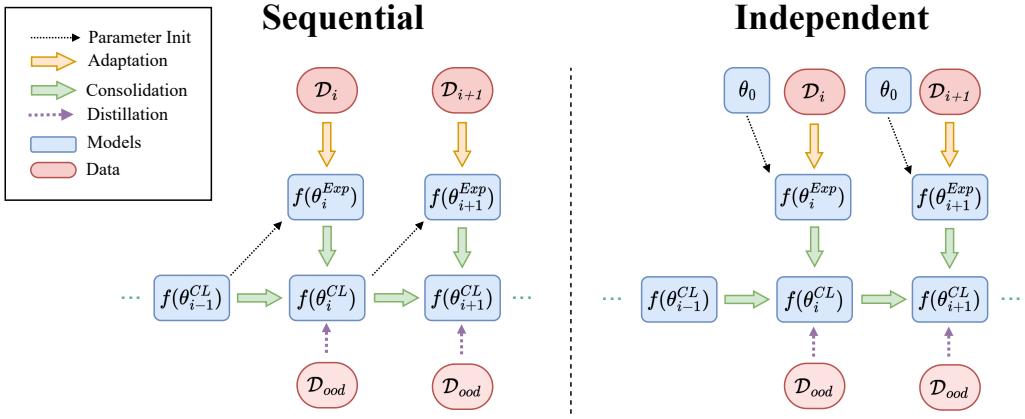


Figure 1: In the sequential setting (left), expert models (top row) are initialized with the current weights θ^{CL}_{i-1}. In the independent setting (right), expert models start from a common initialization θ_0. In both cases, the consolidated model (bottom row) is trained on external data $\mathcal{D}_{ood}$. The challenge of these scenarios is to incorporate the knowledge of the experts into the main model without access to any task data.

is used only in the adaptation phase. In fact, a very simple source of data, such as a single out-of-distribution image, is sufficient. This is possible due to the use of heavy augmentations, in line with recent advances for knowledge distillation (Beyer et al., 2022; Asano & Saeed, 2022). As a consequence, it is possible to perform the adaptation locally on-device and the consolidation using a remote server without sharing the data with the server. A problem with the double distillation is that it is not possible to perform feature distillation because the two teachers compute two different latent representations. DAC solves this problem via our novel *Projected Latent Distillation* loss. As a result, DAC can distill the output and latent space of both teachers without trading off stability or plasticity. The experimental results show that DAC allows to consolidate the knowledge from independent agents even when they are trained on different tasks and only a single out-of-distribution image is available.

The main contributions of the paper can be summarized as follows:

- a formal characterization of multi-agent continual learning. To the best of our knowledge, this is the first work to formalize continual learning in a multi-agent setting and the consolidation problem (Section 2);
- *Data-Agnostic Consolidation*, a novel strategy which performs a data-agnostic double knowledge distillation in the output and latent space via Projected Latent Distillation (Section 3);
- state-of-the-art results for task-aware SplitCIFAR100 (Table 1a, +3.9% on 10 Tasks);
- an extensive experimental analysis that shows the importance of heavy augmentations during distillation and the benefit of separating adaptation and consolidation (Section 4).

2 Multi-Agent Continual Learning

In continual learning, an agent learns from a stream of experiences $S = e_1, \dots, e_n$. In a supervised setting, each experience e_i provides a batch of samples $\mathcal{D}_i = \{\langle x_m, y_m, t_m \rangle\}$, where $x_m \in \mathbb{R}^X$ is the input, $y_m \in \mathcal{Y}_i$ the target label, and $t_m \in \mathbb{N}$ an optional task label. A trivial solution to this problem would be to train a model on the joint dataset $\mathcal{D}_J = \cup_{i=1}^{n} \mathcal{D}_i$. However, due to resource and privacy constraints, we assume that at time i we do not have access to data from previous experiences $e_j, j < i$. This popular setting is called exemplar-free (or data-free) continual learning (Smith et al., 2021; Masana et al., 2020).

A continual learning agent f_θ is a model trained sequentially on a stream S. In a *multi-agent environment* we have multiple agents trained independently on different streams. Ideally, we would

like to share the knowledge between agents via a simple communication mechanism. However, due to privacy and bandwidth constraints, we are prohibited from sending the real samples or having frequent communication between agents. In this paper, we focus on scenarios with a simple communication mechanism where an agent sends its parameters only after training on a single experience. This is a sparse and efficient communication mechanism since we send the model only once per experience. We also separate the problem of *knowledge adaptation* from the problem of *knowledge consolidation*.

In knowledge adaptation, the model starts from an initialization θ_0 and minimizes the loss $\mathcal{L}(\mathcal{D}_i)$. We call the final result the *expert model* for $\mathcal{D}_i$ to highlight the fact that the model encodes knowledge about the experience, and we denote the expert's parameters as θ_i^{Exp}. In knowledge consolidation, we want to combine the agent trained on $S = e_1, \ldots, e_{i-1}$ with parameters θ_{i-1}^{CL} with the expert θ_i^{Exp}. The resulting model $f_{\theta_i^{CL}}$ consolidates the knowledge of both models. As a further constraint, we assume that during the consolidation phase we may not have access to the data $\mathcal{D}_i$. Therefore, we will use an out-of-distribution dataset $\mathcal{D}_{ood}$ to train the consolidated model.

In this paper, we study two different scenarios. In the *sequential setting*, we assume a edge-cloud scenario, where at time i the edge device (expert) has access to experience e_i, while the cloud server (consolidated CL model) never sees any real data due to privacy constraints. The server sends the current consolidated parameters θ_{i-1}^{CL} to the edge device to initialize its model, which is trained on e_i. This is the knowledge adaptation step, where the edge device finetunes its parameters and has full plasticity while ignoring any possible forgetting. At the end of the training step, the edge device sends the new parameters θ_i^{Exp} to the server. The server performs a knowledge consolidation step to combine the new parameters with the previous ones θ_{i-1}^{CL} and obtains θ_i^{CL}. During this step, the server must balance new and past knowledge to avoid catastrophic forgetting. Notice that the main challenge of our setup is the fact that the server will never see any original data and it will only receive the parameters at the end of training.

