arXiv:2011.12216v2 [cs.LG] 18 Feb 2021

Energy-Based Models for Continual Learning

Shuang Li [1] **Yilun Du** [1] **Gido M.van de Ven** [2] **Igor Mordatch** [3]

Abstract

We motivate Energy-Based Models (EBMs) as a promising model class for continual learning problems. Instead of tackling continual learning via the use of external memory, growing models, or regularization, EBMs have a natural way to support a dynamically-growing number of tasks or classes that causes less interference with previously learned information. Our proposed version of EBMs for continual learning is simple, efficient and outperforms baseline methods by a large margin on several benchmarks. Moreover, our proposed contrastive divergence based training objective can be applied to other continual learning methods, resulting in substantial boosts in their performance. We also show that EBMs are adaptable to a more general continual learning setting where the data distribution changes without the notion of explicitly delineated tasks. These observations point towards EBMs as a class of models naturally inclined towards the continual learning regime.

1. Introduction

Humans are able to rapidly learn new skills and continuously integrate them with prior knowledge. The field of Continual Learning (CL) seeks to build artificial agents with the same capabilities (Parisi et al., 2019). In recent years, continual learning has seen increased attention, particularly in the context of classification problems. Continual learning requires models to remember prior skills as well as incrementally learn new skills, without necessarily having a notion of an explicit task identity. Standard neural networks (He et al., 2016; Simonyan & Zisserman, 2014; Szegedy et al., 2015) experience the catastrophic forgetting problem and perform poorly in this setting. Different approaches have been proposed to mitigate catastrophic forgetting, but many rely on the usage of external memory (Lopez-Paz & Ranzato, 2017; Li & Hoiem, 2017), additional models (Shin et al., 2017), or auxiliary objectives and regularization (Kirkpatrick et al., 2017; Schwarz et al., 2018; Maltoni & Lomonaco, 2019), which can constrain the wide applicability of these methods.

In this work, we propose a new approach towards continual learning on classification tasks. Most existing CL approaches tackle these tasks by utilizing normalized probability distribution (i.e., softmax output layer) and trained with a cross-entropy objective. In this paper, we argue that by viewing classification from the lens of training an unnormalized probability distribution, we can significantly improve continual learning performance in classification problems. In particular, we interpret classification as learning an Energy-Based Model (EBM) across classes. Training becomes a wake-sleep process, where the energy of an input data at its ground truth label is decreased while the energy of the input at (an)other selected class(es) is increased. An important advantage is that this framework offers freedom to choose what classes to update in the continual learning process. By contrast, the cross entropy objective reduces the likelihood of *all* negative classes when given a new input, creating updates that lead to catastrophic forgetting.

The energy function, which maps an input-label pair to a scalar energy, also provides a way for the model to select and filter portions of the input that are relevant towards the classification on hand. We show that this enables EBMs training updates for new data to interfere less with previous data. In particular, our formulation of the energy function allows us to compute the energy of an input by learning a conditional gain based on the class label, which serves as an attention filter to select the most relevant information. In the event of a new class, a new conditional gain can be learned.

These unique properties benefit EBMs in addressing two important open challenges in continual learning. 1) First, we show that EBMs are promising for class-incremental learning, which is one of the most challenging settings for continual learning (van de Ven & Tolias, 2019). Generally, successful existing methods for class-incremental learning either store data or use generative replay, which has disadvantages in terms of memory and/or computational efficiency. We show that EBMs perform well in class-incremental learning without using replay and without relying on stored data. 2) The second open challenge that EBMs can address is continual learning without task boundaries. Typically, a continual learning problem is set up as a sequence of distinct

tasks with clear boundaries that are known to the model (the *boundary-aware* setting). Most existing continual learning methods rely on these known boundaries for performing certain consolidation steps (e.g., calculating parameter importance, updating a stored copy of the model). However, assuming such clear boundaries is not always realistic, and often a more natural scenario is the *boundary-agnostic* setting (Zeno et al., 2018; Rajasegaran et al., 2020), in which data distributions gradually change without a clear notion of task boundaries. While many common CL methods cannot be used without clear task boundaries, we show that EBMs can be naturally applied to this more challenging setting.

