CONSISTENCY IS THE KEY TO FURTHER MITIGATING CATASTROPHIC FORGETTING IN CONTINUAL LEARNING

Prashant Bhat, Bahram Zonooz,[*] **Elahe Arani**[*]
Advanced Research Lab, NavInfo Europe, Eindhoven, The Netherlands
{prashant.bhat, elahe.arani}@navinfo.eu, bahram.zonooz@gmail.com

ABSTRACT

Deep neural networks struggle to continually learn multiple sequential tasks due to catastrophic forgetting of previously learned tasks. Rehearsal-based methods which explicitly store previous task samples in the buffer and interleave them with the current task samples have proven to be the most effective in mitigating forgetting. However, Experience Replay (ER) does not perform well under low-buffer regimes and longer task sequences as its performance is commensurate with the buffer size. Consistency in predictions of soft-targets can assist ER in preserving information pertaining to previous tasks better as soft-targets capture the rich similarity structure of the data. Therefore, we examine the role of consistency regularization in ER framework under various continual learning scenarios. We also propose to cast consistency regularization as a self-supervised pretext task thereby enabling the use of a wide variety of self-supervised learning methods as regularizers. While simultaneously enhancing model calibration and robustness to natural corruptions, regularizing consistency in predictions results in lesser forgetting across all continual learning scenarios. Among the different families of regularizers, we find that stricter consistency constraints preserve previous task information in ER better. [1]

1 INTRODUCTION

Continual Learning (CL) refers to a learning paradigm where computational systems sequentially learn multiple tasks with data becoming progressively available over time. An ideal CL system must be plastic enough to integrate novel information and stable enough to not interfere with the consolidated knowledge (Parisi et al., 2019). In deep neural networks, however, sufficient plasticity to acquire new tasks results in large weight changes disrupting consolidated knowledge, referred to as catastrophic forgetting (Goodfellow et al., 2013; McCloskey & Cohen, 1989). Catastrophic forgetting often leads to a swift drop in performance and in the worst case, the previously learned information is completely overwritten by the new one (Parisi et al., 2019). The menace of catastrophic forgetting is not limited to CL alone, but manifests in multitask learning (Kudugunta et al., 2019), and supervised learning under domain shift (Ovadia et al., 2019) as well.

Several approaches have been proposed in the literature to address the stability-plasticity dilemma, the root cause of catastrophic forgetting in CL in neural networks. Weight-regularization methods (Zenke et al., 2017; Schwarz et al., 2018) impose explicit constraints on the neural network updates via an additional regularization term thereby restricting the change in weights pertaining to previous tasks. Although fairly successful in some CL scenarios, these methods fail in scenarios such as class-incremental learning. Parameter-isolation methods (Rusu et al., 2016) allocate a distinct set of parameters to distinct tasks to minimize the interference. However, these methods do not scale well with a large number of tasks. Rehearsal-based methods (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2019) explicitly store and replay a subset of previous task samples along with the current batch of samples, have proven to be most effective in minimizing interference in challenging CL tasks (Farquhar & Gal, 2018).

Simple Experience Replay (ER) (Ratcliff, 1990; Robins, 1995) which interleaves old task samples with the current task samples is one of the top-performing methods across different CL scenarios (Buzzega et al., 2020). There might be scenarios where large memory footprint is not as problematic, in this work we focus on a specific problem setting where large memory footprint is expensive. Given such a scenario, ER does not perform well under low-buffer regimes and longer task sequences as its performance is commensurate with the buffer size. To preserve the information pertaining to previous tasks better, soft-targets can be leveraged as they contain more information than the hard targets and capture

---

[*] Shared last author.
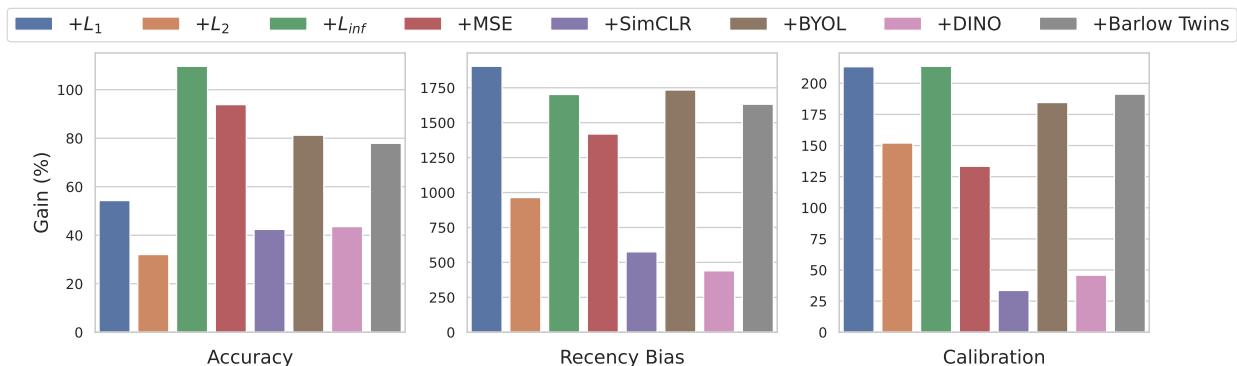[1] Code can be found at: https://github.com/NeurAI-Lab/ConsistencyCL.

Figure 1: Relative performance gain of different consistency regularizers added to ER over vanilla ER. All models are trained on S-TinyImageNet with buffer size 500. Higher value indicates better performance gain. More details can be found in the Appendix C.3.

the rich similarity structure of the data (Hinton et al., 2015). Therefore, several works (Benjamin et al., 2018; Buzzega et al., 2020; Arani et al., 2022; Pham et al., 2021) leverage soft-targets in addition to hard ones thereby drawing more information from the limited buffered samples. Despite differences in their architecture and training paradigm, these methods have one thing in common: enforcing consistency in predictions across time-separated views of buffered samples across tasks. By regularizing consistency in predictions, these approaches preserve rich information about the previous tasks leading to a further reduction in forgetting.

Consistency regularization has been widely used in semi-supervised learning to ensure that model's output is unaffected for samples augmented in a semantic-preserving way (Xie et al., 2020; Zhai et al., 2019; Sajjadi et al., 2016). However, a comprehensive study to understand the role of consistency regularization in CL under a common framework does not exist. Therefore, we examine the role of consistency regularization in CL under a common framework: we store the model's predictions along with the input image and corresponding ground-truth in the buffer. In the subsequent training iterations, we penalize the CL model for its deviation in predictions. We explore Minkowski distance functions (L_p), Kullback-Leibler Divergence (KL-Div), and Mean-Squared Error (MSE) as consistency regularizers. In addition, we cast regularizing consistency as a pretext task of bringing closer the corresponding predictions in the representational space by employing state-of-the-art Self-Supervised Learning (SSL) methods. This gives us the design flexibility to consider a wide variety of methods as regularizers thereby expanding the ambit of our analyses.

We find that any form of consistency regularization is the key to further leveraging buffered information in ER across all different CL scenarios. Moreover, stricter consistency constraints preserve the previous task information better enabling the CL model to forget less. Our empirical results show that Minkowski distance functions, extremely simple and intuitive are surprisingly one of the most effective families in alleviating catastrophic forgetting. Additionally, regularizing consistency in predictions leads to a more robust and well-calibrated model. Consistency regularization also mitigates the task recency bias thereby leading to more evenly distributed task prediction probabilities. Figure 1 provides a brief overview of some of our results.

2 RELATED WORKS

CL has remained one of the long-standing challenges for neural networks due to catastrophic forgetting (Hassabis et al., 2017). The stability-plasticity dilemma, the root cause of catastrophic forgetting, has been well studied in both biological and artificial neural networks (Mermillod et al., 2013; Grossberg, 1982). Early works attempted to minimize forgetting in neural networks through experience rehearsal (Ratcliff, 1990; Robins, 1995; Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2019) by interleaving samples belonging to previous tasks in the replay buffer with the current batch of samples while training on the new tasks. Simple ER (Ratcliff, 1990; Robins, 1995) is one of the top-performing methods across different CL scenarios (Buzzega et al., 2020). In this work we focus on a specific problem setting where storing large buffers can be expensive. Given such a scenario, ER fails to perform well under low-buffer regimes and longer task sequences.