In the *independent setting*, each agent learns independently from different subsets of the data and communicates by sharing their parameters. In this work, we study a scenario where each agent i is trained on a separate experience e_i starting from a common initialization θ_0. As a result, we have a stream of trained experts $f_{\theta_1^{Exp}}, \ldots, f_{\theta_N^{Exp}}$. We can train a consolidated model $f_{\theta_i^{CL}}$ on the entire stream by sequentially consolidating the knowledge of the independent experts. Again, the consolidated model does not have access to any training data.

Figure 1 shows the difference between the sequential and independent setting. Notice that the independent scenario includes the possibility of reusing pretrained models or training the experts in parallel. However, it is more challenging since the agents may be more diverse as a result of the different initialization; in the sequential setting, instead, sequential initialization helps knowledge consolidation, as we will see in Section 4.1.

3 Data-Agnostic Consolidation

As discussed above, we separate learning in a multi-agent environment into two problems: knowledge adaptation and knowledge consolidation. We propose a method that solves each problem separately. The separation helps to simplify the two problems and to achieve consolidation without the original data. In Section 4, we will see that this explicit separation is beneficial even when original data is available during consolidation.

Knowledge Adaptation aims for optimal plasticity and it is solved by finetuning the model on the new experience e_i by minimizing $\mathcal{L}(\mathcal{D}_i, \theta_i)$, where θ_i are the expert parameters. Notice that the loss is computed only on $\mathcal{D}_i$, ignoring previous experiences and therefore resulting in catastrophic forgetting.

Knowledge Consolidation is a function matching problem where the objective function is the model $f^*_{\theta_i^{CL}}$ that combines the previous model $f_{\theta_{i-1}^{CL}}$ and the current expert $f_{\theta_i^{Exp}}$.

We propose *Data-Agnostic Consolidation (DAC)*, a method for knowledge consolidation solving the function matching problem via a double knowledge distillation (Section 3.1). At each iteration, DAC samples from a source of data and uses heavy augmentations to increase its diversity (Section 3.2).

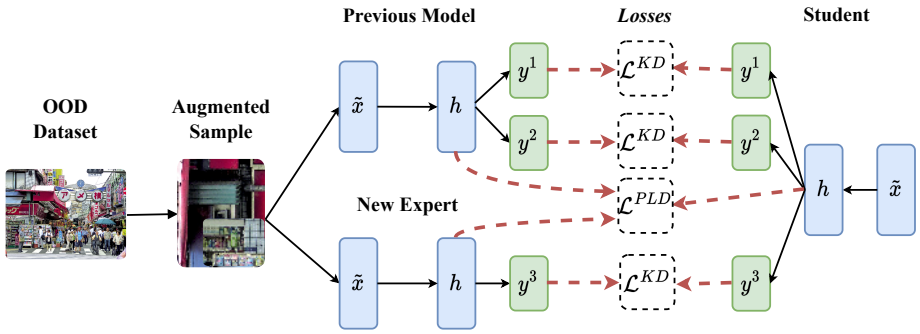


Figure 2: Data-Agnostic Consolidation. The method uses double distillation in the output and latent space, using heavily augmented samples as input.

Finally, DAC leverages a novel approach called *Projected Latent Distillation (PLD)* to distill the latent spaces of the two teachers (Section 3.3).

In the remainder of this section, we assume that $f_{\theta_i^{CL}}$, the current CL model, is a multi-head model, with a separate linear head for each task k. We denote $f^k_{\theta_i^{CL}}$ the function computing the output for task k. Since DAC uses multi-head models during consolidation, during inference we can use the correct head if task labels are available. In task-agnostic scenarios, we can average (or concatenate, if they predict different classes) the outputs of all the heads. A schematic view of the method is shown in Figure 2.

3.1 Knowledge Distillation and Function Matching

At experience i, we want to consolidate a multi-head model $f_{\theta_{i-1}^{CL}}$ with $i-1$ heads and a single-head model $f_{\theta_i^{Exp}}$. The desired result is a multi-head model $f^*_{\theta_i^{CL}}$ with i heads such that

$$f^{*k}_{\theta_i^{CL}}(\boldsymbol{x}) = f^k_{\theta_{i-1}^{CL}}(\boldsymbol{x}), \quad \forall k \in \{1, \dots, i-1\}, \boldsymbol{x} \in \mathbb{R}^N \tag{1}$$

$$f^{*i}_{\theta_i^{CL}}(\boldsymbol{x}) = f_{\theta_i^{Exp}}(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \mathbb{R}^N. \tag{2}$$

Notice that the above definition is an exact solution, there is no noise or uncertainty in our target since our goal is to replicate the two models exactly. In fact, we know the exact target for every possible input. We can find $f_{\theta_i^{CL}}$ by stochastic gradient descent minimizing a double knowledge distillation loss

$$\mathcal{L}^{DKD}(\mathcal{D}) = \mathcal{L}^{KD}(\mathcal{D}, f^i_{\theta_i^{CL}}, f_{\theta_i^{Exp}}) + \sum_{k=1}^{i-1} \mathcal{L}^{KD}(\mathcal{D}, f^k_{\theta_i^{CL}}, f^k_{\theta_{i-1}^{CL}}) \tag{3}$$

where $\mathcal{L}^{KD}(\mathcal{D}, \theta_S, \theta_T)$ is the KL-divergence computed on data $\mathcal{D}$ between the student θ_S and the teacher θ_T. $\mathcal{L}^{DKD}$ minimizes the error of each head separately. Since the target solution in Eq. 2 is defined over the full $\mathbb{R}^N$ domain, we do not necessarily need the original data but we need to define a method to sample inputs from the entire space.