There are three main contributions of our work. First, we introduce energy-based models for classification continual learning problems. We show that EBMs can naturally deal with challenging problems in CL, including the boundary-agnostic setting and class-incremental learning without using replay. Secondly, we propose an energy-based training objective that is *simple* and broadly applicable to different types of models, with significant boosts on their performance. This contrastive divergence based training objective can naturally handle the dynamically growing number of classes and significantly reduces catastrophic forgetting. Lastly, we show that the proposed EBMs perform strongly on four standard CL benchmarks: split MNIST, permuted MNIST, CIFAR-10, and CIFAR-100. These observations point towards EBMs as a class of models naturally inclined towards the CL regime and as an important new baseline upon which to build further developments. The code are made public to facilitate further research [1].

2. Related work

2.1. Continual learning settings

Boundary-aware versus boundary-agnostic. In most existing CL studies, models are trained in a "boundary-aware" setting, in which a sequence of distinct tasks with clear task boundaries is given (e.g., Kirkpatrick et al., 2017; Zenke et al., 2017; Shin et al., 2017). There are no overlaps between any two tasks; for example task 1 has data class labels "1,2" and task 2 has data with class labels "3,4". Models are first trained on the first task and then move to the second one. Moreover, models are typically told when there is a transition from one task to the next. However, it could be argued that it is more realistic for tasks to change gradually and for models to not be explicitly informed about the task boundaries. Such a boundary-agnostic setting has been explored in (Zeno et al., 2018; Rajasegaran et al., 2020; Aljundi et al., 2019). In this setting, models learn in a streaming fashion and the data distributions gradually change over

---

[1] MIT CSAIL [2] Baylor College of Medicine [3] Google Brain. Correspondence to: Shuang Li <lishuang@mit.edu>.

[1] Code and documentation are available at `https://energy-based-model.github.io/Energy-Based-Models-for-Continual-Learning`

time. In (Zeno et al., 2018), the percentage of “1s” gradually decrease while the percentage of “2s” increases during training. Importantly, most existing CL approaches are not applicable to this setting as they require the task boundaries to decide when to perform certain consolidation steps. In this paper, we will show that our proposed approach can naturally handle both the boundary-aware and boundary-agnostic settings.

Task-incremental versus class-incremental learning. Another important distinction in CL is between task-incremental learning (*Task-IL*) and class-incremental learning (*Class-IL*) (van de Ven & Tolias, 2019; Prabhu et al., 2020). In *Task-IL*, also referred to as the multi-head setting (Farquhar & Gal, 2018), models predict the label of an input data by choosing only from the labels in the task where the data come from. In *Class-IL*, also referred to as the single-head setting, models chose between the classes from all tasks so far when asked to predict the label of an input data. *Class-IL* is more challenging than *Task-IL* as it requires models to select the correct labels from the mixture of new and old classes. Generally, to perform well on *Class-IL*, existing methods need to store data, use replay, or pretrain models from another large dataset (Rebuffi et al., 2017; Rajasegaran et al., 2019; Belouadah et al., 2020; Maltoni & Lomonaco, 2019; Hayes & Kanan, 2020).

2.2. Continual learning approaches

Numerous methods have been proposed for continual learning. Here we broadly partition them into three categories: task-specific, regularization, and replay-based approaches.

Task-specific methods. One way to reduce interference between tasks is by using different parts of a neural network for different tasks. For a fixed-size network, such specialization could be achieved by learning a separate mask for each task (Fernando et al., 2017; Serra et al., 2018), by *a priori* defining a different, random mask for every task (Masse et al., 2018), or by using a different set of parameters for each task (Zeng et al., 2019; Hu et al., 2019). Other methods let models grow or recruit new resources when learning new tasks, such as progressive neural networks (Rusu et al., 2016) and dynamically expandable networks (Yoon et al., 2017). Although these methods are generally successful in reducing catastrophic forgetting, a key disadvantage is that they require knowledge of task identities during training and testing. They are therefore not suitable for *Class-IL*.

Regularization-based methods. Regularization is used in CL to encourage the stability of those aspects of the network that are important for previous tasks. A popular strategy is to add a regularization loss to penalise changes of important parameters. EWC (Kirkpatrick et al., 2017)) and online EWC (Schwarz et al., 2018) evaluate the importance of parameters using the diagonal elements in the fisher information matrices, while SI (Zenke et al., 2017) estimates the

---

parameters’ importance online. LwF (Li & Hoiem, 2017) regularizes the network at the functional level. Although regularization-based methods can be computationally efficient, a disadvantage is that typically they gradually reduce the model’s capacity for learning new tasks. Moreover, while in theory these methods can be used for *Class-IL*, in practice they have been shown to fail on such problems (Farquhar & Gal, 2018; van de Ven & Tolias, 2019).