To preserve information pertaining to previous tasks better, several other methods build on top of ER: Function Distance Regularization (FDR) (Benjamin et al., 2018) used mean-squared error to align current outputs with the past exemplars. Dark-experience Replay (DER) (Buzzega et al., 2020) followed the suit and extended the regularization

Algorithm 1 ER with consistency regularization

> **input:** Data streams $\mathcal{D}_t \ \forall \ t \ \in \{1, ..., T\}$, Model f_θ
> Balancing factors α and β, Memory buffer $\mathcal{D}_m = \{\}$

1: **for all** tasks $t \in \{1, 2, .., T\}$ **do**
2:    **for** minibatch $\{x_i, y_i\}_{i=1}^{B} \in \mathcal{D}_t$ **do**
3:       $\mathcal{L}_{er} = 0$
4:       $\hat{y}_i = f_\theta(x_i)$
5:       Compute $\mathcal{L}_{er} \ += \frac{1}{B} \sum_B \mathcal{L}_{ce}(\hat{y}_i, y_i)$
6:       **if** $\mathcal{D}_m \neq \emptyset$ **then**
7:          **for** minibatch $\{x_j, y_j, z_j\}_{j=1}^{B} \in \mathcal{D}_m$ **do**
8:             $\hat{y}_j = f_\theta(x_j)$
9:             Compute $\mathcal{L}_{er} \ += \frac{\alpha}{B} \sum_B \mathcal{L}_{ce}(\hat{y}_j, y_j)$
10:         **for** minibatch $\{x_j, y_j, z_j\}_{j=1}^{B} \in \mathcal{D}_m$ **do**
11:             $\hat{z}_j = f_\theta(x_j)$
12:             Compute $\mathcal{L}_{cr}$
13:       Compute $\mathcal{L} = \mathcal{L}_{er} + \beta \mathcal{L}_{cr}$
14:       Compute the gradients $\frac{\delta \mathcal{L}}{\delta \theta}$
15:       Update the model f_θ
16:       Update the memory buffer $\mathcal{D}_m$
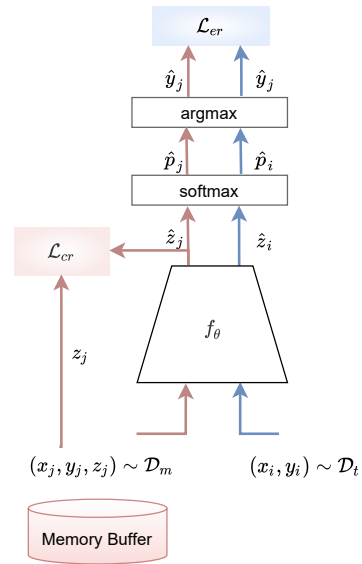17: **return** model f_θ

Figure 2: Schematic diagram of the framework. In addition to ER objective ($\mathcal{L}_{er}$), we penalize the model for deviation from its past predictions through consistency regularization ($\mathcal{L}_{cr}$).

to different CL scenarios. Inspired by the complementary-learning system (CLS) theory (Kumaran et al., 2016) in the brain, several methods include fast and slow learning networks to consolidate the knowledge better: CLS-ER (Arani et al., 2022) used dual memory experience replay to maintain short-term and long-term semantic memories. Despite differences in their approach and underlying architecture, consistency regularization unifies them under one hood: these approaches regularize consistency in predictions of buffered samples across tasks thereby preserving rich information about the previous tasks.

As the training progresses, soft-targets carry more information per training sample than the hard targets. Although this has little influence on the cross-entropy objective, it captures the rich similarity structure of the data (Hinton et al., 2015). Therefore, consistency regularization on soft-targets has been a widely used technique in many domains including semi-supervised learning, self-supervised learning, and knowledge distillation (Bachman et al., 2014; Zhai et al., 2019; Chen et al., 2020; Hinton et al., 2015). In knowledge distillation, several approaches (Li & Hoiem, 2017; Rebuffi et al., 2017) employ past version of the CL model as a teacher and distill knowledge to the current model. Learning without forgetting (LwF) (Li & Hoiem, 2017) reduces the feature drift by smoothening the predictions at the beginning of each task without employing the memory buffer. In semi-supervised learning, the core idea is simple: input image is perturbed in semantic-preserving ways and the model's sensitivity to perturbations is penalized. Consistency regularizer forces the model to learn representations invariant to semantic-preserving perturbations. These perturbations can manifest in many ways: they can be augmentations such as random cropping, Gaussian noise, colorization, or even adversarial attacks. The regularization term generally in semi-supervised learning is either mean-squared error (Sajjadi et al., 2016) between model's output of perturbed and non-perturbed images or KL-divergence (Miyato et al., 2018) between distribution over classes implied by the logits. However, it is difficult to gauge the role of consistency regularization with a limited number of regularizers. On the other hand, SSL objectives (Chen et al., 2020; He et al., 2020) implicitly impose a loose form of consistency: they enforce the representations of the multiple augmented views to be close by, not necessarily restricting them to be the same in the latent space. Although they have been used for pre-training, these methods have not been used as explicit consistency regularizers. Moreover, a diverse family of SSL functions provides design flexibility to better understand the role of consistency regularization.

Consistency regularization is relatively new to CL and has not been well explored. Therefore, we study the effectiveness of consistency regularization in CL and investigate its role in mitigating catastrophic forgetting under a common framework.

3 EXPERIENCE REPLAY WITH CONSISTENCY REGULARIZATION

CL normally consists of T sequential tasks indexed by $t \in \{1, 2, ..., T\}$. During each task, N_t input samples and the corresponding labels $\{(x_i, y_i)\}_{i=1}^{N_t}$ are drawn from the task data distribution $\mathcal{D}_t$. The CL model f_θ consists of a backbone network (such as ResNet-18 (He et al., 2016)) and a linear classifier representing all classes. The model f_θ is sequentially optimized on data stream of one task with the cross-entropy objective function. CL is especially challenging since the data from the previous tasks are unavailable, i.e at any point during training the model f_θ has access to the current data distribution $\mathcal{D}_t$ alone. As the cross-entropy objective is solely optimized for the current task, plasticity overtakes stability resulting in overfitting on the current task and catastrophic forgetting of older tasks.

ER sought to address this problem by storing a subset of training data from previous tasks (into a memory buffer) and replaying them alongside $\mathcal{D}_t$:

$$\mathcal{L}_{er} \triangleq \underset{(x_i, y_i) \sim \mathcal{D}_t}{\mathbb{E}} [\, \mathcal{L}_{ce}(\hat{y}_i, y_i) \,] + \alpha \underset{(x_j, y_j) \sim D_m}{\mathbb{E}} [\, \mathcal{L}_{ce}(\hat{y}_j, y_j) \,] \tag{1}$$

where $\mathcal{L}_{ce}$ is a cross-entropy objective and α is a balancing factor. $\mathcal{D}_m$ represents the distribution of samples stored in the memory buffer. We use reservoir sampling (Vitter, 1985) to populate the memory buffer in order not to rely on the task boundaries. Reservoir sampling selects random samples with equal probability from the input stream throughout the CL training, is suited for general CL without task boundary information. ER partially improves the stability-plasticity dilemma through twin objectives: supervisory signal from $\mathcal{D}_t$ improves plasticity while that from $\mathcal{D}_m$ ameliorates the stability, thus partially addressing catastrophic forgetting. In practice, only a limited number of samples are stored in the memory buffer owing to budget constraints ($|D_t| \gg |D_m|$). As the performance of ER is commensurate with the buffer size, it is quintessential to derive as much information as possible about the previous tasks from limited buffer samples.

Consistency regularization plays a pivotal role in approximating the past behaviour by enforcing consistency across time-separated current and past predictions of buffered samples. Through consistency, rich learned information about the previous tasks can be better reinforced resulting in less forgetting. In addition, enforcing consistency between the corresponding pre-softmax responses has a beneficial effect: it avoids information loss occurring due to softmax operation. Therefore, the final learning objective of ER with consistency regularization is defined as follows:

$$\mathcal{L} = \mathcal{L}_{er} + \beta \mathcal{L}_{cr} \tag{2}$$

where β is a balancing factor and $\mathcal{L}_{cr}$ represents any consistency regularizer on the predictions. Figure 2 provides a schematic diagram of the ER framework with consistency regularization. The detailed training regime is in Algorithm 1. In the next section, we provide details about the various choices for $\mathcal{L}_{cr}$.

4 CONSISTENCY REGULARIZERS

Consistency regularization can be straightforwardly defined with the expected Minkowski distance (L_p) between the corresponding pairs of predictions:

$$\mathcal{L}_{cr} \triangleq \underset{(x_j, y_j, z_j) \sim D_m}{\mathbb{E}} \| \hat{z}_j - z_j \|_p \tag{3}$$

where z_j represents model's pre-softmax responses stored in the buffer. Squared-L_2 distance can also be viewed as Mean-Squared Error (MSE), up to a constant factor. Under mild assumptions (Hinton et al., 2015), the optimization of the KL-divergence is equivalent to minimizing the Euclidean distance between the corresponding logits. Therefore, $\mathcal{D}_{KL}(\hat{p}_j \| p_j)$ can also be used in place of L_p in Equation 3.