3.2 Sampling Data for Consolidation

As explained before, one of the main challenges of multi-agent CL is that we do not have access to the real data when performing the consolidation step. In a single-agent scenario, the current data $\mathcal{D}_i$ will be often available. However, in a multi-agent scenario we may not have $\mathcal{D}_i$ due to privacy constraints. Even in the single-agent scenario, if we perform the consolidation step in a separate server, we may not have access to $\mathcal{D}_i$. Fortunately, it is not necessary to use the original data to optimize the consolidation as described in Section 3.1. However, using random vectors in $\mathbb{R}^N$ would be highly inefficient. An alternative solution is to assume to have access to a small set of samples $\mathcal{D}_{ood}$, possibly coming from different tasks. Following Asano & Saeed (2022) and Beyer et al. (2022), we use a large set of stochastic augmentations to create a dataset of highly diverse samples from a small number of

(a) Results on task-incremental SplitCIFAR100 after task 5 and 10. Baselines denoted by † are taken from Masana et al. (2020)

	SplitCIFAR100	
	5 Tasks	10 Tasks
Naive†	49.8	38.3
EWC†	60.2	56.7
PathInt†	57.3	53.1
MAS†	61.8	58.6
RWalk†	56.3	49.3
DMC†	72.3	66.7
LwM†	76.2	70.4
LwF†	76.7	76,6
DAC (city)	$81.4_{\pm 1.6}$	$80.5_{\pm 0.8}$

(b) Results on task-agnostic independent settings for CORe50. Baselines denoted by † are taken from Carta et al. (2022).

	CORe50	
	Joint	NI
Oracle†	$85.7_{\pm 0.2}$	–
Min. Entropy†	–	$61.3_{\pm 1.8}$
Output Avg.†	–	$69.9_{\pm 0.7}$
Parameter Avg.†	–	$2.0_{\pm 0.0}$
Replay ED†	$87.4_{\pm 0.2}$	$83.7_{\pm 0.5}$
Model Inversion ED†	$50.0_{\pm 2.7}$	$44.3_{\pm 4.9}$
Data Impression ED†	$52.9_{\pm 2.0}$	$43.2_{\pm 2.3}$
Aux. Data ED†	$81.8_{\pm 0.2}$	$44.5_{\pm 2.9}$
DAC ("city")	$84.9_{\pm 0.2}$	$40.9_{\pm 3.0}$

Table 1: Average accuracy A_t on task-incremental SplitCIFAR100 (after tasks 5 and 10) and task-agnostic CORe50 (last model).

images. At each timestep, we apply a large number of transformations such as jittering, rotation, crop, resize, flip, CutMix (Yun et al., 2019). The resulting images will be heavily distorted but to solve the knowledge distillation problem defined in Eq. 2 we do not need realistic images as long as we have enough diversity, as we will show in Section 4. While the preprocessing pipeline can become the bottleneck of the training process with such a large number of transformations, it is always possible to trade-off memory to save computation by precomputing a large number of pre-processed images.

3.3 Projected Latent Distillation

The knowledge distillation loss in Eq. 3 matches the outputs for each teacher's head. However, we would like to also match the latent space of the two teachers. This is more problematic because given an input $\boldsymbol{x}$, the outputs $\boldsymbol{y}_k$ are computed by separate heads, and therefore have no interference, but the hidden activations share the same units in the consolidated model (as shown in Figure 2). As a result, for a specific hidden layer we have two different targets $\boldsymbol{h}_{i-1}^{CL}$ and $\boldsymbol{h}_i^{Exp}$ and a single activation vector $\boldsymbol{h}_i^{CL}$ for the student. Exact matching of the hidden state is not possible. To solve this issue, we propose *Projected Latent Distillation (PLD)*. The underlying intuition is that while we cannot enforce an exact match, we can match the two hidden states up to a linear transformation. During the consolidation phase, we optimize two linear transformations $\boldsymbol{W}^{Exp}$ and $\boldsymbol{W}^{CL}$ that map the teachers' hidden states to the student's hidden states. The loss is then defined as

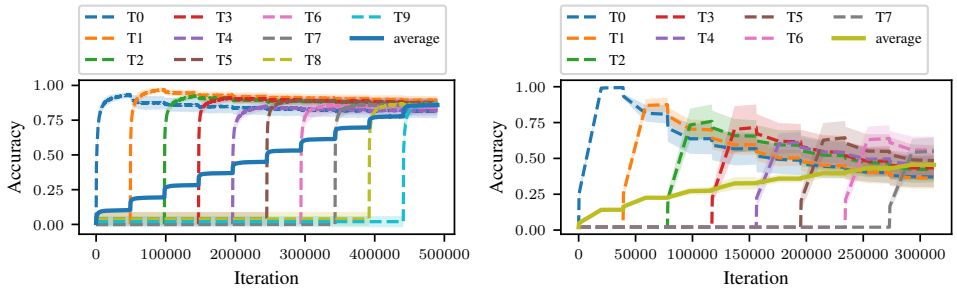
$$\mathcal{L}^{PLD}(\boldsymbol{h}_i^{CL}, \boldsymbol{h}_{i-1}^{CL}, \boldsymbol{h}_i^{Exp}) = \lambda(||\boldsymbol{W}^{Exp}\boldsymbol{h}_i^{CL} - \boldsymbol{h}_i^{Exp}||_2^2 + \tag{4}$$

$$(i-1)||\boldsymbol{W}^{CL}\boldsymbol{h}_i^{CL} - \boldsymbol{h}_{i-1}^{CL}||_2^2), \tag{5}$$

where $\boldsymbol{h}_i^{CL}$ is the student, $\boldsymbol{h}_{i-1}^{CL}$ the previous CL model (already trained on $i-1$ tasks) and $\boldsymbol{h}_i^{Exp}$ the expert hidden states. The $\boldsymbol{W}$ matrices are initialized to the identity matrix and optimized during the consolidation phase. The loss encourages the student model to also match the teachers' hidden states. Notice that the loss for the previous expert is multiplied by $i-1$ to give the same weight to each task. The same effect is present in the distillation in the output space defined by Eq. 3, since we sum all the heads (one for each task). The total loss of DAC is $\mathcal{L}(\mathcal{D}, \boldsymbol{h}_i^{CL}, \boldsymbol{h}_{i-1}^{CL}, \boldsymbol{h}_i^{Exp}) = \mathcal{L}^{DKD}(\mathcal{D}) + \lambda\mathcal{L}^{PLD}(\boldsymbol{h}_i^{CL}, \boldsymbol{h}_{i-1}^{CL}, \boldsymbol{h}_i^{Exp})$, where λ controls the ratio between the ouput and latent distillation losses.