Replay methods. To preserve knowledge, replay methods periodically rehearse previous information during training (Robins, 1995). Exact or experience replay based methods store data from previous tasks and revisit them when training on new tasks. Although straightforward, such methods face critical non-trivial questions, such as how to select the data to be stored and how to use them (Lopez-Paz & Ranzato, 2017; Hou et al., 2019; Wu et al., 2019; Mundt et al., 2020). An alternative is to generate the replayed data. In (Shin et al., 2017), a generative model is sequentially trained to generate samples of previous tasks. While both types of replay can mitigate forgetting, an important disadvantage is that they are computationally relatively expensive. Additionally, storing data might not always be possible while incrementally training a generative model is a challenging problem in itself (Lesort et al., 2019; van de Ven et al., 2020).

In contrast, we propose EBMs for continual learning that reduce catastrophic forgetting without requiring knowledge of task-identity, without gradually restricting the model’s learning capabilities, and without using stored data.

3. CL with softmax-based classifiers

The most common way to do classification with deep neural networks is to use a softmax output layer in combination with a cross-entropy loss. In continual learning, virtually all existing methods for classification are based on the softmax-based classifier (SBC) (Li & Hoiem, 2017; van de Ven & Tolias, 2019; Pellegrini et al., 2019; Zenke et al., 2017). In this section, we start by describing classification with a softmax-based classifier. We then describe the challenges of softmax-based classification in CL.

3.1. Softmax-based classification

Given an input $\mathbf{x} \in \mathbb{R}^D$ and a discrete set $\mathcal{Y} = \{1, \ldots, N\}$ of N possible class labels, a traditional softmax-based classifier defines the conditional probabilities of those labels as:

$$p_{\boldsymbol{\theta}}(y|\mathbf{x}) = \frac{\exp([f_{\boldsymbol{\theta}}(\mathbf{x})]_y)}{\sum_{i \in \mathcal{Y}} \exp([f_{\boldsymbol{\theta}}(\mathbf{x})]_i)}, \quad \text{for all } y \in \mathcal{Y}, \tag{1}$$

where $f_{\boldsymbol{\theta}}(\mathbf{x}) : \mathbb{R}^D \rightarrow \mathbb{R}^N$ is a feed-forward neural network, parameterized by $\boldsymbol{\theta}$, that maps an input $\mathbf{x}$ to a N-dimensional vector of logits. $[\cdot]_i$ indicates the ith element of a vector. A schema of SBC is shown in Figure 1 left.

Training. A softmax-based classifier is typically trained by optimizing the cross-entropy loss function. For a given input $\mathbf{x}$ and corresponding ground truth label y^+, the cross-entropy loss is $\mathcal{L}_{CE}(\boldsymbol{\theta}; \mathbf{x}, y^+) = -\log(p_{\boldsymbol{\theta}}(y^+|\mathbf{x}))$.

Inference. Given an input $\mathbf{x}$, the class label predicted by the softmax-based classifier is the class with the largest conditional probability $\hat{y} = \arg\max_{y \in \mathcal{Y}} p_{\boldsymbol{\theta}}(y|\mathbf{x})$.

3.2. Softmax-based classifiers for continual learning

When used for continual learning, and in particular when used for class-incremental learning, softmax-based classifiers face several challenges. One important issue is that softmax-based classifiers compute the cross-entropy loss over all classes (or sometimes over all classes that have been seen so far). As a result, when training on a new task, the likelihood of the currently observed classes is increased, but the likelihood of old classes is too heavily suppressed since they are not encountered in the new task. The softmax operation introduces competitive, winner-take-all dynamics that make the classifier catastrophically forget past tasks. We show such phenomenon of SBC in Section 5.1.3.

4. CL with Energy-Based Models

In this section, we propose a simple but efficient energy-based training objective that can successfully mitigate the catastrophically forgetting in CL. We first introduce EBMs in section 4.1 and then show how EBMs are used for classification in section 4.2 and continual learning in sections 4.3.