Enforcing consistency in CL is akin to solving a pretext task of bringing current and past exemplar outputs closer in the representational space. It can be simply achieved through maximizing mutual information $I(\hat{z}_j, z_j)$ as follows:

$$\mathcal{L}_{cr} \triangleq - \sum_{\hat{z}_j} \sum_{z_j} Q_{\hat{j}j} \ln \frac{Q_{\hat{j}j}}{Q_{\hat{j}} Q_j} \tag{4}$$

where the conditional joint distribution can be approximated through $Q_{\hat{j}j} = p(\hat{z}_j, z_j | x_j) = Q(\hat{p}_j) \cdot Q(p_j)$. The marginal distributions $Q_{\hat{j}} = Q(\hat{p}_j)$ and $Q_j = Q(p_j)$ can be obtained by summing over rows and columns of $Q_{\hat{j}j}$ matrix. With more tasks and more classes per task, estimating mutual information in high dimensional space becomes a notoriously difficult task, and in practice one often maximizes a tractable lower bound on this quantity (Poole et al.,

2019). InfoNCE (Oord et al., 2018), based on Noise Contrast Estimation (Gutmann & Hyvärinen, 2012), is one such lower bound that has been shown to work well in practice:

$$I(\hat{z}_j, z_j) \geq \underset{q(\hat{z}_j, z_j)}{\mathbb{E}} [< \hat{z}_j, z_j > - \underset{q(Z)}{\mathbb{E}} log \sum_{z_k \in Z} exp < \hat{z}_j, z_k >] + log|Z| \tag{5}$$

where $z_k \in Z \setminus \{\hat{z}_j, z_j\}$ is a set of negatives and $< ., . >$ computes cosine similarity. InfoNCE has recently been popular among SSL methods (Chen et al., 2020; He et al., 2020). Courting regularization of consistency in model's predictions as an auxiliary self-supervised pretext task provides the design flexibility as $\mathcal{L}_{cr}$ can be replaced with any state-of-the art SSL method in our framework. Table 1 presents the overview of regularizers along with the state-of-the-art SSL method under four broad categories. This lets us examine different families of functions as regularizers expanding the ambit of our forthcoming analyses. Implementation details of these approaches are in Appendix A.2.

Table 1: Overview of various consistency regularizers used in this work.

Category		Methods
Minkowski distance functions		L_1, L_2, L_{inf} $MSE = \frac{1}{N}L_2^2$
Kullback-Leibler Divergence		$\mathcal{D}_{KL}(\hat{p_j} \parallel p_j)$
SSL methods	InfoNCE	SimCLR
	Mean squared error	BYOL
	Cross-Entropy	DINO
	Cross-Correlation	Barlow Twins

5 EXPERIMENTAL SETUP

Following (Van de Ven & Tolias, 2019), we evaluate consistency regularization added to ER on class incremental learning (Class-IL), domain incremental learning (Domain-IL), and MNIST-360 (Buzzega et al., 2020) scenarios. In Class-IL, the CL model encounters a new set of classes in each task and learns to distinguish all classes encountered so far after each task. In Domain-IL however, the number of classes remains the same across subsequent tasks, but the input distribution is shifted. MNIST-360 exposes the CL model to both sharp class distribution shift and smooth rotational distribution shift in MNIST-360. More information about the datasets is detailed in Appendix A.1.

We build upon Mammoth (Buzzega et al., 2020) CL repository in PyTorch. We employ ResNet-18 for Class-IL and a 2-layer fully-connected network of 100 neurons each with ReLU activation for Domain-IL and MNIST-360 tasks. To ensure a fair comparison between different regularizers, the training settings have been standardized: backbone, number of epochs training for each task, batch size, reservoir buffer and buffer size have been kept the same for all methods along with the results averaged over three random seeds. We consider L_1, L_2, and L_{inf} from the Minkowski distance family, MSE, KL-Div, MI, and four different SSL algorithms SimCLR, BYOL, DINO, and Barlow Twins (Table 1) as consistency regularizers in addition to ER. Since not all regularizers are used in their original form, we use a shorthand notation (e.g. +SimCLR) to refer to them being used as a regularizer on top of ER. Implementation details of these approaches can be found in Appendix A.2. In the empirical results, we also provide a lower bound *SGD*, without any help to mitigate catastrophic forgetting, and an upper bound *Joint*, where training is done using entire dataset.

6 RESULTS

Table 2 provides the effect of various consistency regularizers added to ER in different CL scenarios. Our results for the Class-IL scenario show that:

(i) Across all datasets, regularizing consistency in predictions is almost always better than vanilla ER. Consistency helps in preserving the rich information about the previous tasks better thereby further mitigating forgetting.

(ii) The improvement in performance is more significant in more complex datasets. For example in S-TinyImageNet, the final Top-1 Accuracy with $+L_{inf}$ is almost double that of vanilla ER for the buffer size of 500.

(iii) Reinforcing our earlier hypothesis, consistency regularization in ER can indeed be cast as a self-supervised pretext task enabling the use of state-of-the-art SSL methods in CL. SSL methods assist ER in mitigating forgetting in the majority of the experiments.

(iv) Among SSL regularizers, cross-correlation-based and MSE-based methods perform comparably well. In Barlow Twins, on-diagonal elements in the cross-correlation matrix carry a higher weight resulting in a stronger form of consistency among predictions. Maximizing cosine similarity between predictions using

Table 2: Comparison of Top-1 Accuracy (%) of vanilla ER and ER plus various consistency regularization under different challenging CL scenarios. Best result is in bold and the second best is underlined.

Buffer size	Method	Class-IL			Domain-IL	
		S-CIFAR-10	S-CIFAR-100	S-TinyImageNet	R-MNIST	MNIST-360
-	Joint	92.20 ± 0.15	70.56 ± 0.28	59.99 ± 0.19	96.52 ± 0.12	82.05 ± 0.62
	SGD	19.62 ± 0.05	17.49 ± 0.28	07.92 ± 0.26	70.76 ± 5.61	21.09 ± 0.21
200	ER	48.19 ± 1.37	21.40 ± 0.22	8.57 ± 0.04	82.09 ± 3.32	51.76 ± 2.19
	$+L_1$	$\mathbf{65.33} \pm 1.24$	28.74 ± 1.92	10.29 ± 0.64	86.99 ± 0.60	57.28 ± 2.04
	$+L_2$	63.04 ± 1.44	29.72 ± 1.16	9.32 ± 0.38	89.72 ± 0.19	$\mathbf{61.68} \pm 1.17$
	$+L_{inf}$	$\underline{63.77} \pm 1.25$	$\mathbf{32.28} \pm 2.52$	$\mathbf{13.09} \pm 0.47$	$\underline{90.01} \pm 0.13$	$\underline{61.03} \pm 1.57$
	+KL-Div	57.78 ± 1.42	21.86 ± 0.25	8.46 ± 0.27	84.69 ± 0.70	50.28 ± 0.59
	+MSE	65.12 ± 1.61	29.60 ± 1.14	11.14 ± 0.63	$\mathbf{90.78} \pm 0.22$	59.04 ± 1.69
	+MI	59.56 ± 1.35	22.49 ± 0.28	8.45 ± 0.24	86.37 ± 0.10	50.84 ± 1.74
	+SimCLR	60.19 ± 2.54	24.75 ± 0.6	10.16 ± 0.33	85.79 ± 0.99	50.66 ± 2.23
	+BYOL	63.57 ± 0.63	$\underline{31.23} \pm 1.45$	12.15 ± 0.47	89.26 ± 0.14	54.42 ± 2.94
	+DINO	63.03 ± 2.08	26.21 ± 0.55	9.87 ± 0.43	86.78 ± 0.72	47.16 ± 2.36
	+Barlow Twins	62.91 ± 1.01	26.51 ± 0.78	$\underline{12.20} \pm 0.24$	88.65 ± 0.37	53.02 ± 0.91
500	ER	60.93 ± 1.50	28.02 ± 0.31	10.08 ± 0.21	88.38 ± 1.54	63.78 ± 2.70
	$+L_1$	$\mathbf{74.39} \pm 0.54$	40.40 ± 1.31	15.59 ± 0.17	89.96 ± 0.39	68.77 ± 1.32
	$+L_2$	72.95 ± 0.74	40.92 ± 0.95	13.77 ± 0.30	91.79 ± 0.63	$\mathbf{73.32} \pm 0.30$
	$+L_{inf}$	$\underline{73.83} \pm 0.35$	$\mathbf{42.15} \pm 1.88$	$\mathbf{20.47} \pm 0.99$	$\underline{92.22} \pm 0.62$	70.89 ± 0.70
	+KL-Div	70.50 ± 0.48	27.91 ± 0.98	10.22 ± 0.17	88.38 ± 1.52	65.50 ± 2.84
	+MSE	73.60 ± 0.72	$\underline{41.40} \pm 0.96$	18.04 ± 1.28	$\mathbf{92.74} \pm 0.77$	$\underline{71.64} \pm 0.55$
	+MI	70.95 ± 1.16	29.21 ± 1.33	10.54 ± 0.33	88.75 ± 1.36	66.88 ± 0.69
	+SimCLR	74.26 ± 0.43	31.26 ± 0.15	13.93 ± 0.48	89.12 ± 1.43	66.90 ± 1.56
	+BYOL	73.56 ± 0.21	38.74 ± 2.43	$\underline{18.43} \pm 0.77$	91.17 ± 0.42	66.87 ± 0.55
	+DINO	72.14 ± 0.70	35.33 ± 0.78	14.19 ± 0.69	88.08 ± 0.41	62.41 ± 3.23
	+Barlow Twins	73.23 ± 1.28	35.23 ± 1.73	18.05 ± 0.26	91.08 ± 0.67	67.03 ± 1.00

BYOL in our framework translates to minimizing squared error when the predictions are L_2 normalized (Appendix A.3). Therefore, +BYOL's performance closely follows that of +MSE's.