4 Experiments

We show experimental results in the sequential and independent setting introduced in Section 2. We use CIFAR100 (Krizhevsky) and CORe50 (Lomonaco & Maltoni, 2017) to create our benchmarks.



(a) SplitCIFAR100.

(b) CORe50-NI.

Figure 3: Task-specific and stream average accuracy on the training set during training.

The source code is implemented in Avalanche (Lomonaco et al., 2021) and publicly available[1]. More details about the experiments can be found in the appendix.

We use SplitCIFAR100 in the task-incremental setting 10, and 20 tasks, where the stream is divided into experiences of 10, and 5 classes, respectively. We use a slimmed ResNet18 as defined in Lopez-Paz & Ranzato (2017). We also use CORe50 (Lomonaco & Maltoni, 2017), a task agnostic-benchmark, in the joint and domain-incremental (NI) settings. CORe50 images have a 224×224 resolution, which we rescale to 128×128. We use a MobileNetv2 (Howard et al., 2017) pretrained on ImageNet, as it is popular in the literature (Maltoni & Lomonaco, 2019).

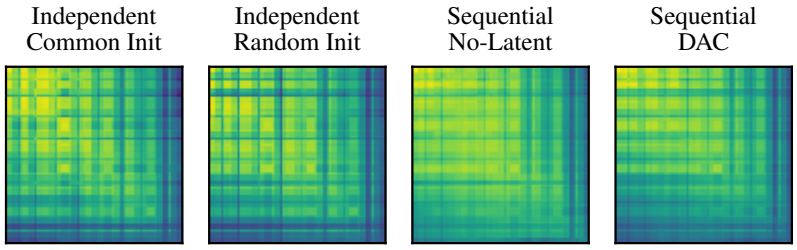
All the results are the average of 5 runs on different seeds. Results are evaluated with the average accuracy over the entire test stream. Given $A_{t,i}$ as the task i accuracy for task i after training on task t, the average accuracy $A_t = \sum_{i=1}^{t} A_{t,i}$. The default data source for DAC is a single image, "city". This is a high resolution image of a japanese street (shown in the Appendix).

4.1 Results

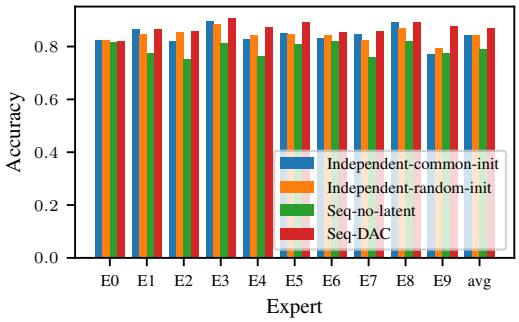
Results for SplitCIFAR100 in the task-aware sequential setting are shown in table 1a. We use the results in (Masana et al., 2020) as baselines. Our results show state of the art performance on the 10 task setting. Despite the limited data source, a single out-of-domain image ("city"), DAC outperforms LwF, which uses the real data $\mathcal{D}_i$ for distillation. We argue that there are two properties of DAC which justify this improvement. First, the use of heavy augmentations improves distillation even with limited data, as already shown in (Beyer et al., 2022) and (Asano & Saeed, 2022) for offline training. Furthermore, during the consolidation DAC weighs the current task and all the previous ones in the same way by using the same loss for all tasks and summing them (Eq. 3, 5). Instead, LwF uses the cross-entropy for the current task and the KL divergence for the previous ones, which makes it more difficult to balance stability and plasticity (the well known stability-plasticity dilemma, (French, 1999)). Interestingly, DAC scales very well with the number of tasks in the 10 tasks setting, unlike most of the other methods. We note that increasing the number of tasks makes the consolidation problem harder but it also makes each task easier to solve because each task will have less classes. We hypothesize that this issue is due to the poor stability-plasticity trade-off of most methods. We evaluated DAC on the 20 task setting (5 classes per task), which results in the average test accuracy of $86.2_{\pm 1.0}$, even higher than the 10 task setting.

Experiments on CORe50 in the independent setting are shown in Table 1b. Notice that while our method uses a multi-head, CORe50 is a task-agnostic benchmark. Therefore, we convert the final multi-head model into a task-agnostic model by averaging the output of all the heads. CORe50 is also a more challenging benchmark due to the higher image resolution (128×128). We compare against the methods in Carta et al. (2022), which perform knowledge distillation using synthetic data or the entire ImageNet dataset. Overall, DAC obtains the best performance in the joint scenario, while Aux. Data ED is better in the domain incremental (NI) scenario. Notice that Aux. Data ED uses ImageNet (with more than 1 milion images) for distillation.

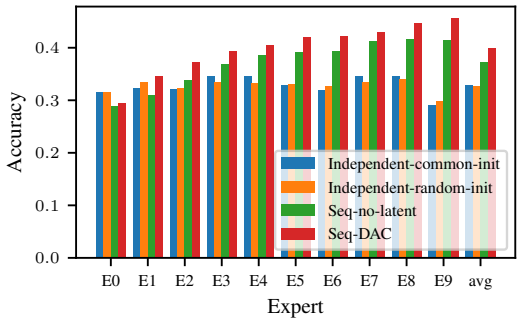
[1]Only after the review. An anonymized version is available as supplementary material.



