BrainWash: A Poisoning Attack to Forget in Continual Learning

Ali Abbasi
Vanderbilt University
ali.abbasi@vanderbilt.edu

Parsa Nooralinejad
University of California, Davis
pnoorali@ucdavis.edu

Hamed Pirsiavash
University of California, Davis
hpirsiav@ucdavis.edu

Soheil Kolouri
Vanderbilt University
soheil.kolouri@vanderbilt.edu

Abstract

Continual learning has gained substantial attention within the deep learning community, offering promising solutions to the challenging problem of sequential learning. Yet, a largely unexplored facet of this paradigm is its susceptibility to adversarial attacks, especially with the aim of inducing forgetting. In this paper, we introduce "BrainWash," a novel data poisoning method tailored to impose forgetting on a continual learner. By adding the BrainWash noise to a variety of baselines, we demonstrate how a trained continual learner can be induced to forget its previously learned tasks catastrophically, even when using these continual learning baselines. An important feature of our approach is that the attacker requires no access to previous tasks' data and is armed merely with the model's current parameters and the data belonging to the most recent task. Our extensive experiments highlight the efficacy of BrainWash, showcasing degradation in performance across various regularization and memory replay-based continual learning methods. Our code is available here: https://github.com/mint-vu/Brainwash

1. Introduction

In real-world scenarios, data distributions are inherently non-stationary, constantly evolving and shifting in unpredictable ways. Such variability poses a significant challenge to machine learning and computer vision, where model generalizability assumes stationary training and testing/deployment distributions. Continual Learning (CL) [13, 37, 61] has emerged as a prolific research domain focusing on efficient learning from an ongoing stream of data or tasks. CL primarily seeks to: 1) enhance backward knowledge transfer, which aims to maintain or improve performance on previously learned tasks, thereby mitigating catastrophic forgetting, and 2) bolster forward knowledge transfer, where learning a current task can boost performance on or reduce the learning time for future tasks. CL has significantly progressed in computer vision tasks, including incremental image recognition [33, 59]. With the increase in the adoption of CL algorithms, examining their vulnerabilities is imperative to inform the development of more robust CL methodologies.

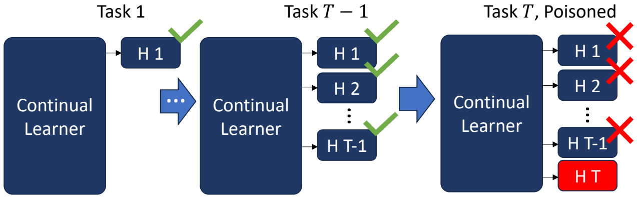

Figure 1. BrainWash is a poisoning attack targeting continual learning systems. It sabotages a task so that, upon learning it, the system's rate of forgetting previously learned tasks is increased.

Most research in CL has focused on overcoming catastrophic forgetting. Existing methods can be categorized into three groups: 1) memory replay, 2) regularization, and 3) parameter isolation methods. Nonetheless, there has been limited focus on the robustness of CL approaches against various types of adversarial attacks. Recent studies have begun to address this gap by proposing backdoor attacks [32, 57] and certain poisoning attacks [27, 39] within the CL context. These contributions are critical in profiling the vulnerabilities of CL methods, paving the way for developing more resilient CL algorithms. Additionally, these findings have implications for closely related and emerging fields such as machine unlearning [3, 9].

Recent works show that an adversary can insert misinformation into a task to distort a continual learner's performance. For instance, Umer et al. [57] show that backdoors can be placed into a task to hijack the performance of a CL method, and the backdoor remains effective even when new tasks are learned. Here, we pose a fundamental question: Is it possible to 'brainwash' a continual learner by poisoning its current task in such a way that performance on all pre-

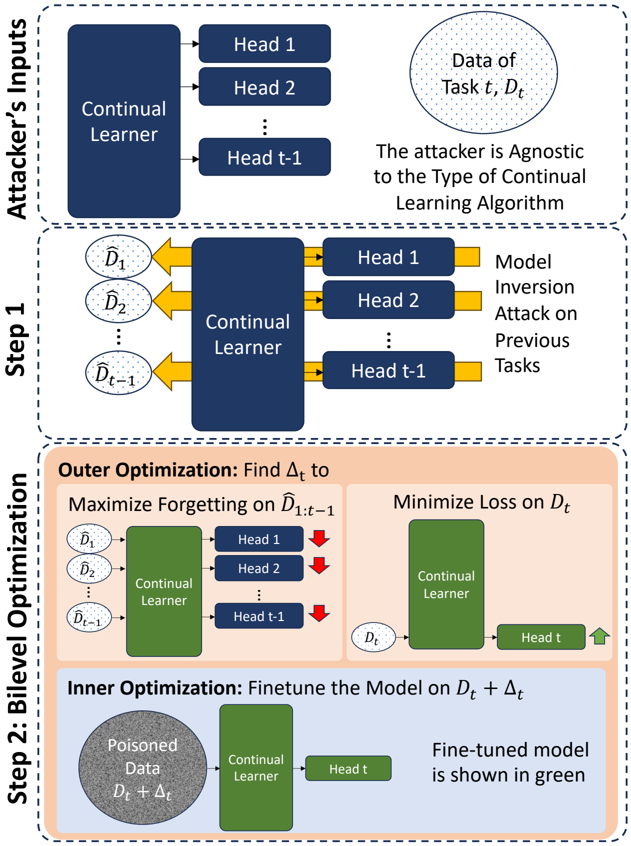
Attacker's Inputs
Continual Learner
Head 1
Head 2
⋮
Head t-1
Data of Task t, D_t
The attacker is Agnostic to the Type of Continual Learning Algorithm
Step 1
$\hat{D}_1$
$\hat{D}_2$
⋮
$\hat{D}_{t-1}$
Continual Learner
Head 1
Head 2
⋮
Head t-1
Model Inversion Attack on Previous Tasks
Step 2: Bilevel Optimization
Outer Optimization: Find Δ_t to
Maximize Forgetting on $\hat{D}_{1:t-1}$
$\hat{D}_1$
$\hat{D}_2$
$\hat{D}_{t-1}$
Continual Learner
Head 1
Head 2
Head t-1
Minimize Loss on D_t
Continual Learner
D_t
Head t
Inner Optimization: Finetune the Model on $D_t + \Delta_t$
Poisoned Data $D_t + \Delta_t$
Continual Learner
Head t
Fine-tuned model is shown in green

Figure 2. In our proposed threat model, the attacker gains access to the CL model and the data for the forthcoming task but remains unaware of the data from preceding tasks and the specific CL method employed by the victim (top panel). The attack methodology unfolds in two steps. Firstly, the attacker executes a model inversion attack on the CL model to reconstruct an approximation of the victim's data from earlier tasks (middle panel). Secondly, the attacker employs bi-level optimization to contaminate the data for the current task. This is done in such a way that performance on the reconstructed data from previous tasks is significantly degraded.

vious tasks is significantly degraded? More succinctly, can a task be designed to induce maximum forgetting of prior knowledge in a CL context? We affirmatively answer this question and demonstrate its validity across a wide range of regularization-based CL methods, assuming minimal and realistic conditions. This concept is depicted in Figure 1.