(v) All regularizers perform equally well on S-CIFAR-10. However, approximate methods (e.g., +MI, +Sim-CLR, +DINO) struggle to keep up with the rest of regularizers as number of classes increases. These methods do not fully mimic the past behaviour but approximate it by bringing the corresponding predictions closer in the latent space creating a knowledge gap. This knowledge gap is especially more in high-dimensional spaces leading to poor performance. We argue that stricter consistency constraints such as $+L_p$ and +MSE are more successful to close this knowledge gap.

(vi) Similarly, +KL-Div fails miserably in higher dimensional spaces. We think this is due to the loss of information occurring in probability space due to the squashing function (e.g., softmax) (Buzzega et al., 2020).

(vii) The Chebyshev distance $+L_{inf}$, a simple form of regularization, is surprisingly one of the top performers in Class-IL scenario across all datasets. As ER has a high task recency bias (see Section 7.4), the current predictions are skewed towards current task logits. However, the past predictions are skewed towards corresponding past task logits. As $+L_{inf}$ considers only the greatest difference along any coordinate dimension, we suspect that maximum difference lies either along the current task logits or corresponding past task logits. Therefore, $+L_{inf}$ penalizes the task recency bias and aligns past and current predictions along past task logits' dimensions in subsequent training iterations. Due to this, $+L_{inf}$ is able to retain its top-notch performance even in high-dimensional spaces.

The benefits of maintaining consistency in predictions is not only limited to the Class-IL scenario. R-MNIST under Domain-IL scenario requires the CL model to classify all MNIST digits for 20 subsequent tasks, for rotated images by a random angle in the interval $[0, \pi)$. In order to perform well, models need to enhance their positive forward transfer all the while lessening interference. Our earlier inferences can well be extrapolated here: using consistency regularization is almost always beneficial to preserve rich information about the previous tasks thereby enhancing

Table 3: Comparison of Top-1 Accuracy (%) on S-CIFAR-100 under longer task sequences. Best result is in bold and the second best is underlined.

Method	5 tasks	10 tasks	20 tasks
ER	$28.02_{\pm0.31}$	$21.49_{\pm0.47}$	$16.52_{\pm0.86}$
$+L_1$	$40.40_{\pm1.31}$	$35.75_{\pm0.25}$	$\mathbf{25.30}_{\pm6.40}$
$+L_2$	$40.92_{\pm0.95}$	$34.32_{\pm0.49}$	$23.51_{\pm4.71}$
$+L_{inf}$	$\mathbf{42.15}_{\pm1.88}$	$\mathbf{38.03}_{\pm0.81}$	$\underline{25.05}_{\pm7.00}$
+KL-Div	$27.91_{\pm0.98}$	$22.38_{\pm1.13}$	$15.32_{\pm5.40}$
+MSE	$\underline{41.40}_{\pm0.96}$	$\underline{36.20}_{\pm0.52}$	$22.25_{\pm5.87}$
+MI	$29.21_{\pm1.33}$	$22.73_{\pm1.39}$	$17.40_{\pm2.93}$
+SimCLR	$31.26_{\pm0.15}$	$25.89_{\pm0.99}$	$19.48_{\pm4.55}$
+BYOL	$38.74_{\pm2.43}$	$35.84_{\pm0.96}$	$22.99_{\pm4.42}$
+DINO	$35.33_{\pm0.78}$	$30.24_{\pm1.43}$	$20.49_{\pm4.37}$
+Barlow Twins	$35.23_{\pm1.73}$	$31.07_{\pm1.50}$	$18.79_{\pm5.94}$

Table 4: Comparison of Top-1 Accuracy (%) on S-CIFAR-10 under extremely low-buffer sizes. Best result is in bold and the second best is underlined.

10	20	50	100
$22.20_{\pm1.36}$	$26.69_{\pm1.21}$	$32.51_{\pm1.77}$	$41.10_{\pm1.10}$
$28.55_{\pm2.58}$	$\underline{34.77}_{\pm3.59}$	$\underline{47.42}_{\pm2.20}$	$\mathbf{57.92}_{\pm1.47}$
$26.00_{\pm6.01}$	$33.97_{\pm5.30}$	$45.86_{\pm3.14}$	$55.69_{\pm1.00}$
$27.91_{\pm3.19}$	$33.32_{\pm5.19}$	$\mathbf{47.84}_{\pm2.39}$	$\underline{57.67}_{\pm0.60}$
$21.93_{\pm0.95}$	$25.06_{\pm1.39}$	$35.74_{\pm2.50}$	$47.93_{\pm2.04}$
$\mathbf{30.21}_{\pm1.78}$	$32.55_{\pm1.90}$	$45.27_{\pm2.34}$	$56.09_{\pm0.87}$
$23.51_{\pm0.34}$	$29.10_{\pm1.48}$	$36.13_{\pm3.48}$	$50.19_{\pm0.74}$
$27.72_{\pm2.63}$	$28.54_{\pm3.02}$	$42.61_{\pm2.91}$	$51.91_{\pm0.19}$
$\underline{29.04}_{\pm1.57}$	$\mathbf{35.40}_{\pm2.59}$	$43.62_{\pm3.40}$	$55.73_{\pm2.52}$
$24.96_{\pm0.98}$	$34.36_{\pm1.14}$	$39.65_{\pm1.48}$	$52.60_{\pm1.33}$
$25.24_{\pm4.40}$	$33.54_{\pm3.43}$	$42.92_{\pm3.56}$	$54.48_{\pm1.55}$

positive forward transfer. $+L_p$ family of regularizers are top performers while approximate methods doing slightly worse. Among SSL algorithms, +BYOL and +Barlow Twins perform equally well compared to top performers.

MNIST-360 is a new protocol developed to address the general continual learning desiderata (Buzzega et al., 2020). It models a stream of two consecutive MNIST digits ({0,1}, {1,2}, ..., {7,8}) with increasing rotation applied in subsequent batches. It is worth noting that such a setting offers both a sharp shift in terms of classes and a smooth shift in terms of rotation. MNIST-360 involves recurring rotated digits in subsequent sequences which makes the transfer of knowledge from previous occurrences important. As is the case in earlier scenarios, $+L_p$ consistency regularization significantly augments the ER by effectively preserving the previous knowledge. The difference in performance between stricter and approximate consistency regularizers in this scenario is due to the additional complexity added by the rotation. $+L_p$ regularizers leverage rich information about the past better when revisiting the previous tasks and outperform the ER across different buffer sizes.

It is indeed clear from the above analyses under different CL scenarios and datasets that consistency regularization plays a prominent role in alleviating the catastrophic forgetting in ER. Enforcing consistency in predictions separated through time helps in preserving rich information about the previous tasks better thereby helping to further mitigate the catastrophic forgetting in CL.

7 MODEL ANALYSIS

7.1 EFFECT OF LONGER TASK SEQUENCES AND LOW-BUFFER REGIMES

Catastrophic forgetting worsens as the number of tasks in a sequence increases, referred to as long-term catastrophic forgetting (Peng et al., 2021). The number of samples in the buffer representing each previous task drastically reduces in longer task sequences resulting in poor performance for ER. As the number of classes are low in S-CIFAR-10, we consider S-CIFAR-100 for this analysis. Table 3 shows the performance of models with 5, 10, and 20 task sequences on S-CIFAR-100 with a fixed buffer size of 500. Our analysis of Class-IL can straightforwardly be extrapolated here: Except +KL-Div, all regularizers augment ER in preserving rich previous task information in longer task sequences. As was the case earlier, stricter consistency constraints (e.g., $+L_p$, +MSE) enable better performance than approximate methods. $+L_{inf}$ brings a relative improvement of at least 50% across all task sequences. Although SSL methods lag behind $+L_{inf}$, they still outperform ER with a good margin.

Table 4 presents the performance under extremely low-buffer sizes. As S-CIFAR-10 consists of 10 classes, extreme low-buffer regimes (e.g. 10, 20) may not even have representative samples from all classes in the buffer due to Reservoir Sampling. Under low buffer regimes, ER learns discriminative features specific to buffered samples thereby inaccurately approximating the previous task data distributions. These learned features are not representative of the past tasks and do not generalize well. Since soft-targets capture additional information with regard to the similarity structure of the data, regularizing consistency in predictions of soft-targets preserves the previous task information better compared to vanilla ER.