(a) CKA between the first and last experts of SplitCIFAR100 (10 Tasks) computed using the last task.



(b) Accuracy on the expert's task.



(c) Task-agnostic linear probing on all tasks.

Figure 4: Accuracy of the experts on their own task and linear probing of the expert's representation finetuned on the entire CIFAR100 dataset.

Figure 3 shows the learning curves for SplitCIFAR100 (10 Tasks) and CORe50-NI. We notice that on Split CIFAR100 the forgetting, i.e. the difference between the accuracy after training on task i and the task at the end of training on the entire stream, is very low. This means that the consolidation process works with minimal forgetting. On CORe50, we see more forgetting. We hypothesize that training the consolidation for more epochs on CORe50 could reduce the gap.

Comparison between Sequential and Independent Scenario

At a first glance, the sequential and independent scenario may seem very similar. The only difference between the two lies in the expert's initialization. In the sequential setting, each agent is initialized with the weights learned at time $t-1$, while in the independent setting all the agents start from a common initialization θ^0. In particular, the model's initialization affects the similarity between the experts and therefore the difficulty of the consolidation problem.

In this section, we study how the two settings affect the expert's accuracy with respect to 3 dimensions: forward transfer, the generalization of the hidden representations to other tasks, and the representation similarity between different experts. We also ablate the use of Projected Latent Distillation for DAC to investigate its effect. We use the experts trained on SplitCIFAR100 (10 Tasks). In the independent setting, experts are trained either with the same random initialization (`ind-same`) or a different one (`ind-random`). In the sequential setting we compare DAC without latent distillation (`seq-no-latent`), and the full DAC (`seq-DAC`).

We would like the initialization of the sequential setting to encourage forward transfer between the CL model and the expert, i.e. the sequential initialization should improve the expert's performance compared to the independent setting. Figure 4 shows the accuracy of the 10 experts on the task they have seen during training (left) and the average accuracy over all tasks for a task-agnostic linear probe which uses the final layer's representation, finetuned on all the tasks (right). Surprisingly, while `seq-no-latent` obtains a higher accuracy than `ind-same` and `ind-random` on the linear probing, the average expert's accuracy on the task seen during training is actually lower. Therefore, it seems that without latent distillation there is negative forward transfer. Instead, `seq-DAC` experts are better than the `ind-same` experts, which means that the initialization favors a positive forward

Table 2: Test accuracy of DAC with different data sources on SplitCIFAR100 (10 Tasks).

	In-Domain	Single Image	Natural	Accuracy
Current Data ($\mathcal{D}_i$)	✓		✓	$84.2_{\pm 0.5}$
ImageNet			✓	$84.8_{\pm 0.6}$
city		✓	✓	$80.5_{\pm 0.8}$
animals		✓	✓	$82.1_{\pm 0.5}$
bridge		✓	✓	$79.0_{\pm 0.6}$
hubble		✓		$65.1_{\pm 0.3}$
noise		✓		$10.7_{\pm 0.5}$

transfer. The linear probe accuracy (Fig. 4c) on all tasks shows positive forward transfer for both `seq-no-latent` and `seq-DAC`, albeit `seq-DAC` has a better performance.

In Figure 4a, we measure representation similarity with the CKA (Nguyen et al., 2022). The figure shows the CKA between the first (Task 0) and last (Task 9) experts[2]. In general, the CKA similarity is relatively high for the different configurations since they all share the same architecture and are trained on similar data. Somewhat surprisingly, the similarity between `seq-DAC` experts is lower than the similarity of `seq-no-latent` experts. In principle, we expected to find a positive correlation between the representation similarity and the forward transfer. This appears not to be the case since `seq-DAC` has better forward transfer but `seq-no-latent` shows closer similarity. We hypothesize that the PLD loss encourages the consolidated model to learn more diverse representations, decreasing the representation similarity over time but increasing the forward transfer thanks to the richer representation.

Comparison Between Different Data Sources

We have already shown that a single image is already competitive with state of the art methods. In this section, we study how different data source properties help the consolidation process (Table 2). We use data sources taken from the real stream, single images vs full dataset, and natural images against other domains and static noise. We use:

- city: high-resolution (around 2560×1920, 1.85MB) image of a japanese market;
- animals: medium-resolution (600×225, 338KB) poster with several animals;
- hubble: high-resolution image (2300×2100, 6.90MB) from the Hubble telescope;
- bridge: Image of the San Francisco Golden Gate Bridge (1165×585, 1.17MB);
- ImageNet: samples from ImageNet (using only CutMix);
- noise: static noise.

Images and samples are shown in Appendix B. In general, we notice that using a single image is sufficient to reach a very high performance. However, there is still a large gap between the use of real data ($\mathcal{D}_i$) and a single out-of-distribution image. It is important to notice that DAC helps even when using the real data since we obtain an accuracy of 84.2 against the 65.8 of LwF. However, even very different domains ("hubble") still work better than completely random data ("noise"). Finally, diversity, given either by a large $\mathcal{D}_{ood}$ or by heavy augmentations seems to be the most important factor since using ImageNet (more than 1M images) is slightly better than using data from the current task (5000 images).

4.2 Analysis

We can summarize the main findings as follow:

Benefits of separate adaptation and consolidation phases. The consolidation problem becomes easier once it's separated from the adaptation, resulting in a higher accuracy, a better stability-plasticity tradeoff, and better scaling w.r.t. the number of tasks. Notice all of these improvements are

[2]The CKA for the entire stream is shown in the appendix.

shown even when DAC does not have access to the current data $\mathcal{D}_i$ during consolidation, unlike all the other methods (except DMC).

Sequential and independent scenario. The independent scenario is more difficult than the sequential one due to lower similarity between the experts and zero opportunity for forward transfer. In the sequential setting, DAC shows positive forward transfer, but only when the PLD loss is used. This result suggests that latent distillation helps to combine models with different representations and to learn richer latent representations.