4.1. Energy-based models

EBMs (LeCun et al., 2006) are a class of maximum likelihood models that define the likelihood of a data point $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^D$ using the Boltzmann distribution:

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \frac{\exp(-E_{\boldsymbol{\theta}}(\mathbf{x}))}{Z(\theta)}, \quad Z(\boldsymbol{\theta}) = \int_{\mathbf{x} \in \mathcal{X}} \exp(-E_{\boldsymbol{\theta}}(\mathbf{x})) \tag{2}$$

where $E_{\boldsymbol{\theta}}(\mathbf{x}) : \mathbb{R}^D \rightarrow \mathbb{R}$, known as the energy function, maps each data point $\mathbf{x}$ to a scalar energy value, and $Z(\boldsymbol{\theta})$ is the partition function. In deep learning applications, the energy function $E_{\boldsymbol{\theta}}$ is a neural network parameterized by $\boldsymbol{\theta}$.

EBMs are powerful models that have been applied to different domains, such as structured prediction (Belanger & McCallum, 2016; Gygli et al., 2017; Rooshenas et al., 2019; Tu & Gimpel, 2019), machine translation (Tu et al., 2020), text generation (Deng et al., 2020), reinforcement learning (Haarnoja et al., 2017), image generation (Salakhutdinov & Hinton, 2009; Xie et al., 2016; 2018; Nijkamp et al., 2019), memory modeling (Bartunov et al., 2019), classification (Grathwohl et al., 2019), and biologically-plausible training (Scellier & Bengio, 2017). As far as we are aware, EBMs for continual learning has so far remained unexplored.

4.2. Energy-based models for classification

To solve the classification tasks, we adapt the above general formulation of an EBM as follows. Given inputs $\mathbf{x} \in \mathbb{R}^D$

and a discrete set $\mathcal{Y}$ of possible class labels, we propose to use the Boltzmann distribution to define the conditional likelihood of label y given $\mathbf{x}$:

$$p_{\boldsymbol{\theta}}(y|\mathbf{x}) = \frac{\exp(-E_{\boldsymbol{\theta}}(\mathbf{x}, y))}{Z(\boldsymbol{\theta}; \mathbf{x})}, Z(\boldsymbol{\theta}; \mathbf{x}) = \sum_{y' \in \mathcal{Y}} \exp(-E_{\boldsymbol{\theta}}(\mathbf{x}, y')) \tag{3}$$

where $E_{\boldsymbol{\theta}}(\mathbf{x}, y) : (\mathbb{R}^D, \mathbb{N}) \to \mathbb{R}$ is the energy function that maps an input-label pair $(\mathbf{x}, y)$ to a scalar energy value, and $Z(\boldsymbol{\theta}; \mathbf{x})$ is the partition function for normalization.

Training. We want the distribution defined by $E_{\boldsymbol{\theta}}$ to model the data distribution p_D, which we do by minimizing the negative log likelihood of the data

$$\mathcal{L}_{\text{ML}}(\boldsymbol{\theta}) = \mathbb{E}_{(x,y)\sim p_D}[-\log p_{\boldsymbol{\theta}}(y|\mathbf{x})] \tag{4}$$

with the expanded form:

$$\mathcal{L}_{\text{ML}}(\boldsymbol{\theta}) = \mathbb{E}_{(\mathbf{x},y)\sim p_D}\left[E_{\boldsymbol{\theta}}(\mathbf{x}, y) + \log(\textstyle\sum_{y' \in \mathcal{Y}} e^{-E_{\boldsymbol{\theta}}(\mathbf{x}, y')})\right]. \tag{5}$$

Equation 5 minimizes the energy of $\mathbf{x}$ at the ground truth label y and minimizes the overall partition function by increasing the energy of $\mathbf{x}$ at other labels y'.