Consistency regularization brings a significant performance boost to ER even under extremely low-buffer sizes and longer task sequences by mitigating catastrophic forgetting all the while reducing the reliance on buffer size. Therefore,

	mRA		Noise					Blur						Weather					Digital			
		Gaussian	Impulse	Shot	Speckle	Defocus	Gaussian	Glass	Motion	Zoom	Brightness	Fog	Frost	Snow	Spatter	Contrast	Elastic	JPEG	Pixelate	Saturate
ER	18.2	22.3	20.6	22.9	22.4	14.4	13.2	13.1	11.2	11.6	25.3	12.4	18.4	25.2	24.9	11.3	13.4	20.1	18.5	25.3
+Barlow Twins	21.2	22.8	22.1	23.5	22.9	19.2	17.6	16.2	15.9	16.3	29.1	16.0	21.3	24.6	26.4	12.9	18.8	23.3	24.9	30.0
+KL-Div	21.7	26.5	23.9	27.1	26.5	17.1	15.3	15.7	13.7	13.4	29.9	14.7	22.9	28.9	28.9	12.1	16.6	24.1	24.6	30.6
+SimCLR	21.9	23.7	23.0	25.1	24.9	17.6	16.1	17.9	15.3	14.6	31.5	14.7	23.7	28.2	28.7	13.4	17.2	23.2	26.4	31.2
+BYOL	22.4	24.0	23.2	25.6	25.5	20.4	19.0	18.1	17.7	17.2	30.6	16.3	21.6	25.4	27.9	13.8	19.8	23.9	24.3	31.7
+MSE	22.5	25.3	23.8	26.5	26.2	18.6	16.9	17.9	15.1	15.5	30.8	15.8	23.2	28.6	29.6	14.0	18.5	24.7	24.6	31.5
+L_1	22.5	27.0	25.3	27.8	27.3	17.5	16.0	17.3	14.6	14.0	30.3	16.3	23.9	28.9	29.2	13.9	17.5	25.1	24.9	30.6
+L_{inf}	22.6	25.1	23.4	26.4	26.1	19.4	17.6	17.3	15.8	16.0	31.1	15.9	23.4	27.9	28.3	13.4	19.4	25.5	26.2	31.5
+MI	23.1	23.5	22.9	25.4	25.2	20.4	18.8	15.6	16.9	17.1	32.7	17.5	23.9	29.0	29.9	14.8	19.6	26.0	26.2	33.0
+L_2	25.3	28.9	27.1	30.0	29.4	21.6	19.6	19.9	18.6	17.9	34.0	18.1	26.4	30.9	31.6	15.1	21.3	28.2	28.4	34.0
+DINO	25.4	28.3	26.4	29.6	29.1	21.6	19.9	19.7	17.5	18.1	34.6	18.8	26.4	31.8	32.7	17.0	20.8	28.1	27.9	34.4

Figure 3: Robustness to common natural input corruptions on noise, blur, weather, and digital categories as Top-1 Accuracy (%). Methods are sorted in the ascending manner based on their mean robust accuracy (mRA).
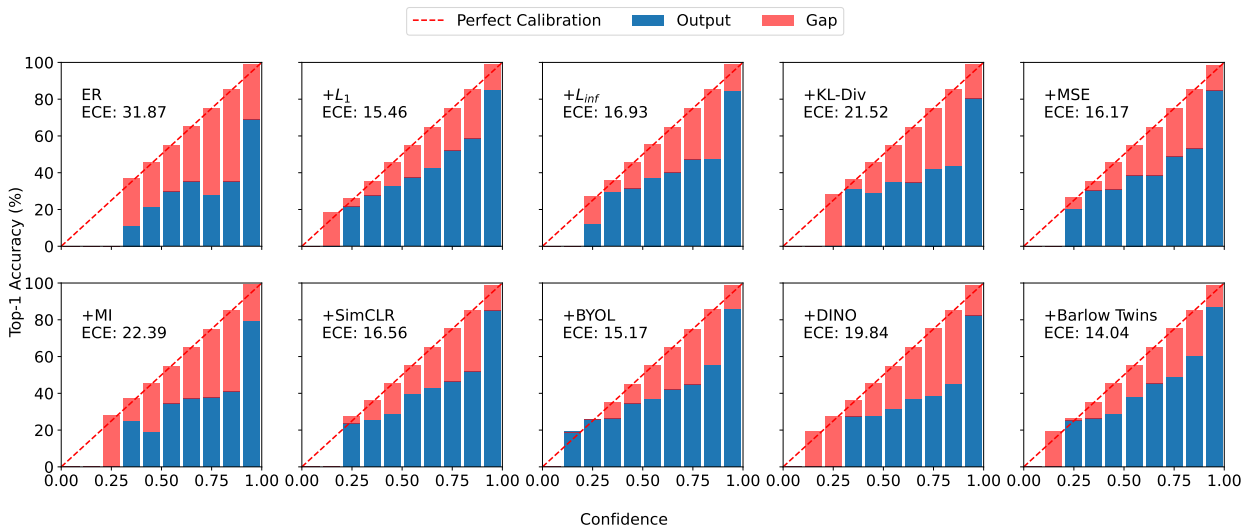

Figure 4: Confidence estimates and corresponding Expected Calibration Error (ECE) of CL models trained on S-CIFAR-10 with buffer size 500. Lower ECE indicates more reliability.

consistency regularization is the need of the hour for rehearsal-based methods to preserve rich information about the previous tasks better.

7.2 ROBUSTNESS TO IMAGE CORRUPTIONS

Distributional shift often influenced by weather and illumination changes is widely prevalent in the real world. DNNs have been shown to be sensitive to distributional shifts such as common input corruptions (Hendrycks & Dietterich, 2018). Consequently, CL agents operating in high-stakes applications such as autonomous driving need to be robust to a variety of such distributional shifts. In this section, we evaluate the effect of added consistency regularizers to ER against common synthetic image corruptions using CIFAR-10-C (Hendrycks & Dietterich, 2018). CL models are trained on S-CIFAR-10 (buffer size 500) and evaluated on the CIFAR-10-C validation set. Figure 3 presents the robust accuracy of CL models on 15 different out-of-distribution datasets under noise, blur, weather, and digital categories. ER is least robust while all regularizers on top of ER are more robust to induced natural corruptions. As ER relies solely on hard targets to approximate the past behaviour, it learns features specific to buffered samples. Therefore, learned representations do not generalize well to out-of-distribution datasets. As past predictions hold rich information about previous tasks, the regularization scheme enforces ER to learn generalizable features making it more robust to distributional shifts.
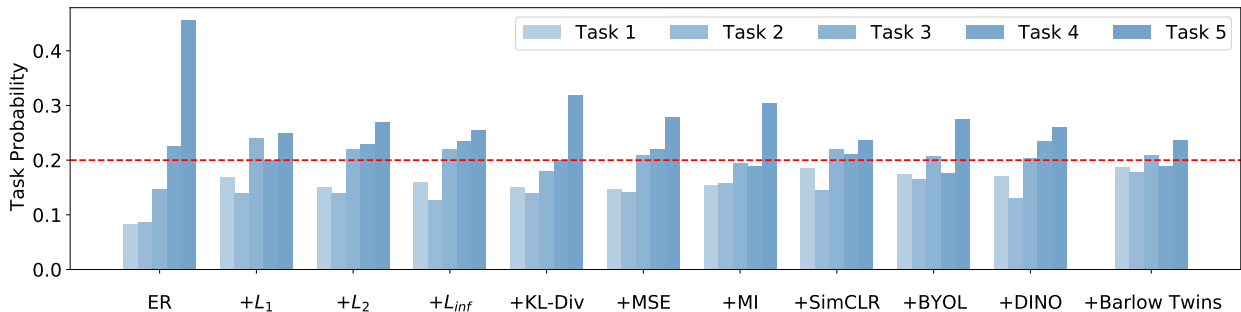
Figure 5: Average task probabilities of CL models trained on CIFAR-10 with 500 buffer size. Within each bar group, right most bar indicates the most recent task.

7.3 CONFIDENCE CALIBRATION

A well-calibrated model improving reliability is quintessential for safety-critical CL systems. Confidence calibration entails the problem of predicting probability estimates representative of the true correctness likelihood. Miscalibration can be defined as the difference in expectation between confidence and accuracy (Guo et al., 2017). Expected Calibration Error (ECE) (Appendix E) approximates the miscalibration in classification by partitioning the predictions into equal-sized bins and computes the difference between the weighted average of the bins' accuracy and confidence (Naeini et al., 2015). A lower ECE value represents better calibration in underlying models. Figure 4 shows ECE along with a reliability diagram on S-CIFAR-10 using a calibration framework (Kuppers et al., 2020). ER is highly miscalibrated and far more overconfident than models trained using a regularizer. On the other hand, all added regularizers to ER improve the model calibration. Stricter consistency constraints have better calibration, even reducing the ECE score by half in most cases. Consistency regularization ensures that predicted softmax scores are better indicators of the actual likelihood of a correct prediction thereby improving the reliability of ER in real-world applications.

7.4 TASK RECENCY BIAS

Replay-based methods are prone to task recency bias - the tendency of a CL model to be biased towards classes from the most recent tasks (Masana et al., 2020; Hou et al., 2019). Specifically, the model sees only a few samples from the old tasks while aplenty from the most recent task, leading to decisions biased towards new classes and the confusion among old classes. Figure 5 presents the the normalized probabilities of each task of a S-CIFAR-10 trained model, computed by averaging probabilities of all samples belonging to the associated classes in a Class-IL setting. The predictions in vanilla ER are biased mostly towards recent tasks, with most recent task being almost $4X$ as much as the first task. Therefore, the predictions stored in the buffer are completely biased towards their corresponding task logits. Since consistency regularization penalizes any deviation in predictions, task recency bias is mitigated as a by-product.