Data sources. Limited data sources, such as a single out-of-distribution image, show a good performance due to the heavy augmentations. This opens up to the opportunity of offloading the consolidation to a server without sharing the data.

5 Related Works

Recent work on knowledge distillation (KD) (Hinton et al., 2015) explores the possibility of distillation with limited data. There exist several works on data-free KD, especially methods that try to create synthetic images with generators (Liu et al., 2021). More relevant to our work, Baradad et al. (2021) propose handcrafted noise and simple procedurally generated images, while Fang et al. (2021) creates image by combining together slices of several images. Asano & Saeed (2022) shows the beneficial effect of heavy augmentations and Beyer et al. (2022) additionally shows the benefit of long training schedules.

In continual learning, many popular methods are based on knowledge distillation (Li & Hoiem, 2017; Buzzega et al., 2020). Progress and Compress (Schwarz et al., 2018) uses an adaptation and and compression step reminescent of our consolidation, but it still needs the original data. Gomez-Villa et al. (2022) applies feature distillation in a continual self-supervised setting. In continual learning, exemplar-free scenarios are very popular, especially in the class-incremental setting (DFCIL). This scenario is different from our setting since the model still has access to $\mathcal{D}_i$ (see also Appendix A). Many strategies addresses DFCIL by using alternative sources of data. Carta et al. (2022) uses synthetic data generated via model inversion, Zhang et al. (2020) uses external data, and Smith et al. (2021) uses a data-free training process similar to GANs. Lee et al. (2019) can optionally use external data for model calibration. Yu et al. (2022) and Dong et al. (2021) uses external data for semantic segmentation and object detection, respectively, by exploiting the notion of background on these tasks, which provide a neutral label, which makes them inapplicable for classification tasks. Among all these strategies, only Carta et al. (2022) (ED, Table 1b) and Zhang et al. (2020) (DMC, Table 1a) are fully applicable to our constrained scenario.

6 Conclusion

In this paper, we studied the problem of knowledge sharing between agents learning in non-stationary environments. First, we formalized this problem as a multi-agent continual learning scenario. Then, we highlighted how each learning step can be split into an adaptation and consolidation phase. We proposed DAC as a general double distillation method. DAC uses heavy augmentations to achieve competitive results with very limited data sources. Additionally, DAC uses PLD to distill the latent space of the two teachers. The results show state-of-the-art performance and highlight how each of the DAC components improves the final performance.

Multi-agent CL is still an open problem. We focused on the problem of knowledge consolidation in this paper, but many other aspects are worth of study, such as increasing the communication frequency, sending other forms of encoded knowledge instead of the parameters, or studying efficient and sparse communication protocols between the agents.

To conclude, we would like to point out that improvements in the multi-agent setting can easily transfer to the single-agent scenario, as it happened for DAC, which improved the state-of-the-art performance on the task-incremental SplitCIFAR100. We hope that the research in multi-agent scenarios will also help single-agent CL by providing important insights, such as the relationship between adaptation and consolidation explored in this paper.

REPRODUCIBILITY STATEMENT

We release our source code (anonymized in the supplementary material for the review, public on github afterwards). All of our experiments use Avalanche (Lomonaco et al., 2021), a continual learning library based on PyTorch (Paszke et al., 2019). We release all of the experiments' configurations using Hydra (Yadan, 2019)(hierarchical yaml configuration files), which means that each experiment in the paper can be reproduced by running a main python script with the desired configuration, as detailed in the README of the source code. Experimental details relevant for independent implementations are available in the appendix.

REFERENCES

Yuki M. Asano and Aaqib Saeed. Extrapolating from a Single Image to a Thousand Classes using Distillation. *arXiv:2112.00725 [cs]*, January 2022.

Manel Baradad, Jonas Wulff, Tongzhou Wang, Phillip Isola, and Antonio Torralba. Learning to See by Looking at Noise. *arXiv:2106.05963 [cs]*, December 2021.

Lucas Beyer, Xiaohua Zhai, Amélie Royer, Larisa Markeeva, Rohan Anil, and Alexander Kolesnikov. Knowledge Distillation: A Good Teacher Is Patient and Consistent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10925–10934, 2022.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark Experience for General Continual Learning: A Strong, Simple Baseline. *arXiv:2004.07211 [cs, stat]*, April 2020.

Antonio Carta, Andrea Cossu, Vincenzo Lomonaco, and Davide Bacciu. Ex-Model: Continual Learning From a Stream of Trained Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3790–3799, 2022.

Na Dong, Yongqiang Zhang, Mingli Ding, and Gim Hee Lee. Bridging Non Co-occurrence with Unlabeled In-the-wild Data for Incremental Object Detection, October 2021.

Gongfan Fang, Yifan Bao, Jie Song, Xinchao Wang, Donglin Xie, Chengchao Shen, and Mingli Song. Mosaicking to Distill: Knowledge Distillation from Out-of-Domain Data. In *Thirty-Fifth Conference on Neural Information Processing Systems*, May 2021.

Robert M. French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3 (4):128–135, April 1999. ISSN 1364-6613. doi: 10.1016/S1364-6613(99)01294-2.

Alex Gomez-Villa, Bartlomiej Twardowski, Lu Yu, Andrew D. Bagdanov, and Joost van de Weijer. Continually Learning Self-Supervised Representations with Projected Functional Regularization. *arXiv:2112.15022 [cs]*, May 2022.

Tyler L. Hayes and Christopher Kanan. Lifelong Machine Learning with Deep Streaming Linear Discriminant Analysis. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 887–896, Seattle, WA, USA, June 2020. IEEE. ISBN 978-1-72819-360-1. doi: 10.1109/CVPRW50498.2020.00118.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. pp. 1–9, 2015. ISSN 3531207857. doi: 10.1063/1.4931082.

Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv:1704.04861 [cs]*, April 2017.

Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. pp. 60.

Kibok Lee, Kimin Lee, Jinwoo Shin, and Honglak Lee. Overcoming Catastrophic Forgetting with Unlabeled Data in the Wild, October 2019.

Timothée Lesort, Vincenzo Lomonaco, Andrei Stoian, Davide Maltoni, David Filliat, and Natalia Díaz-Rodríguez. Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. *Information Fusion*, 58:52–68, June 2020. ISSN 1566-2535. doi: 10.1016/j.inffus.2019.12.004.

Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Processing Magazine*, 37(3):50–60, May 2020. ISSN 1558-0792. doi: 10.1109/MSP.2020.2975749.

Zhizhong Li and Derek Hoiem. Learning without Forgetting. *arXiv:1606.09282 [cs, stat]*, February 2017.

Yuang Liu, Wei Zhang, Jun Wang, and Jianyong Wang. Data-Free Knowledge Transfer: A Survey. *arXiv:2112.15278 [cs]*, December 2021.

Vincenzo Lomonaco and Davide Maltoni. CORe50: A new dataset and benchmark for continuous object recognition. In Sergey Levine, Vincent Vanhoucke, and Ken Goldberg (eds.), *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pp. 17–26. PMLR, November 2017.

Vincenzo Lomonaco, Lorenzo Pellegrini, Andrea Cossu, Antonio Carta, Gabriele Graffieti, Tyler L. Hayes, Matthias De Lange, Marc Masana, Jary Pomponi, Gido M. van de Ven, Martin Mundt, Qi She, Keiland Cooper, Jeremy Forest, Eden Belouadah, Simone Calderara, German I. Parisi, Fabio Cuzzolin, Andreas S. Tolias, Simone Scardapane, Luca Antiga, Subutai Ahmad, Adrian Popescu, Christopher Kanan, Joost van de Weijer, Tinne Tuytelaars, Davide Bacciu, and Davide Maltoni. Avalanche: An End-to-End Library for Continual Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3600–3610, 2021.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in Neural Information Processing Systems*, 2017-Decem(Nips):6468–6477, 2017.

Davide Maltoni and Vincenzo Lomonaco. Continuous learning in single-incremental-task scenarios. *Neural Networks*, 116:56–73, August 2019. ISSN 0893-6080. doi: 10.1016/j.neunet.2019.03.010.

Marc Masana, Xialei Liu, Bartlomiej Twardowski, Mikel Menta, Andrew D. Bagdanov, and Joost van de Weijer. Class-incremental learning: Survey and performance evaluation. *arXiv:2010.15277 [cs]*, October 2020.

Thao Nguyen, Maithra Raghu, and Simon Kornblith. Do Wide and Deep Networks Learn the Same Things? Uncovering How Neural Network Representations Vary with Width and Depth. In *International Conference on Learning Representations*, February 2022.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H Wallach, H Larochelle, A Beygelzimer, F d\textquotesingle Alché-Buc, E Fox, and R Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

Jonathan Schwarz, Jelena Luketina, Wojciech M. Czarnecki, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. *35th International Conference on Machine Learning, ICML 2018*, 10: 7199–7208, 2018. ISSN 9781510867963.

James Smith, Yen-Chang Hsu, Jonathan Balloch, Yilin Shen, Hongxia Jin, and Zsolt Kira. Always Be Dreaming: A New Approach for Data-Free Class-Incremental Learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9374–9384, 2021.

Gido M. van de Ven and Andreas S. Tolias. Three scenarios for continual learning, April 2019.

Omry Yadan. Hydra - A framework for elegantly configuring complex applications. Github, 2019.

Lu Yu, Xialei Liu, and Joost van de Weijer. Self-Training for Class-Incremental Semantic Segmentation, March 2022.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features, August 2019.

Junting Zhang, Jie Zhang, Shalini Ghosh, Dawei Li, Serafettin Tasci, Larry Heck, Heming Zhang, and C.-C. Jay Kuo. Class-incremental Learning via Deep Model Consolidation. *arXiv:1903.07864 [cs]*, January 2020.

Table 3: Summary of the main properties of different scenario related to multi-agent continual learning.

	edge data privacy	low computational demands at the edge	low synchronization overhead
Joint Training			
Continual Learning	✓		
Federated Learning	✓	✓	
Ours	✓	✓	✓

A Comparison Between Related Learning Scenarios

The multi-agent continual learning scenario present some similarities with other scenarios in the literature. We provide a more detailed discussion of their differences here hoping to highlight their difference:

single-agent CL : a single agent learning from a nonstationary stream of data. Knowledge sharing is not necessary and current data is always available.

rehearsal-free CL : includes scenarios such as data-free class-incremental learning (DFCIL), where a single agent learning from a nonstationary stream of data. Data from previous experiences is unavailable due to privacy constraints or severe storage limitations. Knowledge sharing is not necessary and current data is always available.

federated : client-server organization with a single centralized controller. All the clients are learning the same task. The server has full control over the training process and the client synchronize every few training iterations.

sequential multi-agent : Each agent learns a separate task and shares its knowledge with the others. Training agent i starts after agent $i + 1$ has completed its training.

independent multi-agent : Each agent is trained in parallel, starting from a common initialization. Knowledge consolidation happens after all the agents have been trained.

In the multi-agent and federated settings, we assume that privacy between different entities must be ensured, which means that agents do not share raw data with each other. Figure 5 shows the training process of the four different scenarios assuming explicit adaptation and consolidation phases as defined in Section 2. Table 3 summarizes the properties of the different scenario.