Recent advancements in foundational models have led to the creation of massive models with billions of parameters. These models require significant data resources, yet their training is limited by computational power, restricting repeated passes over the data. Additionally, data isn't sampled in an independently and identically distributed (i.i.d.) manner, necessitating continual learning to integrate new data without forgetting existing knowledge [25]. This forgetting vulnerability could be exploited by adversaries introducing manipulated training data to erase key information.

This paper examines a realistic threat model targeting regularization and memory-based CL methods. Under this model, the attacker gains access to the victim's current model and aims to manipulate the victim's next task. Crucially, the attacker remains unaware of the specific CL algorithm employed by the victim to learn tasks and lacks access to data from prior tasks. We propose a novel method denoted as "BrainWash" that allows for poisoning the current task data to maximize forgetting on prior tasks.

In short, BrainWash consists of two main steps. First, we perform a model inversion attack [24, 60] on the continual learner to approximate the data from the previous tasks. Second, to poison the current task, we construct a bi-level optimization problem such that: 1) the performance on inverted data of previous tasks is minimized, and 2) the performance on the clean data of the current task is maximized. Figure 2 demonstrates the threat model and the two steps.

Contributions. Our main contributions in this paper are:

1. Devising a novel poisoning attack algorithm for regularization-based continual learning methods, denoted as BrainWash.
2. Demonstrating the effectiveness of BrainWash on benchmark CL datasets and across diverse regularization-based CL algorithms.
3. Providing extensive ablation studies to deepen our understanding of BrainWash.

2. Related Work

Continual Learning is a subfield of ML focused on learning from nonstationary streams of data or tasks [13, 37]. Its objectives include improving backward knowledge transfer to maintain or enhance performance on previously learned tasks helping to prevent catastrophic forgetting. It also aims to strengthen forward knowledge transfer, where mastering a current task can improve performance or decrease learning time for future tasks. Catastrophic forgetting prevention is a central goal in this field. To tackle catastrophic forgetting, strategies in continual learning are typically grouped into three main categories: 1) memory-based methods, 2) regularization-based methods, and 3) architectural methods. Memory-based methods involve techniques such as memory rehearsal or replay, generative replay, and gradient projection [2, 18, 19, 40, 49, 50, 52, 56, 58, 59, 66]. These methods often rely on storing and revisiting previous learning experiences or artificially generating them to reinforce learning. Regularization-based methods apply penalties on changing parameters that are vital for tasks already learned [1, 5, 34, 35, 63, 69]. These approaches help in preserving the knowledge acquired from previous tasks while allowing new learning. Architectural methods focus on modifying the learning model itself. Strategies include expanding the model structure [51, 53], isolating parameters specific to certain tasks [43, 44], and using masking techniques [7, 47, 67] to manage the learning process for different tasks. In this paper, we focus on regularization-based methods, mainly due to their effective balance between plasticity and stability, allowing for the integration of new knowledge while preserving essential information from past learning experiences. We propose a data poisoning attack that maximizes forgetting for regularization-based continual learners.

Data Poisoning is a training phase attack on a ML model in which the attacker deliberately alters the victim's training data maliciously [4, 8, 20, 26, 31, 55, 70]. After the victim trains their model using this compromised data, the model would serve the attacker's detrimental objectives, such as significantly reducing the model's test accuracy on all or specific classes (i.e., targeted vs. non-targeted attacks).

Data poisoning is formally defined as a bi-level optimization problem [6, 8]. In the outer level optimization, the attacker optimizes the poisoning, which can be additive noise [26], patch-based noise [12], or a conditional generative model for noise [21], to enforce their malicious intention on the 'resulting network' parameters. This 'resulting network' itself is the solution to the inner optimization problem that minimizes the training objective as would be done by the victim. When the ML model is a deep neural network, this bi-level optimization problem is generally intractable, as it requires backpropagation through the entire SGD training procedure [46]. Hence, the existing literature often approximates this bi-level optimization using various strategies, including first-order approximation methods [29] and more sophisticated methods based on alternating optimization [21]. Similar to [22, 29], our poisoning attack also uses a first-order approximation method for solving the induced bi-level optimization. In contrast with [29], however, our bi-level optimization objective is maximizing forgetting in a continual learner.

Model Inversion [24, 48, 60] encompasses attack strategies designed to either reconstruct training data or deduce sensitive attributes from a trained model. These strategies are broadly divided into 'optimization-based' and 'training-based' methods. Our study primarily explores optimization-based methods, which are widely adopted in the literature [48, 68]. These methods primarily adjust inputs in the data space to maximally stimulate specific output neurons, such as target classes. However, a key challenge arises from the many-to-one mapping characteristic of deep neural networks, where a variety of inputs can lead to the same output. To address this, the literature introduces various forms of priors or regularization terms, making this optimization process more tractable. Such regularization terms range from simpler approaches like Total Variation and image norm [42, 45] to more advanced techniques involving feature statistics [68] and the use of generative models [64]. In this paper, we adopt a model inversion approach similar to Yin et al. [68] to approximate the data that the continual learner has been trained on from previous tasks.

3. Threat model

We consider a victim using a regularization or memory-based CL method to learn a series of tasks. For example, imagine a home robot that continuously learns from its environment to adapt to a new home [10]. The attacker's objective is to poison the training data of the latest task (like learning about a new room), causing the CL model to forget previously learned tasks upon acquiring new information. Furthermore, the attacker poisons the data in our setup by engineering norm-constrained additive noise. We examine two scenarios for an attack: 1) the 'reckless threat model' where the victim deploys the model without monitoring its performance, allowing the attacker to maximize forgetting of prior tasks without regard for current task performance; and 2) the 'cautious threat model' where the victim monitors the model's performance on a potentially poisoned task, necessitating the attacker to balance inducing forgetting while maintaining acceptable current task accuracy, making it a more challenging scenario. In both settings, we assume that the attacker does not have access to the continual learner's training data from previous tasks.

4. Method

In this work, we aim to design a poisoning attack for regularization and memory-based multi-head CL approaches that brainwashes the model, causing it to forget its previous tasks. We assume the attacker has full access to the model and data from the latest task the continual learner will encounter. However, the attacker does not have access to continual learner's data from the previous tasks.