8 CONCLUSION

We provided a comprehensive study on the role of consistency regularization under a common ER framework across various CL scenarios and datasets. We also studied the state-of-the-art SSL algorithms as regularizers by interpreting the regularization of consistency in predictions as a self-supervised pretext task. Consistency regularization augments ER in preserving rich information about the previous tasks better thereby mitigating catastrophic forgetting further. Even under extremely low-buffer sizes and longer task sequences, it yields discernible performance improvement. Stricter consistency constraints are better in mitigating forgetting compared to other approximate methods considered in this study. Minkowski distance functions, extremely simple and efficient, are surprisingly effective in alleviating catastrophic forgetting. Additionally, regularizing consistency in predictions leads to more robust and well-calibrated model. It also mitigates the task recency bias thereby leading to more evenly distributed task prediction probabilities. Therefore, regularizing consistency in predictions is crucial for wholesome improvement of ER. We believe that the representation-replay and generative-replay based methods can also benefit from the analysis conducted in this paper. We hope that this study further spurs research in advancing CL towards sample efficient experience replay.

REFERENCES

Elahe Arani, Fahad Sarfraz, and Bahram Zonooz. Learning fast, learning slow: A general continual learning method based on complementary learning system. In *International Conference on Learning Representations*, 2022.

Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. *Advances in neural information processing systems*, 27, 2014.

Ari Benjamin, David Rolnick, and Konrad Kording. Measuring and regularizing networks in function space. In *International Conference on Learning Representations*, 2018.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. In *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, 2020.

Lucas Caccia, Eugene Belilovsky, Massimo Caccia, and Joelle Pineau. Online learned continual compression with adaptive quantization modules. In *International Conference on Machine Learning*, pp. 1240–1250. PMLR, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9650–9660, October 2021.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. In *ICLR*, 2019.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758, 2021.

MohammadReza Davari, Nader Asadi, Sudhir Mudur, Rahaf Aljundi, and Eugene Belilovsky. Probing representation forgetting in supervised and unsupervised continual learning. *arXiv preprint arXiv:2203.13381*, 2022.

Sebastian Farquhar and Yarin Gal. Towards robust evaluations of continual learning. *arXiv preprint arXiv:1805.09733*, 2018.

Enrico Fini, Victor G Turrisi da Costa, Xavier Alameda-Pineda, Elisa Ricci, Karteek Alahari, and Julien Mairal. Self-supervised models are continual learners. *arXiv preprint arXiv:2112.04215*, 2021.

Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Pires, Zhaohan Guo, Mohammad Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. In *Neural Information Processing Systems*, 2020.

Stephen Grossberg. How does a brain build a cognitive code? *Studies of mind and brain*, pp. 1–52, 1982.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pp. 1321–1330. PMLR, 2017.

Michael U Gutmann and Aapo Hyvärinen. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13(2), 2012.

Demis Hassabis, Dharshan Kumaran, Christopher Summerfield, and Matthew Botvinick. Neuroscience-inspired artificial intelligence. *Neuron*, 95(2):245–258, 2017. ISSN 0896-6273. doi: https://doi.org/10.1016/j.neuron.2017.06.011.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2018.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 831–839, 2019.

Tianyu Hua, Wenxiao Wang, Zihui Xue, Sucheng Ren, Yue Wang, and Hang Zhao. On feature decorrelation in self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9598–9608, 2021.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

A. Krizhevsky. Learning multiple layers of features from tiny images. 2009.

Sneha Reddy Kudugunta, Ankur Bapna, Isaac Caswell, Naveen Arivazhagan, and Orhan Firat. Investigating multilingual nmt representations at scale. *arXiv preprint arXiv:1909.02197*, 2019.

Dharshan Kumaran, Demis Hassabis, and James L McClelland. What learning systems do intelligent agents need? complementary learning systems theory updated. *Trends in cognitive sciences*, 20(7):512–534, 2016.

Fabian Kuppers, Jan Kronenberger, Amirhossein Shantia, and Anselm Haselhoff. Multivariate confidence calibration for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 326–327, 2020.

Ya Le and X. Yang. Tiny imagenet visual recognition challenge. 2015.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

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

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30:6467–6476, 2017.

Divyam Madaan, Jaehong Yoon, Yuanchun Li, Yunxin Liu, and Sung Ju Hwang. Representational continuity for unsupervised continual learning. In *International Conference on Learning Representations*, 2021.

Marc Masana, Xialei Liu, Bartlomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost van de Weijer. Class-incremental learning: survey and performance evaluation on image classification. *arXiv preprint arXiv:2010.15277*, 2020.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pp. 109–165. Elsevier, 1989.

Martial Mermillod, Aurélia Bugaiska, and Patrick Bonin. The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects. *Frontiers in psychology*, 4:504, 2013.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua V Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *arXiv preprint arXiv:1906.02530*, 2019.

Pingbo Pan, Siddharth Swaroop, Alexander Immer, Runa Eschenhagen, Richard Turner, and Mohammad Emtiyaz E Khan. Continual deep learning by functional regularisation of memorable past. *Advances in Neural Information Processing Systems*, 33:4453–4464, 2020.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

Jian Peng, Bo Tang, Hao Jiang, Zhuo Li, Yinjie Lei, Tao Lin, and Haifeng Li. Overcoming long-term catastrophic forgetting through adversarial neural pruning and synaptic consolidation. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2021. doi: 10.1109/TNNLS.2021.3056201.

Quang Pham, Chenghao Liu, and Steven Hoi. Dualnet: Continual learning, fast and slow. *Advances in Neural Information Processing Systems*, 34, 2021.

Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, pp. 5171–5180. PMLR, 2019.

Roger Ratcliff. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*, 97(2):285, 1990.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

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

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Advances in neural information processing systems*, 29, 2016.

Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. In *International Conference on Machine Learning*, pp. 4528–4537. PMLR, 2018.

Michalis K Titsias, Jonathan Schwarz, Alexander G de G Matthews, Razvan Pascanu, and Yee Whye Teh. Functional regularisation for continual learning with gaussian processes. In *ICLR*, 2020.

Gido M Van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019.

Gido M van de Ven, Hava T Siegelmann, and Andreas S Tolias. Brain-inspired replay for continual learning with artificial neural networks. *Nature communications*, 11(1):1–14, 2020.

Jeffrey S Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1): 37–57, 1985.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33:6256–6268, 2020.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*, 2021.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, pp. 3987–3995. PMLR, 2017.

Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1476–1485, 2019.

A IMPLEMENTATION DETAILS

A.1 DATASETS

Following (Buzzega et al., 2020), we evaluate on the following CL scenarios:

Class-IL: The CL model encounters a new set of classes in each task and must learn to distinguish all classes encountered thus far after each task. In practice, we split CIFAR-10 (Krizhevsky, 2009), CIFAR-100 (Krizhevsky, 2009), and TinyImageNet (Le & Yang, 2015) into partitions of 2, 20, and 20 classes per task, respectively.

Domain-IL: The number of classes remain the same across subsequent tasks. However, a task-dependent transformation is applied changing the input distribution for each task. Specifically, R-MNIST (Lopez-Paz & Ranzato, 2017) rotates the input images by a random angle in the interval $[0; \pi)$. R-MNIST requires the model to classify all 10 MNIST (Lecun et al., 1998) digits for 20 subsequent tasks. Permuted-MNIST (Kirkpatrick et al., 2017), on the other hand, applies random permutation to the pixels. Both Rotated and Permuted MNIST require the model to classify all 20 MNIST (Lecun et al., 1998) digits for 20 subsequent tasks. Permuted-MNIST violates cross-task resemblance desiderata, therefore, not considered in our work.

MNIST-360 (Buzzega et al., 2020) models a stream of MNIST data with batches of two consecutive digits at a time. Each sample is rotated by an increasing angle and the sequence is repeated six times. MNIST-360 exposes the CL model to both sharp class distribution shift and smooth rotational distribution shift.

A.2 DERIVATION OF REGULARIZERS

In this section, we provide implementation details of different consistency regularizers.

A.2.1 L_p FAMILY

Derivation of L_p family of loss functions is straight forward. We consider L_1, L_2 and L_{inf} in our comparisons. These variants can be easily obtained by replacing p in Equation 3.

A.2.2 KL-DIVERGENCE

KL-Divergence is a non-symmetric measure of the difference between two probability distributions $P(Z_\theta)$ and $Q(Z_r)$, is closely related to relative entropy, information divergence, and information for discrimination. In order to obtain P and Q, we apply softmax operation on the predictions.

A.2.3 SELF-SUPERVISED LEARNING BASED REGULARIZERS

A broad categorization of leading SSL methods can be found in Table 1. For all these methods, predictions are L_2 normalized before being fed into the respective loss function. We employ one algorithm from each category in our comparison.