Inference. Given an input $\mathbf{x}$, the class label predicted by our EBMs is the class with the smallest energy at $\mathbf{x}$:

$$\hat{y} = \underset{y' \in \mathcal{Y}}{\arg\min}\, E_\theta(\mathbf{x}, y'), \tag{6}$$

4.3. Energy-based models for continual learning

4.3.1. EBM TRAINING OBJECTIVE

We notice that in Equation 5, the energy over all class labels y' given data $\mathbf{x}$ are maximized. Directly maximizing energy across all labels raises the same problem as the softmax-based classifier models that the old classes are suppressed when training a model on new classes and thus cause the catastrophic forgetting. Inspired by (Hinton, 2002; Du & Mordatch, 2019; Xie et al., 2016), we find that the contrastive divergence approximation of Equation 5 can mitigate this problem and lead to a simpler equation. To do so, we define the following contrastive divergence loss:

$$\mathcal{L}_{\text{CD}}(\boldsymbol{\theta}; \mathbf{x}, y) = \mathbb{E}_{(\mathbf{x},y)\sim p_D}\left[E_{\boldsymbol{\theta}}(\mathbf{x}, y) - E_{\boldsymbol{\theta}}(\mathbf{x}, y^-)\right], \tag{7}$$

where y is the ground truth label of data $\mathbf{x}$ and y^- is a negative class label randomly sampled from the set of class labels in the current training batch $\mathcal{Y}_B$ such that $y^- \neq y$.

Different from the softmax-based classifier, EBMs maximize likelihood not by normalizing over all classes but instead by contrastively increasing the energy difference between the ground truth label and another negative label for a given data point. This operation causes less interference with previous classes and enables EBMs to suffer less from catastrophic forgetting.

Importantly, such a sampling strategy also allows our EBMs to be naturally applied, without any modification, to different CL settings described in Section 2.1. Since we select the negative sample(s) from the current batch, our EBMs do not require knowledge of the task on hand. This allows application of EBMs in the *boundary-agnostic* setting, in which the underlying tasks or task boundaries are not given.

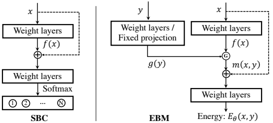

Figure 1: Schematic of the model architectures of the softmax-based classifier (SBC) and energy-based models (EBM). SBC takes an image $\mathbf{x}$ as input and outputs a fixed pre-defined N-dimensional vector to represent the probabilities of N different classes. EBM takes a data $\mathbf{x}$ and a class y as input and outputs their energy value. The dash lines are optional skip connections.

We find that the proposed EBM training objective is efficient enough to achieve good performance on different CL datasets (Table 1 and Table 7). We note however that it is possible to use other strategies for choosing the negative classes in the partition function in Equation 5. In Table 3, we explore alternative strategies: 1) using all classes in the current training batch $\mathcal{Y}_B$ as negative classes, and 2) using all classes seen so far as negative classes. The usage of negative samples in the EBM training objective provides freedom for models to choose which classes to train on which is important for preventing catastrophic forgetting in continual learning.

4.3.2. ENERGY NETWORK

Another important difference from softmax-based classifiers is that the choice of model architectures becomes more flexible in EBMs. Traditional classification models only feed in $\mathbf{x}$ as input. In contrast, EBMs have many different ways to combine $\mathbf{x}$ and y in the energy function with the only requirement that $E_\theta(\mathbf{x}, y) : (\mathbb{R}^D, \mathbb{N}) \to \mathbb{R}$. In EBMs, we can treat y as an attention filter or gate to select the most relevant information between $\mathbf{x}$ and y.

To compute the energy of any data $\mathbf{x}$ and class label y pair, we use y to influence a conditional gain on $\mathbf{x}$, which serves as an attention filter (Xu et al., 2015) to select the most relevant information between $\mathbf{x}$ and y. In Figure 1 (right), we first send $\mathbf{x}$ into a small network to generate the feature $f(\mathbf{x})$. The label y is mapped into a same dimension feature $g(y)$ using a small learned network or a random projection. We use the gating block G to select the most relevant information between $\mathbf{x}$ and y:

$$m(\mathbf{x}, y) = G(f(\mathbf{x}), g(y)). \tag{8}$$

The output is finally sent to weight layers to generate the energy value $E_\theta(\mathbf{x}, y)$. See Supplement Section C for more details about our model architectures.

Table 1: Evaluation of class-incremental learning on the *boundary-aware* setting on four datasets. Each experiment is performed at least 10 times with different random seeds, the results are reported as the mean ± SEM over these runs. Note our comparison is restricted to methods that do not replay stored or generated data.