We propose to utilize model inversion attacks [24, 68] to obtain an approximation for the continual learner's data from prior tasks. Using the victim's model, the inverted data from previous tasks, and the data for the current task, the attacker formalizes the poisoning problem through a bi-level optimization and then solves it via a first-order approximation method. In what follows, we briefly review our notations and then describe 1) the model inversion attack, 2) poisoning as a bi-level optimization problem, and 3) our proposed first-order approximation solver.

4.1. Notations

We denote the training data for task $t \in \{1, \cdots, T\}$ as $D_t = \{(x_t^i, y_t^i)\}_{i=1}^{N_t} \subset \mathcal{X} \times \mathcal{Y}_t$, where $x_t^i \in \mathcal{X}$ denotes the i'th sample from the t'th task (e.g., an input image) and $y_t^i \in \mathcal{Y}_t = \{1, \cdots, K_t\}$ denotes its corresponding label with K_t and N_t denoting the number of classes and examples for task t respectively. Let $f(\cdot; \theta)$ denote the CL's backbone that extracts deep representations from the input data, where θ indicates the backbone's parameters, and let $h_t(\cdot; \psi_t)$ denote the classification head for task t, with ψ_t representing its parameters.

Throughout the paper, we consider the supervised classification problem and denote the classification loss (e.g., cross-entropy) as $\mathcal{L}(\cdot)$. Moreover, we use $\ell_p(\cdot)$ to denote the p'th norm of a vector, and in particular, use ℓ_∞ norm in our experiments. Lastly, we indicate learned parameters calculated on clean data with a superscript asterisk and those calculated on the poisoned data with a tilde. For instance, ψ_t^* represents the optimal parameters for the t^{th} head, while $\theta^*_{1:T-1}$ denotes the optimal parameters of the backbone after learning tasks 1 to $T-1$, all calculated on the clean data. And $\tilde{\theta}$ and $\tilde{\psi}_T$ denote the backbone parameters and the parameters of the T'th head after poisoning.

4.2. Model Inversion

We propose executing a model inversion (MI) on the victim's CL model to approximate data from previous tasks. The outcome of this attack is proxy datasets for the previous tasks denoted as $\hat{D}_t = \{(\hat{x}_t^i, \hat{y}_t^i)\}_{i=1}^M$, where $\hat{x}_t^i$ is an inverted sample corresponding to label $\hat{y}_t^i$. To construct $\hat{D}_t$ for $t \in \{1, \cdots, T-1\}$, the attacker can infer the number of classes, K_t, by examining the logits in the t'th head, $h_t(\cdot; \psi_t^*)$. Following [68], we formulate the MI for a set of randomly sampled target one-hot labels $\{\hat{y}_t^i \in \mathcal{Y}_t\}_{i=1}^M$ as:

$$\{\hat{x}_t^i\}_{i=1}^M = \underset{\{x^i \in \mathcal{X}\}_{i=1}^M}{\arg\min} \sum_{i=1}^M \mathcal{L}(x^i, \hat{y}_t^i, \theta^*_{1:T-1}, \psi_t^*) + \sum_{i=1}^M \mathcal{R}_{\text{prior}}(x^i) + \alpha_f \mathcal{R}_{\text{feat}}(\{x^i\}_{i=1}^M, \theta^*_{1:T-1}), \quad (1)$$

where $\mathcal{R}_{\text{prior}}$ is an image regularization term that acts as a weak prior for natural images [45], and $\mathcal{R}_{feat}$ is a feature-statistics regularization [68]. For $\mathcal{R}_{\text{prior}}(x)$ we use:

$$\mathcal{R}_{\text{prior}}(x) = \alpha_{\text{TV}} \mathcal{R}_{\text{TV}}(x) + \alpha_{\ell_2} \mathcal{R}_{\ell_2}(x), \quad (2)$$

where $\mathcal{R}_{TV}(x)$ represents the total variation of image x, $\mathcal{R}_{\ell_2}(x)$ is the ℓ_2 norm of the image, and $\alpha_{\text{TV}}, \alpha_{\ell_2}, \alpha_f > 0$ denote the regularization coefficients. The feature-statistics regularization $\mathcal{R}_{feat}$ leverages the prevalence of batch normalization layers [30] in modern deep neural networks and the fact that they maintain a running mean and variance of training representations. Hence, $\mathcal{R}_{feat}$ requires the feature-statistics of the inverted samples $\{\hat{x}_t^i\}_{i=1}^M$ to align with those of the batch normalization layers, via:

$$\mathcal{R}_{\text{feat}}(\{x^i\}_{i=1}^M, \theta^*_{1:T-1}) = \sum_l \left\| \mu_l(\{x^i\}_{i=1}^M) - m_l \right\|_2^2 + \sum_l \left\| \sigma_l^2(\{x^i\}_{i=1}^M) - v_l \right\|_2^2. \quad (3)$$

Here, m_l and v_l are the running means and variances stored at the l^{th} batch normalization layer, and μ_l and σ_l^2 are the corresponding mean and variance of $\{x^i\}_{i=1}^M$ across examples at this layer. Note that our model inversion does not rely heavily on this regularizer, so it is still applicable in other architectures with no batch normalization layer.

4.3. Poisoning Formulation

We first formalize our poisoning attack for the 'reckless' attacker. The attacker constructs additive noise to the data from task T, such that when the victim trains their model on this task, the performance on tasks 1 to $T-1$ plummets. We assume that the attacker is oblivious to the victim's specific CL approach. Mathematically, we formalize the 'reckless' attacker problem as a bi-level optimization problem:

$$\begin{aligned} \{\delta_T^i\}_{i=1}^{N_T} &= \underset{\{\delta^i\}_{i=1}^{N_T}}{\arg\max} \sum_{t=1}^{T-1} \sum_{j=1}^{M} \mathcal{L}(\hat{x}_t^j, \hat{y}_t^j, \tilde{\theta}(\delta), \psi_t^*) \\ s.t. \quad \tilde{\theta}(\delta), \tilde{\psi}_T(\delta) &= \underset{\theta, \psi_T}{\arg\min} \sum_{i=1}^{N_T} \mathcal{L}(x_T^i + \delta^i, y_T^i, \theta, \psi_T) \\ \ell_\infty(\delta^i) &< \epsilon, \quad \forall i \end{aligned} \quad (4)$$

Here, δ_T^i is the optimal additive noise for sample i of task T, resulting in the poisoned data $x_T^i + \delta_T^i$, $\hat{x}_t^j$ represents the j'th inverted sample from task t, and ϵ is the threshold for the ℓ_∞ norm of the noise, which ensures inconspicuousness. Lastly, $\tilde{\theta}(\delta)$ is the updated parameters of the CL model when trained on the poisoned data from task T, which depends on the noise, δ. Importantly, in the inner optimization, the attacker is simply fine-tuning the model on the poisoned data of task T, starting from $\theta^*_{1:T-1}$ and ending at $\tilde{\theta}(\delta)$. Unlike the victim, the attacker does not use a CL algorithm for the inner optimization. We show that even though the inner optimization differs from the victim's exact optimization process, the constructed noise is highly effective against various CL approaches.