We employ SimCLR (Chen et al., 2020) from InfoNCE as in Equation 5. Several approaches have been proposed to reduce the reliance on large negatives in contrastive learning. BYOL (Grill et al., 2020) and SimSiam (Chen & He, 2021) addressed this lacuna through negative-free cosine similarity loss. In our implementation, BYOL can be instantiated as follow:

$$\mathcal{L}_{sc} = 2 - 2\frac{<z_\theta, z_r>}{\|z_\theta\|_2 \ \|z_r\|_2} \tag{6}$$

On the other hand, clustering-based method DINO (Caron et al., 2021)) creates a proxy task through cross-entropy loss. To keep our design simple, we do not use centering operation through momentum update. We simply treat stored predictions as target by applying softmax operation on both predictions. Barlow-Twins (Zbontar et al., 2021) proposed an objective function measuring the cross-correlation matrix and making it as close to the identity matrix as possible. The objective function for Barlow Twins can be defined as:

$$\mathcal{L}_{sc} = \sum_i (1 - C_{ii})^2 + \sum_i \sum_{i \neq j} C_{ij}^2 \tag{7}$$

where C is the cross-correlation matrix computed between the predictions along the batch dimension.

Despite the technical variety, most if not all SSL methods are prone to trivial, collapsed solutions by design (Hua et al., 2021). Learning leads to degenerative solutions when all the representations collapse to a single point in the

representation space. Our framework inherently avoids degenerative solutions: cross-entropy objective makes sure that representations do not collapse to a single point. Therefore, stored representations are scattered across the representation space. Since our framework tries to improve affinity between stored representations and representations generated through current model, all positive pairs cannot collapse to a single point, thereby avoiding total collapse without the helps of ad-hoc techniques.

A.3 Relation to contemporary methods

In this section, we review several works that are closely related to our proposal in CL.

Knowledge distillation: Several approaches employ past version of the CL model as a teacher and distill knowledge to the current model. Learning without forgetting (LwF) (Li & Hoiem, 2017) reduces the feature drift by smoothening the predictions at the beginning of each task without employing the memory buffer. On the other hand, iCaRL (Rebuffi et al., 2017) combines replay with knowledge distillation. iCaRL trains a nearest-mean-of-exemplars classifier using memory buffer as a training set and mitigates catastrophic forgetting through knowledge distillation. In our proposal however, we do not use any previous checkpoints as a teacher. As storing predictions alongside input images has a limited memory footprint, we opt for storing and replaying logits to ensure consistency in predictions.

Function regularization: Similar to our proposal, several methods (e.g. Titsias et al. (2020); Pan et al. (2020); Benjamin et al. (2018); Buzzega et al. (2020)) regularize the change in the function learned by the CL model. Titsias et al. (2020) constrains the neural network predictions from deviating too far from those that solve previous tasks through KL-Divergence. Function Distance Regularization (FDR) (Benjamin et al., 2018) uses past exemplars and model outputs to align past and current outputs. However, FDR stores model outputs at task boundaries. Dark-Experience Replay (DER++) (Buzzega et al., 2020) extended the work in FDR by introducing a *reservoir* thereby eliminating the need for task boundaries without experiencing a drop in performance. Although these approaches use some form of consistency regularization, their effectiveness has not been verified under a common framework.

Latent replay: Consistency regularization in rehearsal-based CL can also be connected to generative-rehearsal based methods (van de Ven et al., 2020; Caccia et al., 2020) with latent representation replay. As an alternative to storing data in the buffer, these methods focus on generating the data to be replayed with a learned generative neural network model of past observations. In addition, they leverage temperature-scaled soft targets for the replayed samples to distill the knowledge. In our case, we use past predictions themselves as a soft targets and enforce the current predictions to be aligned with them.

Intersection of SSL and CL: There have been several works (Madaan et al., 2021; Fini et al., 2021; Davari et al., 2022) that fall in the intersection of SSL and CL. Fini et al. (2021) showed that self-supervised loss functions can be seamlessly converted into distillation losses for CL by adding a predictor network that maps the current state of the representations to their past state. However, this method requires that the past state of the CL model be stored at the end of each task. Pham et al. (2021) employed SSL for consolidating generalizable features in a slow-learning network in a complementary learning system inspired CL setup. Our proposal drastically differs from these approaches: We do not use SSL as a primary learning objective. Instead, SSL objective functions are used to enforce consistency in predictions across time.

B Task performance

We provide detailed task performance throughout the CL training in Figure 6. The CL models are trained on the S-CIFAR10 with the buffer size 500. These results supplement the results presented in Table 2 where only the average of accuracies after last task are presented. As can be seen, Regularizing consistency in predictions consistently augments ER throughout the CL training.

C Model Analysis on S-TinyImageNet

We extend the CL model analysis to S-TinyImageNet in which the models are trained on this data with buffer size 500.

C.1 Model Calibration

Figure 7 presents the reliability diagrams of CL models along with corresponding ECE score. As is the case in Section 7.3, ER is highly miscalibrated. Owing to the increased complexity in S-TinyImageNet, the miscalibration is much more pronounced in ER. Except MI, all other consistency regularizers improve model calibration. Especially, L_1 and
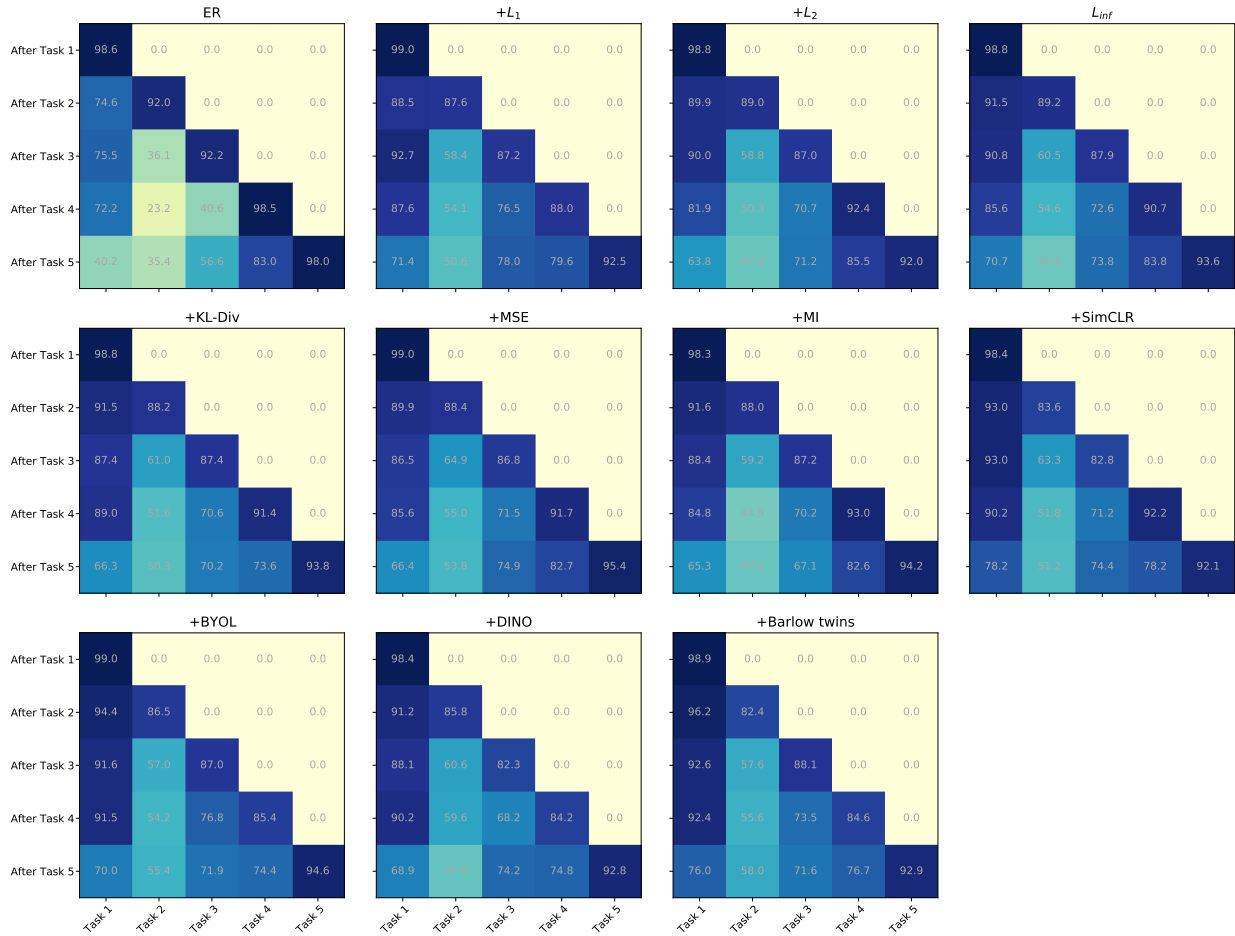
Figure 6: Task performance of different CL models on S-CIFAR-10 with buffer size 500.

L_{inf} greatly improve the model calibration in S-TinyImageNet, reinforcing our earlier finding that stricter consistency constraints assist ER better than approximate counterparts.