Method	**splitMNIST**	**permMNIST**	**CIFAR-10**	**CIFAR-100**
SBC	19.90 ± 0.02	17.26 ± 0.19	19.06 ± 0.05	8.18 ± 0.10
EWC	20.01 ± 0.06	25.04 ± 0.50	18.99 ± 0.03	8.20 ± 0.09
Online EWC	19.96 ± 0.07	33.88 ± 0.49	19.07 ± 0.13	8.38 ± 0.15
SI	19.99 ± 0.06	29.31 ± 0.62	19.14 ± 0.12	9.24 ± 0.22
LwF	23.85 ± 0.44	22.64 ± 0.23	19.20 ± 0.30	10.71 ± 0.11
MAS	19.50 ± 0.30	-	20.25 ± 1.54	8.44 ± 0.27
BGD	19.64 ± 0.03	84.78 ± 1.30	-	-
EBM	$\mathbf{53.12 \pm 0.04}$	$\mathbf{87.58 \pm 0.50}$	$\mathbf{38.84 \pm 1.08}$	$\mathbf{30.28 \pm 0.28}$

Our EBMs allow any number of classes in new batches by simply training or defining a new conditional gain $g(\mathbf{y})$ for the new classes and generating its energy value with data point $\mathbf{x}$. This formulation gives us freedom to learn new classes without pre-defining their number in advance. We note that the standard classifiers have to change their model architecture by adding new class heads to the softmax output layer when dealing with new classes. In contrast, EBMs do not need to modify the model architecture or resize the network when adding new classes.

4.3.3. INFERENCE

During inference, because we evaluate according to the class-incremental learning scenario (van de Ven & Tolias, 2019; Tao et al., 2020a; He et al., 2018), the model must predict a class label by choosing from all classes seen so far. Let $\mathbf{x}_k$ be one data point from a batch $\mathcal{B}_k$ with an associated discrete label $y \in \mathcal{Y}_k$, where $\mathcal{Y}_k$ contains the class labels in $\mathcal{B}_k$. Then there are $\mathcal{Y} = \bigcup_{k=1}^K \mathcal{Y}_k$ different class labels in total after seeing all the batches. The MAP estimate is

$$\hat{y} = \underset{y}{\arg\min}\, E_{\theta_K}(\mathbf{x}_k, y), \quad y \in \bigcup \mathcal{Y}_k, \tag{9}$$

where $E_{\theta_K}(\mathbf{x}_k, y)$ is the energy function with parameters θ_K resulting from training on the batches $\{\mathcal{B}_1, \cdots, \mathcal{B}_K\}$. The energy function can compute an energy for any discrete class input, including unseen classes, which avoids needing to predefine the number of classes in advance.

4.3.4. ALTERNATIVE EBM TRAINING OBJECTIVE

EBMs are not limited to modeling the conditional distribution between $\mathbf{x}$ and y as shown in Equation 3. Another way to use the Boltzmann distribution is to define the joint likelihood of $\mathbf{x}$ and y. Then the training objective becomes:

$$\mathcal{L}_{\text{CD}}(\boldsymbol{\theta}; \mathbf{x}, y) = \mathbb{E}_{(\mathbf{x},y)\sim p_D}\left[E_{\boldsymbol{\theta}}(\mathbf{x}, y) - E_{\boldsymbol{\theta}}(\mathbf{x}', y')\right], \tag{10}$$

where $\mathbf{x}'$ and y' are randomly sampled from the current training batch. Since the main focus of this paper is to propose a simple but efficient EBM training objective for CL, we only show the results of using Equation 7 in this main paper. In Supplement Section A, we show that the new training objective in Equation 10 can further improve the results.

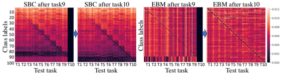

Figure 2: Energy landmaps of SBC and EBMs after training on task T_9 and T_{10} on permuted MNIST. The darker the diagonal is, the better the model is in preventing forgetting previous tasks.

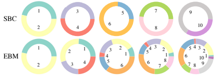

Figure 3: Predicted label distribution after learning each task on the split MNIST dataset. The SBC only predicts classes from the current task, while our EBM predicts classes for all seen classes.

5. Experiments

In this section, we want to answer the following questions: How do the proposed EBMs perform on different CL settings? Can we apply the EBM training objective to other methods? Can we qualitatively understand the differences between EBMs and baselines? Is the proposed EBM training objective better than other objectives? What is the best architecture for label conditioning? And can EBMs combine with existing CL approaches. To answer these questions, we first report experiments on the *boundary-aware* setting in Section 5.1. We then show that EBMs can also be applied to the *boundary-agnostic* setting in Section 5.2.