The bi-level optimization in (4) solely focuses on maximizing forgetting on tasks 1 through $T-1$, even at the cost of not learning task T, i.e., the 'reckless threat model.' In the 'cautious threat model,' on the other hand, the victim actively monitors the validation accuracy of the current task. This necessitates the attacker to carefully craft the training-time noise, such that forgetting of previous tasks is maximized, while the error on the clean data from task T is minimized. The bi-level optimization for the 'cautious' attacker is similar to that of the 'reckless' attacker, with an additional term in the outer optimization loop:

$$\{\delta_T^i\}_{i=1}^{N_T} = \underset{\{\delta^i\}_{i=1}^{N_T}}{\arg\max} \sum_{t=1}^{T-1} \sum_{j=1}^{M} \mathcal{L}(\hat{x}_t^j, \hat{y}_t^j, \tilde{\theta}(\delta), \psi_t^*) - \eta \sum_{i=1}^{N_T} \mathcal{L}(x_T^i, y_T^i, \tilde{\theta}(\delta), \tilde{\psi}_T(\delta)), \quad (5)$$

subject to the same constraints in (4). Note that the added loss term with weight $\eta > 0$ in the outer-level optimization ensures that the model trained on the poisoned data performs well on the clean data. Next, we discuss our strategy for solving these bi-level optimizations.

4.4. First-Order Approximation

Solving the bi-level optimization problems in (4) and (5) are intractable, as they require backpropagation through the entire Stochastic Gradient Decent (SGD) training procedure of the inner optimization. To address this, we follow [29] and leverage a first-order method that approximates the bi-level optimization problem using meta-learning [22].

In short, we simplify the inner objective (i.e., fine-tuning on poisoned data) by limiting the training to only k SGD steps for each evaluation of the outer objective. In other words, the outer backpropagation is only performed through the inner optimization's k unrolled SGD steps. Notably, such k-step methods are shown to decrease approximation error exponentially [54] and have significant generalization benefits [23]. In all our experiments, we set $k = 1$. Note that, for each iteration of the inner optimization, the head parameters for task T, ψ_T, are initialized randomly at each iteration while the backbone parameters, θ, are initialized from the learned backbone at the end of task $T-1$, $\theta^*_{1:T-1}$.

5. Experiments

This section provides experimental results evaluating our attack on various CL algorithms on three benchmark datasets.

5.1. Datasets and Model

We perform studies on three major CL benchmarks:

- **10-Split CIFAR-100:** CIFAR-100[36] consists of 100 classes of 32 × 32 images. We generated 10 ten-way classification tasks by splitting the classes. This dataset serves as our small-scale benchmark.
- **10-Split miniImagenet:** we also evaluated our noise on miniImageNet[62] which is a dataset of 60,000 84 × 84 images, divided in 100 categories. Similarly, we divide the miniImagenet to 10 classification tasks. This dataset serves as our medium-scale benchmark.
- **20-Split tinyImagenet:** As our large-scale benchmark, we used the 20-split tinyImageNet[38], which is a dataset of 200 classes with 100,000 images in total, each with the size of 64 × 64.

All experiments in this section use the ResNet-18 [28] architecture with Stochastic Gradient Descent (SGD) optimizer with learning rate 1e-2 and mini batchsize 16. All images were normalized in the range of $[0, 1]$, and the poisoned data was truncated to this range.

5.2. Regularization-Based CL Methods

In our experiments, we consider five renowned regularization-based methods starting from the classic Elastic Weight Consolidation (EWC) [34], Memory Aware Synapsis (MAS) [5], and Riemannian Walk (RWALK), to more recent methods like Active Forgetting of Negative Transfer (AFEC) [65] and Auxiliary Networks in Continual Learning (ANCL) [33]. Generally, the regularization-based CL methods assign importance values to network parameters and penalize the training for drastic changes in the important parameters. At a high level, this can be formulated as:

$$\theta^*_{1:T} = \arg\min_{\theta} \sum_{i=1}^{N_T} \mathcal{L}(x^i_T, y^i_T, \theta) + \lambda \mathcal{R}_{\text{CL}}(\theta, \theta^*_{1:T-1}),$$

where $\mathcal{R}_{\text{CL}}$ is a CL method-dependent regularizer that enforces stability of the continual learner, λ is the regularization coefficient that balances the stability vs. plasticity trade-off, and θ is initialized at $\theta^*_{1:T-1}$. It is widely accepted that the performance of regularization-based methods highly depends on the choice of λ.

5.3. Evaluation Metrics

We use the Backward Transfer (BWT) [41] and the (poisoned) model's accuracy on the last task for our evaluation metric. For the sake of completion, let $A_{t,i}$ denote the performance of the CL model on task i after learning task $t \geq i$. Then, BWT is defined as:

$$\text{BWT} = \frac{1}{T-1} \sum_{i=1}^{T-1} A_{T,i} - A_{i,i}. \tag{6}$$

Our poisoning attack aims to maximize forgetting on previous tasks or equivalently minimize the BWT.

5.4. Experiment Setup

We explore CL models that have been trained on $T-1$ tasks using the ideal regularization coefficient (λ) for their respective CL methods, aiming for a balance between plasticity and stability. For each victim model, we introduce poison to task T under two different ℓ_∞ norm bounds: $\epsilon = 0.1$ and $\epsilon = 0.3$. These bounds are applied in both 'reckless' and 'cautious' attacker scenarios, as described by Equations (4) and (5), leading to four distinct experimental setups. The victim models then learn the poisoned task T using their CL methods. Post learning, we evaluate the Backward Transfer (BWT) and the accuracy on the clean data of the last task for these victim models. For comparative analysis, we also include the BWT and accuracy of the victim models trained on the unpoisoned version of task T and on task T with added uniform noise. The outcomes of all these experimental configurations are detailed in Table 1. Our results indicate a significant BWT decrease when models are trained on BrainWash data. Additionally, it is observed that the 'cautious' attacker often achieves higher accuracy on task T compared to the 'reckless' attacker, albeit with a trade-off of a less potent attack. As anticipated, the poisoning effect increases with ϵ.