C.2 TASK-RECENCY BIAS

Figure 8 presents the the normalized probabilities of each task of a S-CIFAR-10 trained model, computed by averaging probabilities of all samples belonging to the associated classes in a Class-IL setting. The predictions in ER are biased mostly towards recent tasks, with most recent task being almost 0.95. Therefore, the predictions stored in the buffer are completely biased towards their corresponding task logits. Since consistency regularization penalizes any deviation in predictions, recency bias is mitigated as a by-product. Stricter consistency constraints such as $+L_1$, $+L_{inf}$ have a lower bias. Among SSL algorithms, +Barlow Twins and +BYOL have lower task recency bias.

C.3 OVERVIEW OF S-TINYIMAGENET EVALUATION

We provide an overview of relative performance gain of different regularizers trained on S-TinyImageNet with buffer size 500 over ER in Figure 1. We provide relative performance gain in terms of accuracy, task-recency bias and model calibration. The relative gains are computed as follows:

$$Accuracy\ Gain = 100 * (Acc_{cr} - Acc_{er})/Acc_{er} \tag{8}$$

$$Recency\ bias\ Gain = 100 * ((1 - (T10_{cr} - T1_{cr})) - (1 - (T10_{er} - T1_{er})))/(1 - (T10_{er} - T1_{er})) \tag{9}$$

$$Calibration\ Gain = 100 * ((100 - ECE_{cr}) - (100 - ECE_{er}))/(100 - ECE_{er}) \tag{10}$$

where cr refers to performance of any consistency regularizer, $T10$ and $T1$ are average task probabilities of task 10 and task 1.
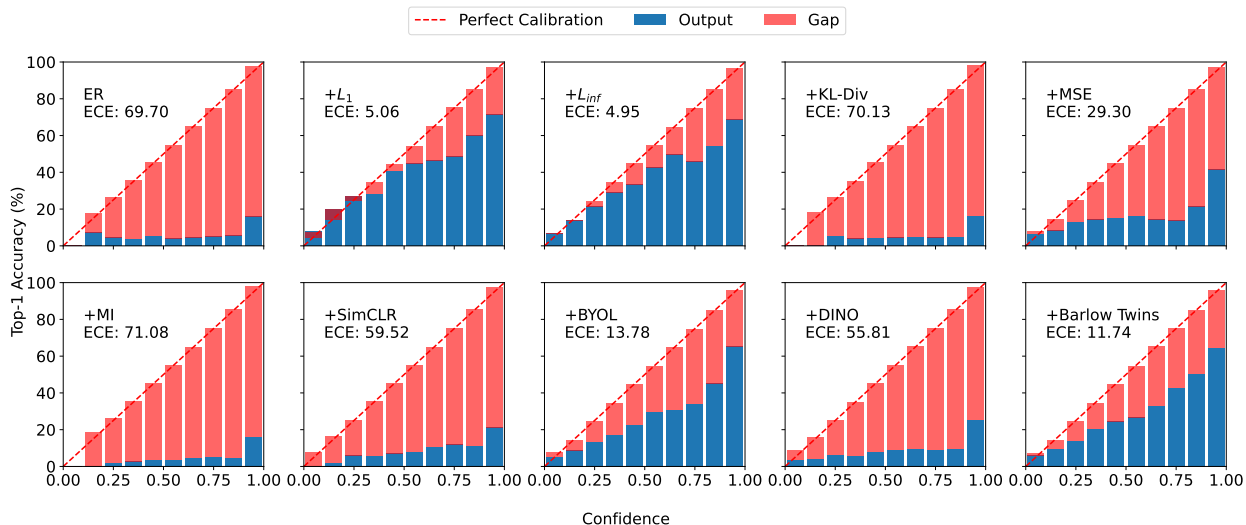
Figure 7: Confidence estimates and corresponding Expected Calibration Error (ECE) of S-TinyImageNet trained CL models. Lower ECE is better.

Figure 8: Average task probabilities of CL models trained on S-TinyImageNet with 500 buffer size. Methods are sorted in descending order of their bias towards the most recent task. ER has the highest task recency bias while stricter consistency constraints have one of the lowest biases.

D FORWARD TRANSFER ON R-MNIST

We report the forward transfer results on R-MNIST experiments presented earlier in Table 2. Following (Lopez-Paz & Ranzato, 2017), we compute forward transfer as a difference between the accuracy just before starting training on a given task and the one of the random-initialized network; it is averaged across all tasks. Forward transfer is relevant for Domain-IL scenarios as long as underlying input transformation is not disruptive. Table 5 shows the forward transfer on R-MNIST for different buffer sizes. Higher forward transfer is a desirable property as information consolidated in the previous tasks can be leveraged for learning new tasks. All methods including vanilla ER have positive forward transfer in R-MNIST. Both +MSE and $+L_{inf}$ have higher positive transfer while approximate methods such as +MI and +SimCLR have one of the lowest forward transfers. Strict but simple consistency regularizers improve positive forward transfer across buffer sizes thereby boosting the overall performance.

Table 5: Forward transfer on R-MNIST

Method	Buffer size = 200	Buffer size = 500
ER	$60.30_{\pm 1.32}$	$63.82_{\pm 2.50}$
$+L_1$	$59.96_{\pm 1.23}$	$64.67_{\pm 0.69}$
$+L_2$	$63.68_{\pm 0.93}$	$66.80_{\pm 0.60}$
$+L_{inf}$	$\underline{64.08}_{\pm 1.03}$	$\underline{67.30}_{\pm 0.14}$
+KL-Div	$58.29_{\pm 2.71}$	$62.84_{\pm 0.53}$
+MSE	$\mathbf{64.87}_{\pm 1.02}$	$\mathbf{67.42}_{\pm 0.43}$
+MI	$60.05_{\pm 2.15}$	$63.61_{\pm 0.32}$
+SimCLR	$59.44_{\pm 2.29}$	$63.16_{\pm 0.23}$
+BYOL	$63.04_{\pm 1.39}$	$65.86_{\pm 0.71}$
+DINO	$60.77_{\pm 1.98}$	$62.13_{\pm 1.79}$
+Barlow Twins	$62.46_{\pm 1.51}$	$65.63_{\pm 0.08}$

E   EXPECTED CALIBRATION ERROR

Miscalibration in neural networks can be defined as a difference in expectation between confidence and accuracy (Guo et al., 2017) i.e.

$$\mathbb{E}_{\hat{P}}[|\mathbb{P}(\hat{Y} = Y \mid \hat{P} = p) - p|] \tag{11}$$

where $\hat{Y}$ is a class prediction and $\hat{P}$ is its associated confidence, i.e. probability of correctness. ECE approximates Equation 11 by partitioning the predictions into M equal spaced bins and taking a weighted average of the bins' accuracy/confidence difference i.e.

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} |\text{acc}(B_m) - \text{conf}(B_m)| \tag{12}$$

where n is the number of samples. Lower ECE indicates better calibration of the model.

F   EFFECT OF α ON THE PERFORMANCE OF ER

We test the robustness of our analysis by tuning α in our experiments. One might also wonder that a simple fix to the task-recency bias in ER is to increase the magnitude of α. Table 6 presents Effect of α on vanilla ER's performance on S-CIFAR10 with buffer size 500. One can see that increasing the magnitude of the α disproportionately affects the overall performance (i.e., accuracy, calibration). Therefore, increasing the magnitude of α does not yield any discernible benefits including reduction in task recency bias.

Table 6: Effect of α on vanilla ER's performance on S-CIFAR10 with buffer size 500.

α	Accuracy Top-1 (%)	ECE score	Task-5 probability
1	$\mathbf{62.65}$	$\underline{31.8}$	$\mathbf{0.456}$
2	55.14	33.73	0.5291
5	$\underline{56.14}$	$\mathbf{31.74}$	$\underline{0.458}$
10	47.62	39.67	0.489

G   SENSITIVITY OF APPROXIMATE METHODS TO β

Our analysis shows that strict but simple regularizers perform better than approximate methods considered in this work. However, one might wonder whether the approximate methods perform better if their penalty (β) were to be increased. Table 7 shows the effect of β on the performance of approximate methods on TinyImageNet with buffer size 500. As can be seen, the performance of approximate methods is still quite inferior to the strict consistency constraints. The results presented in our work therefore are robust to the choice of β.

Table 7: Effect of β on the performance of approximate methods on TinyImageNet with buffer size 500.

β	+MI	+SimCLR	+DINO
0.05	9.93 ± 0.21	13.05 ± 0.08	13.04 ± 0.84
0.1	10.08 ± 0.28	13.93 ± 0.84	13.91 ± 0.55
0.5	10.26 ± 0.26	14.81 ± 0.79	14.23 ± 0.54
1	10.41 ± 0.40	14.68 ± 0.81	14.12 ± 0.46
1.5	10.33 ± 0.37	14.57 ± 0.50	13.55 ± 0.56
2	10.56 ± 0.28	14.00 ± 0.63	13.81 ± 0.81
5	10.32 ± 0.13	13.15 ± 0.74	11.44 ± 0.92
10	9.84 ± 0.33	11.70 ± 0.82	9.60 ± 0.49
50	7.78 ± 0.36	7.96 ± 0.41	8.62 ± 0.25