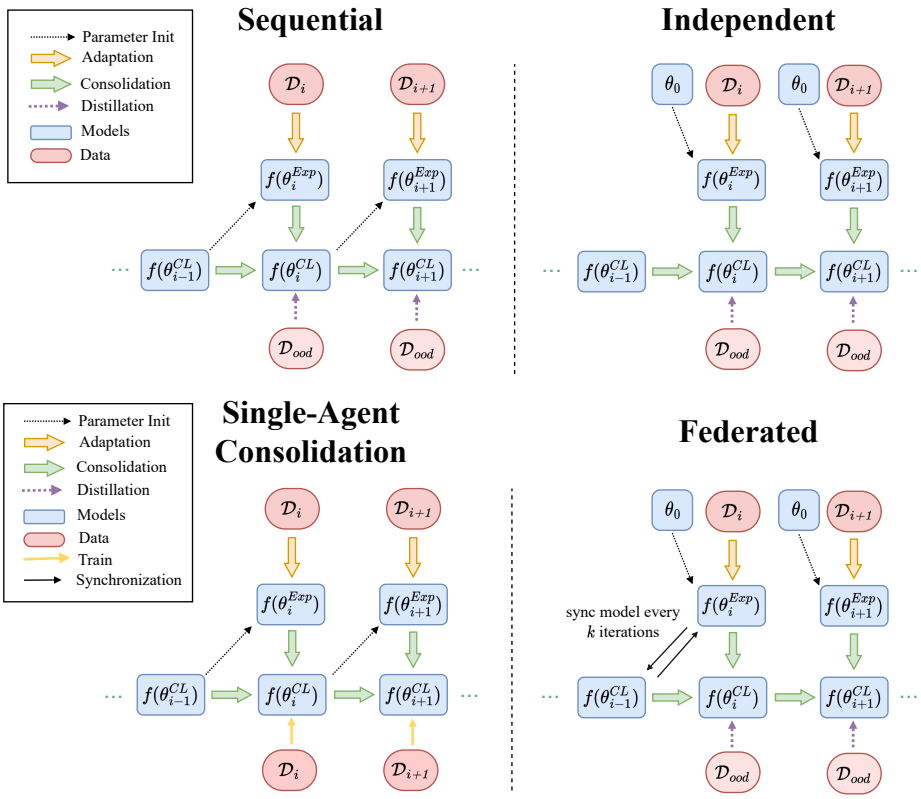


Figure 5: Schematic comparison of different learning scenarios.

B DATA SOURCES

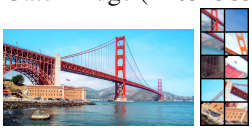
In this section, we show samples from the images used for knowledge distillation. *city*: high-resolution (around 2560x1920, 1.85MB) image of a japanese market;

animals: medium-resolution (600x225, 338KB) poster with several animals;

hubble: high-resolution image (2300x2100, 6.90MB) from the Hubble telescope;

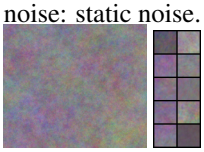
animals: medium-resolution (600x225, 338KB) poster with several animals;

bridge: Image of the San Francisco Golden Gate Bridge (1165x585, 1.17MB);



ImageNet: samples from ImageNet (without augmentations);

noise: static noise.



C HYPERPARAMETERS

SplitCIFAR100: We use a slimmed ResNet18 as a backbone for both the teacher and consolidated model. During the consolidation, we use Adam with learning rate set to 0.0001, with a batch size of 512 and $500'000$ iterations. We use a temperature of 0.5 for distillation. For the PLD loss, we set $\lambda = 0.01$ and apply the loss at `layer4.0` and `linear` (logits).

CORe50: We use a MobileNet v2 pretrained on ImageNet as a backbone for both the teacher and consolidated model. During the consolidation, we use Adam with learning rate set to 0.0001, with a batch size of 128 and $100'000$ iterations. We use a temperature of 1.0 for distillation. For the PLD loss, we set $\lambda = 100.0$ and apply the loss at `classifier` (logits).

D CKA

In this section, we show the CKA, as described in Section 4.1, for the entire stream. We compute the CKA between the first expert and the expert after experience i.

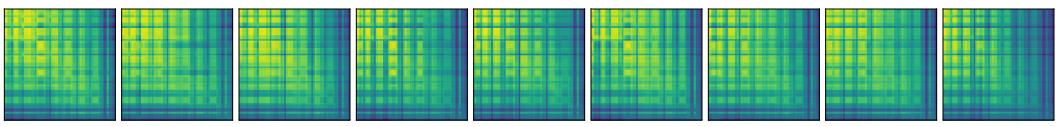


Figure 6: CKA for `ind-same`.

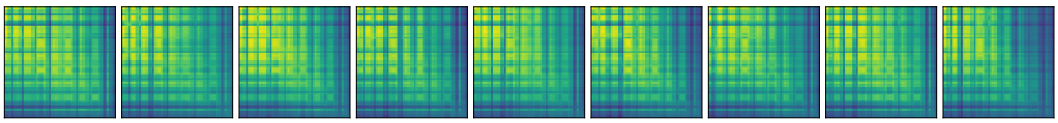


Figure 7: CKA for `ind-random`.

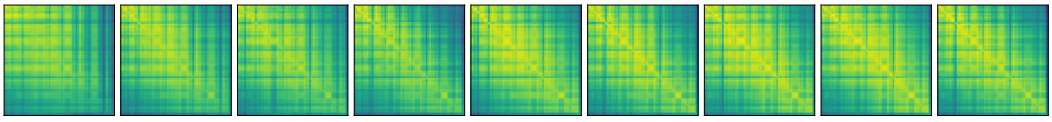


Figure 8: CKA for `seq-no-latent`.

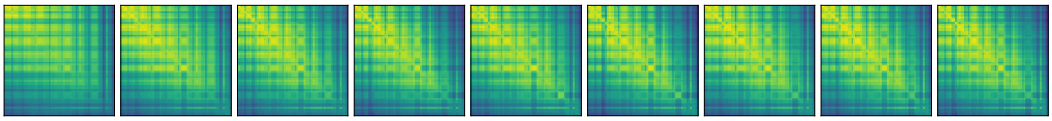


Figure 9: CKA for `seq-DAC`.