To aid in comprehending the results presented in Table 1, we have depicted the miniImageNet results as a spider chart in Figure 3. Key observations from this visualization include: 1) a discernible trade-off between enhanced for-

				$\epsilon = 0.1$			$\epsilon = 0.3$	
Dataset	**Method**	**Clean**	**Uniform**	**Cautious**	**Reckless**	**Uniform**	**Cautious**	**Reckless**
		BWT(Acc)	BWT(Acc)	BWT(Acc)	BWT(Acc)	BWT(Acc)	BWT(Acc)	BWT(Acc)
CIFAR-100	EWC [34]	-5.2 (68.3)	-5.1 (67.0)	-9.5 (58.8)	-12.6 (51.0)	-12.2 (57.5)	-24.7 (42.2)	-29.1 (25.5)
	AFEC [65]	-2.9 (65.6)	-3.6 (64.1)	-7.9 (57.2)	-8.8 (55.4)	-14.4 (52.4)	-24.2 (49.0)	-24.6 (36.9)
	ANCL [33]	-0.1 (81.5)	-0.2 (80.4)	-0.9 (61.7)	-4.4 (64.9)	-3.6 (68.5)	-6.2 (53.8)	-30.4 (44.3)
	MAS [5]	-1.8 (62.6)	-1.8 (67.7)	-5.8 (61.4)	-6.4 (53.7)	-9.6 (67.5)	-22.8 (50.7)	-17.9 (45.0)
	RWALK [11]	-6.0 (70.1)	-5.1 (68.5)	-16.3 (55.8)	-14.5 (62.8)	-25.5 (48.8)	-32.5 (47.0)	-21.6 (53.5)
mini ImageNet	EWC	-3.9 (56.8)	-1.5 (64.2)	-15.0 (42.5)	-23.1 (28.3)	-14.6 (58.0)	-27.9 (32.2)	-34.4 (22.5)
	AFEC	-1.3 (53.3)	-1.4 (52.9)	-14.7 (39.7)	-22.6 (30.9)	-15.1 (37.9)	-27.6 (22.8)	-38.2 (13.2)
	ANCL	-1.8 (74.5)	-2.2 (68.5)	-6.7 (34.8)	-5.3 (29.5)	-7.3 (59.0)	-14.6 (38.0)	-14.1 (21.4)
	MAS	-6.7 (54.6)	-6.9 (57.2)	-25.7 (40.3)	-30.3 (23.6)	-18.8 (48.8)	-39.8 (22.6)	-38.4 (16.4)
	RWALK	-5.6 (66.3)	-8.4 (53.6)	-13.5 (45.9)	-17.9 (38.0)	-22.6 (38.0)	-21.4 (37.4)	-27.4 (22.4)
tiny ImageNet	EWC	-0.4 (53.0)	0.7 (52.6)	-5.8 (39.2)	-7.1 (35.6)	-7.3 (44.6)	-28.4 (11.6)	-25.9 (16.2)
	AFEC	-1.3 (51.6)	-2.8 (51.0)	-10.3 (34.2)	-15.4 (30.4)	-13.5 (34.0)	-26.2 (15.0)	-27.8 (14.8)
	ANCL	-1.7 (73.4)	-1.5 (72.2)	-4.3 (46.0)	-6.7 (32.6)	-3.4 (51.8)	-10.7 (37.2)	-16.0 (22.8)
	MAS	-1.1 (59.4)	-1.8 (60.0)	-5.0 (54.2)	-12.9 (31.2)	-8.0 (46.8)	-25.5 (37.8)	-28.0 (22.6)
	RWALK	-14.1 (40.6)	-14.6 (39.6)	-25.9 (38.8)	-25.7 (44.2)	-29.8 (21.2)	-33.5 (26.6)	-33.3 (25.8)

Table 1. The backward transfer (BWT) and accuracy (Acc) for training different CL methods on the benchmark datasets when the last task is "clean" and when it is poisoned with BrainWash for $\epsilon \in \{0.1, 0.3\}$. Uniform stands for uniform noise in the range of $[-\epsilon, +\epsilon]$.

getting (i.e., decrease in BWT) and improved accuracy on the last task, 2) a consistent increase in last task accuracy for the 'cautious' attacker, though this comes with a reduction in forgetting efficiency, and 3) the notable superiority of ANCL and RWALK in withstanding our poisoning attack compared to other evaluated methods.

5.5. Replay-based CL Methods

We also explored the effectiveness of BrainWash on replay-based CL methods, specifically ER [16] and ER-ACE [14] with a single head network and a memory buffer size of 1000. We conducted experiments in a single-head, class incremental learning scenario using 10-split CIFAR100. Our threat model included two attacker profiles: one with read access to the victim's memory and one without memory access. The results are reported in Table 2. We note that scenarios where the attacker has write access to the memory represent a different yet significantly simpler threat model not considered here. We evaluate the reckless attacker with $\delta = 0.3$, and report BWT and accuracy on the final task for clean and poisoned settings under these threat models.

Table 2 shows that BrainWash remains effective against replay-based methods. Importantly, the replay-based methods allow for a diverse set of threat models with various levels of difficulty. For instance, whether the attacker has read or write access to the memory buffer or not, or the size of the memory buffer could significantly impact the results.

(BWT/Acc)	**Clean**	**BrainWash (Model Inv.)**	**BrainWash (Access to Mem.)**
ER [16]	-18.9 / 71.2	-23.1 / 53.8	**-27.7** / 49.5
ER-ACE [14]	-9.8 / 80.1	-16.6 / 55.1	**-17.0** / 55.9

Table 2. BrainWash against single-head replay-based CL methods.

5.6. Data Poisoning Baselines

While there are no directly comparable baselines, we evaluated BrainWash against three additional data poisoning baselines. This assessment used a task incremental setting on 10-split CIFAR100 with EWC and the backward transfer and the accuracy of the last task are detailed in Table 3. We acknowledge that the evaluated baselines are not specifically designed for attacking continual learners; however, they provide a better insight into the dynamics of different attacks in a CL setting. For instance, the Unlearnable Examples [15], drastically reduce the attacked task's accuracy while not being as effective as ours in inducing forgetting.

Clean	**Unlearnable Examples [15]**	**DeepConfuse [21]**	**MetaPoison [29]**	**BrainWash**
-5.2 / 68.3	-21.4 / 3.7	-19.42 / 10.0	-22.94 / 35.3	**-29.1** / 25.5

Table 3. BrainWash against data poisoning baselines

Next, we conduct various ablation studies to gain deeper insights into BrainWash.

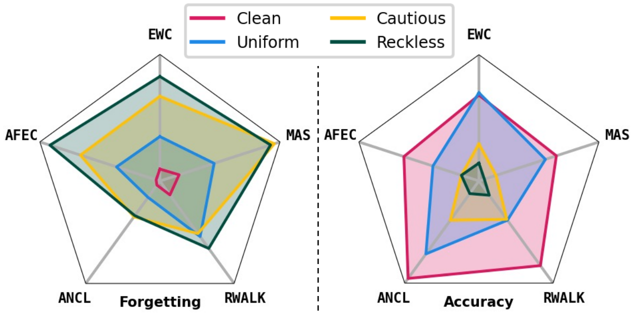

Figure 3. Forgetting (i.e., negative backward transfer) and accuracy of task T for different attacking strategies and on different CL approaches trained on miniImageNet with $\epsilon = 0.3$. As can be seen, forgetting is minimal when the continual learner is trained on the clean data. Adding uniform noise increases forgetting, while 'cautious' and 'reckless' attackers increase forgetting by a large margin. Also, the trade-off between the attack's success and the accuracy of the last task is apparent.

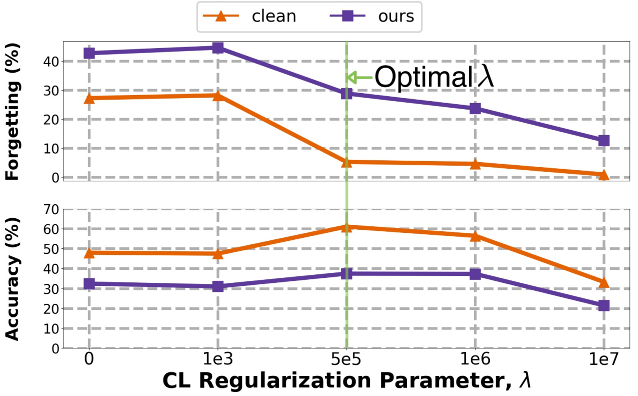

Figure 4. Performance of BrainWash against victims using different regularization coefficients.

6. Abaltion Studies

We performed various ablation studies to evaluate the sensitivity of BrainWash to different design choices. Please note that throughout the ablation experiments, we use the term **"forgetting"**, corresponding to the negative of BWT. Moreover, all our ablation studies were performed on EWC.

6.1. Sensitivity to λ

As previously mentioned in Section 5.4, our results reflect the victim model's natural behavior, particularly in choosing the optimal λ for their CL algorithm. It's important to note that the degree of induced forgetting and the overall performance of the victim is significantly influenced by the value of λ. Also, there is a notion that increasing the network's stability (i.e., opting for higher values of λ) might act as an effective defense against BrainWash, under the assumption that the lesser the amount of "bad data" learned by the victim, the lower the level of forgetting. This section delves into the relationship between BrainWash's effectiveness and the choice of λ.

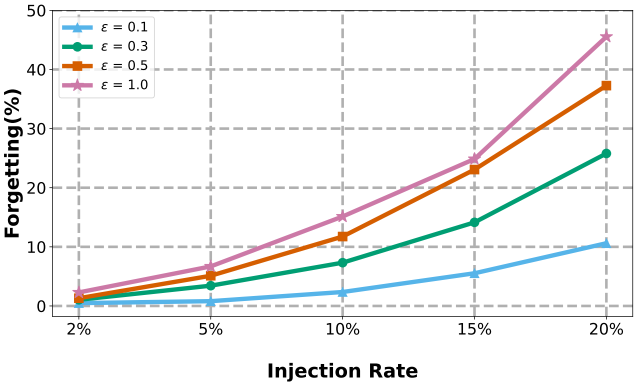

Figure 5. Attack's effectiveness with respect to different injection rates. Forgetting is defined as the negative value of backward transfer. Note that the x-axis reports the injection rate for the whole dataset, so 20% means 100% of the final task is poisoned.

To explore this, we conducted an experiment on 10-split CIFAR-100. Here, the assumption is that the victim model has been trained on the first nine tasks using various fixed λ values, which remain constant throughout the CL process. The BrainWash is then applied to the data of the 10^{th} task. It is crucial to point out that BrainWash remains oblivious to the λ value used by the victim. Fig. 4 shows the sensitivity of BrainWash to the victim's choice of λ, which identifies different degrees of network plasticity. The top plot shows the amount of forgetting while the bottom plot demonstrates the average accuracy of the victim on the past 9 tasks as a function of λ. Figure 6.1 indicates that BrainWash consistently leads to increased forgetting and reduced average performance across different λ values, even when the victim opts for the optimal λ for their context, such as $\lambda = 5e5$ in this experiment. Although increasing the network's intransigence (using higher λ values) marginally diminishes the attack's effectiveness, this approach also significantly compromises the victim's overall performance. Therefore, BrainWash proves to be resilient to varying λ choices, and utilizing high λ values is not an efficient defense strategy.

6.2. Dependency on Injection Rate and Noise Norm

To delve deeper into the dynamics of BrainWash, we assessed its impact by varying the noise injection rate (the percentage of data that is poisoned) and the noise magnitude. In an experiment using CIFAR-100 divided into five tasks, we poisoned the last task with different injection rates and noise magnitudes. It's important to note that poisoning the entire task equates to a 20% injection rate (since one out of the five tasks is poisoned), and poisoning 10% of the last task corresponds to a 2% injection ratio.

Figure 5 demonstrates how both the injection rate and noise amplitude influence the extent of for getting. Our observations indicate a direct correlation between forgetting,

	Clean	Inv Data No Reg	Inv Data with Reg	Real Data
Forgetting	5.2	23.6	28.8	28.16

Table 4. Difference between the induced forgetting while using different alternatives for the past data

noise norm, and injection rate. This figure also reveals a constant trade-off between the subtlety of the noise and the amount of forgetting induced: increasing either the norm or the injection rate leads to more pronounced forgetting. However, in scenarios where stealthiness is crucial, such increases in noise or rate can potentially expose BrainWash. Despite this, the results show that effective forgetting can still be achieved even with a minimal injection rate.

6.3. The Effect of Model Inversion

As previously mentioned, we considered a scenario where the attacker might not have access to data from previous tasks. To address this, we suggested using model inversion, employing inverted samples as proxies. In this context, we examined the significance of model inversion for the effectiveness of BrainWash. Our study on CIFAR100, segmented into 10 tasks, evaluated the impact of BrainWash under three distinct conditions: 1) access to actual data from preceding tasks, 2) application of a basic model inversion technique without any regularization, and 3) utilization of regularized model inversion, as detailed in Section 4.2.

The findings are presented in Table 4, which illustrates the percentage of forgetting associated with each of these strategies. The term 'Clean' refers to the inherent forgetting experienced by the continual learner trained on the clean final task. Notably, BrainWash, when implemented with regularized model inversion, achieves results comparable to the scenario with direct access to real data. Furthermore, the robustness of BrainWash to the choice of model inversion method is evident, as the basic, non-regularized inversion demonstrates only marginal underperformance compared to its more advanced counterpart.

6.4. Different Task Lengths

Here we focus on assessing the efficacy of BrainWash at various stages of a continual learner's training. We divided CIFAR-100 into 20 five-way classification tasks and measured the extent of forgetting immediately after introducing noise in the 10^{th}, 15^{th}, and 20^{th} tasks. The results, depicted in Table 5, confirm that BrainWash effectively induces forgetting at different training stages of the continual learner.

An intriguing finding is the variation in the injection rate of BrainWash across these stages. For instance, at task 10, BrainWash has an injection rate of 10% (being applied to one out of ten tasks). However, this rate decreases when poisoning is applied to 15 or 20 tasks, leading to a corresponding reduction in the attack's strength. The result illustrates how the impact of BrainWash is influenced by its relative scale in the context of the overall training process.

Method	10 Tasks	15 Tasks	20 Tasks
Clean	1.4	2.17	2.16
Ours	17.47	14.61	14.04

Table 5. Effect of different number of tasks on forgetting

6.5. Efficacy on Different Architectures

To show the independence of BrainWash to the architecture, we repeat the 10-split CIFAR-100 experiment with the RegNetX [17] on ANCL, and report the results in Table 6.

(BWT/Acc)	Clean	BrainWash
RegNetX-1.6GF [17]	-6.96 / 66.8	-17.6 / 43.3
Resnet-18	-0.1 / 81.5	-30.4 / 44.3

Table 6. BrainWash performed on ANCL on RegNetX.

7. Conclusion

This study presents BrainWash, an innovative poisoning attack for regularization-based continual learning (CL) models. Its primary objective is to maximize the forgetting of a continual learner on previously learned tasks. We introduced two threat models: the 'reckless' and the 'cautious' attacker. Both threat models assume that the attacker can only access the trained model and the clean data from the last task. Critically, the attacker is unaware of the specific CL method employed by the victim or any related hyperparameters. Our core strategy involves using model inversion to approximate data from earlier tasks. The attacker then employs a bilevel optimization problem to poison the current task's data. When the victim trains their model on this manipulated data, their performance on prior tasks is adversely affected. The 'reckless' attacker disregards the victim's performance on the last task's clean data, while the 'cautious' attacker seeks to preserve high accuracy on it.

In our extensive experiments, we employed five well-known continual learning methods and three benchmark datasets to demonstrate the efficacy of BrainWash as a potent poisoning attack. Moreover, we provided a series of detailed ablation studies to offer a thorough understanding of BrainWash's mechanics and impacts.

Acknowledgement

The authors thank Oracle Cloud Infrastructure (OCI) for generously providing computational resources for this study. This work was partially supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR00112290115 and HR00112190135 and NSF awards 1845216 and 2339898.

References

[1] *Lifelong Learning with Sketched Structural Regularization*, 2021. PMLR. 2

[2] Ali Abbasi, Parsa Nooralinejad, Vladimir Braverman, Hamed Pirsiavash, and Soheil Kolouri. Sparsity and heterogeneous dropout for continual learning in the null space of neural activations. In *Conference on Lifelong Learning Agents*, pages 617–628. PMLR, 2022. 2

[3] Ali Abbasi, Chayne Thrash, Elaheh Akbari, Daniel Zhang, and Soheil Kolouri. Covarnav: Machine unlearning via model inversion and covariance navigation. *arXiv preprint arXiv:2311.12999*, 2023. 1

[4] Scott Alfeld, Xiaojin Zhu, and Paul Barford. Data poisoning attacks against autoregressive models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016. 3

[5] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *ECCV*, 2018. 2, 5, 6

[6] Jonathan F Bard and James E Falk. An explicit solution to the multi-level programming problem. *Computers & Operations Research*, 9(1):77–100, 1982. 3

[7] Eseoghene Ben-Iwhiwhu, Saptarshi Nath, Praveen Kumar Pilly, Soheil Kolouri, and Andrea Soltoggio. Lifelong reinforcement learning with modulating masks. *Transactions on Machine Learning Research*, 2023. 2

[8] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1467–1474, 2012. 3

[9] Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 141–159. IEEE, 2021. 1

[10] Matthew Chang, Theophile Gervet, Mukul Khanna, Sriram Yenamandra, Dhruv Shah, So Yeon Min, Kavit Shah, Chris Paxton, Saurabh Gupta, Dhruv Batra, et al. Goat: Go to any thing. *arXiv preprint arXiv:2311.06430*, 2023. 3

[11] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European conference on computer vision (ECCV)*, pages 532–547, 2018. 6

[12] Jinyin Chen, Longyuan Zhang, Haibin Zheng, Xueke Wang, and Zhaoyan Ming. Deeppoison: Feature transfer based stealthy poisoning attack for dnns. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 68(7):2618–2622, 2021. 3

[13] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385, 2021. 1, 2

[14] Caccia et al. New insights on reducing abrupt representation change in online continual learning. In *ICLR*, 2021. 6

[15] Huang et al. Unlearnable examples: Making personal data unexploitable. In *ICLR*, 2020. 6

[16] Rolnick et al. Experience replay for continual learning. *NeurIPS*, 2019. 6

[17] Radosavovic et al. Designing network design spaces. In *CVPR*, 2020. 8

[18] Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *International Conference on Artificial Intelligence and Statistics*, pages 3762–3773. PMLR, 2020. 2

[19] Sebastian Farquhar and Yarin Gal. Towards robust evaluations of continual learning. *arXiv preprint arXiv:1805.09733*, 2018. 2

[20] Ji Feng, Qi-Zhi Cai, and Zhi-Hua Zhou. Learning to confuse: Generating training time adversarial data with auto-encoder. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019. 3

[21] Ji Feng, Qi-Zhi Cai, and Zhi-Hua Zhou. Learning to confuse: generating training time adversarial data with autoencoder. *Advances in Neural Information Processing Systems*, 32, 2019. 3, 6

[22] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017. 3, 5

[23] Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *International conference on machine learning*, pages 1568–1577. PMLR, 2018. 5

[24] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333, 2015. 2, 3

[25] Saurabh Garg, Mehrdad Farajtabar, Hadi Pouransari, Raviteja Vemulapalli, Sachin Mehta, Oncel Tuzel, Vaishaal Shankar, and Fartash Faghri. Tic-clip: Continual training of clip models. *arXiv preprint arXiv:2310.16226*, 2023. 2

[26] Jonas Geiping, Liam H Fowl, W. Ronny Huang, Wojciech Czaja, Gavin Taylor, Michael Moeller, and Tom Goldstein. Witches' brew: Industrial scale data poisoning via gradient matching. In *International Conference on Learning Representations*, 2021. 3

[27] Gyojin Han, Jaehyun Choi, Hyeong Gwon Hong, and Junmo Kim. Data poisoning attack aiming the vulnerability of continual learning. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 1905–1909. IEEE, 2023. 1

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778. IEEE, 2016. 5

[29] W Ronny Huang, Jonas Geiping, Liam Fowl, Gavin Taylor, and Tom Goldstein. Metapoison: Practical general-purpose clean-label data poisoning. *Advances in Neural Information Processing Systems*, 33:12080–12091, 2020. 3, 5, 6

[30] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015. 4

[31] Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *2018 IEEE symposium on security and privacy (SP)*, pages 19–35. IEEE, 2018. 3

[32] Siteng Kang, Zhan Shi, and Xinhua Zhang. Poisoning generative replay in continual learning to promote forgetting. In *Proceedings of the 40th International Conference on Machine Learning*, pages 15769–15785. PMLR, 2023. 1

[33] Sanghwan Kim, Lorenzo Noci, Antonio Orvieto, and Thomas Hofmann. Achieving a better stability-plasticity trade-off via auxiliary networks in continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11930–11939, 2023. 1, 5, 6

[34] 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. 2, 5, 6

[35] Soheil Kolouri, Nicholas A Ketz, Andrea Soltoggio, and Praveen K Pilly. Sliced cramer synaptic consolidation for preserving deeply learned representations. In *International Conference on Learning Representations*, 2020. 2

[36] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-100 (canadian institute for advanced research). 5

[37] Dhireesha Kudithipudi, Mario Aguilar-Simon, Jonathan Babb, Maxim Bazhenov, Douglas Blackiston, Josh Bongard, Andrew P Brna, Suraj Chakravarthi Raja, Nick Cheney, Jeff Clune, et al. Biological underpinnings for lifelong learning machines. *Nature Machine Intelligence*, 4(3):196–210, 2022. 1, 2

[38] Ya Le and Xuan S. Yang. Tiny imagenet visual recognition challenge. 2015. 5

[39] Huayu Li and Gregory Ditzler. Targeted data poisoning attacks against continual learning neural networks. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2022. 1

[40] Sen Lin, Li Yang, Deliang Fan, and Junshan Zhang. Trgp: Trust region gradient projection for continual learning. In *International Conference on Learning Representations*, 2022. 2

[41] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017. 5

[42] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5188–5196, 2015. 3

[43] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2018. 2

[44] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 67–82, 2018. 2

[45] Alexander Mordvintsev, Christopher Olah, and Mike Tyka. Deepdream-a code example for visualizing neural networks. *Google Research*, 2(5), 2015. 3, 4

[46] Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C Lupu, and Fabio Roli. Towards poisoning of deep learning algorithms with back-gradient optimization. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pages 27–38, 2017. 3

[47] Saptarshi Nath, Christos Peridis, Eseoghene Ben-Iwhiwhu, Xinran Liu, Shirin Dora, Cong Liu, Soheil Kolouri, and Andrea Soltoggio. Sharing lifelong reinforcement learning knowledge via modulating masks. In *Second Conference on Lifelong Learning Agents (CoLLAs) 2023*, 2023. 2

[48] Ngoc-Bao Nguyen, Keshigeyan Chandrasegaran, Milad Abdollahzadeh, and Ngai-Man Cheung. Re-thinking model inversion attacks against deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16384–16393, 2023. 3

[49] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[50] Mohammad Rostami, Soheil Kolouri, Praveen Pilly, and James McClelland. Generative continual concept learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5545–5552, 2020. 2

[51] 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. 2

[52] Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. In *International Conference on Learning Representations*, 2020. 2

[53] 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*, pages 4528–4537. PMLR, 2018. 2

[54] Amirreza Shaban, Ching-An Cheng, Nathan Hatch, and Byron Boots. Truncated back-propagation for bilevel optimization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1723–1732. PMLR, 2019. 5

[55] Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. *Advances in neural information processing systems*, 31, 2018. 3

[56] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 2994–3003, 2017. 2

[57] Muhammad Umer, Glenn Dawson, and Robi Polikar. Targeted forgetting and false memory formation in continual learners through adversarial backdoor attacks. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020. 1

[58] Gido M Van de Ven and Andreas S Tolias. Generative replay with feedback connections as a general strategy for continual learning. *arXiv preprint arXiv:1809.10635*, 2018. 2

[59] 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):4069, 2020. 1, 2

[60] Michael Veale, Reuben Binns, and Lilian Edwards. Algorithms that remember: model inversion attacks and data protection law. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 376 (2133):20180083, 2018. 2, 3

[61] Eli Verwimp, Shai Ben-David, Matthias Bethge, Andrea Cossu, Alexander Gepperth, Tyler L Hayes, Eyke Hüllermeier, Christopher Kanan, Dhireesha Kudithipudi, Christoph H Lampert, et al. Continual learning: Applications and the road forward. *arXiv preprint arXiv:2311.11908*, 2023. 1

[62] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2016. 5

[63] Johannes von Oswald, Christian Henning, João Sacramento, and Benjamin F Grewe. Continual learning with hypernetworks. *arXiv preprint arXiv:1906.00695*, 2019. 2

[64] Kuan-Chieh Wang, Yan Fu, Ke Li, Ashish Khisti, Richard Zemel, and Alireza Makhzani. Variational model inversion attacks. *Advances in Neural Information Processing Systems*, 34:9706–9719, 2021. 3

[65] Liyuan Wang, Mingtian Zhang, Zhongfan Jia, Qian Li, Chenglong Bao, Kaisheng Ma, Jun Zhu, and Yi Zhong. Afec: Active forgetting of negative transfer in continual learning. *Advances in Neural Information Processing Systems*, 34: 22379–22391, 2021. 5, 6

[66] Shipeng Wang, Xiaorong Li, Jian Sun, and Zongben Xu. Training networks in null space of feature covariance for continual learning. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 184–193, 2021. 2

[67] Mitchell Wortsman, Vivek Ramanujan, Rosanne Liu, Aniruddha Kembhavi, Mohammad Rastegari, Jason Yosinski, and Ali Farhadi. Supermasks in superposition. *Advances in Neural Information Processing Systems*, 33:15173–15184, 2020. 2

[68] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8715–8724, 2020. 3, 4

[69] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, pages 3987–3995. PMLR, 2017. 2

[70] Chen Zhu, W Ronny Huang, Hengduo Li, Gavin Taylor, Christoph Studer, and Tom Goldstein. Transferable clean-label poisoning attacks on deep neural nets. In *International Conference on Machine Learning*, pages 7614–7623. PMLR, 2019. 3