5.1. Experiments on Boundary-Aware Setting

5.1.1. DATASETS AND EVALUTION PROTOCOLS

Datasets. We evaluate the proposed EBMs on the split MNIST (Zenke et al., 2017), permuted MNIST (Kirkpatrick et al., 2017), CIFAR-10 (Krizhevsky et al., 2009), and CIFAR-100 (Krizhevsky et al., 2009) datasets. The split MNIST dataset is obtained by splitting the original MNIST (LeCun et al., 1998) into 5 tasks with each task having 2 classes. It has 60,000 training images and 10,000 test images. The permuted MNIST protocol has 10 tasks and each task has 10 classes. We separate CIFAR-10 into 5 tasks, each task with 2 classes. CIFAR-100 is split into 10 tasks with each task having 10 classes. The last two datasets each have 50,000 training images and 10,000 test images.

Evaluation protocols. As noted by (van de Ven & Tolias, 2019), the above task protocols could be evaluated according either the task-incremental, domain-incremental, or class-incremental learning scenario. Most CL approaches perform well on the first two simpler scenarios, but fail when asked to do class-incremental learning, which is considered as the most natural and also the hardest setting for continual learning (Tao et al., 2020a; He et al., 2018; Tao et al., 2020b). In this paper, we perform all experiments according to the class-incremental learning scenario.

Table 2: Continual learning approaches using our training objective and their original one.

	split MNIST		CIFAR-10	
	Original	Ours	Original	Ours
SBC	19.90 ± 0.02	44.98 ± 0.05	19.06 ± 0.05	19.22 ± 1.12
EWC	20.01 ± 0.06	50.68 ± 0.04	18.99 ± 0.03	36.51 ± 1.20
Online EWC	19.96 ± 0.07	50.99 ± 0.03	19.07 ± 0.13	36.16 ± 1.02
SI	19.99 ± 0.06	49.44 ± 0.03	19.14 ± 0.12	35.12 ± 1.70
EBM	-	**53.12** ± **0.04**	-	**38.84** ± **1.08**

Table 3: Performance of EBM on CIFAR-100 with different strategies for selecting the negative samples.

Dataset	**CIFAR-100**
All Neg Seen	8.07 ± 0.10
All Neg Batch	29.03 ± 0.53
1 Neg Batch	**30.28** ± **0.28**

5.1.2. COMPARISONS WITH EXISTING METHODS

The most successful existing methods for *Class-IL* either rely on an external quota of memory (Rebuffi et al., 2017; Hayes et al., 2020) or on using generative replay (Shin et al., 2017; van de Ven et al., 2020). A disadvantage of these methods is that they are relatively expensive in terms of memory and/or computation. In this paper, we focus on CL without using replay and without storing data.

Comparisons with baselines. We compare our proposed EBM method with available baseline models that do not use replay, including the standard softmax-based classifier (SBC), EWC (Kirkpatrick et al., 2017), Online EWC (Schwarz et al., 2018), SI (Zenke et al., 2017), LwF (Li & Hoiem, 2017), MAS (Aljundi et al., 2019), and BGD (Zeno et al., 2018). The *Class-IL* results on four datasets are shown in Table 1. For SBC, EWC, Online EWC, Online EWC, LwF on splitMNIST, permuted MNIST, and CIFAR-100, we use the results reported in (van de Ven & Tolias, 2019; van de Ven et al., 2020). For BGD, we use the results from (Zeno et al., 2018). For MAS, we use the result from (Prabhu et al., 2020; Zhang et al., 2020). We performed the other experiments ourselves using the public available code.

All the baselines and EBMs use similar model architectures with similar number of model parameters for fair comparison. For split MNIST and permuted MNIST, we use several fully-connected layers. For CIFAR-10 and CIFAR-100, we use a convolutional network (Supplement Section C for details). For CIFAR-100, all compared models used convolutional layers that were pre-trained on CIFAR-10. Each experiment was performed at least 10 times with different random seeds, with results reported as the mean ± SEM. Similar training regimes were used for the EBMs and baselines. On split MNIST, permuted MNIST, and CIFAR-10, we trained for 2000 iterations per task. On CIFAR-100, we trained for 5000 iterations per task. All experiments used the Adam optimizer with learning rate $1e^{-4}$. As shown in Table 1, EBMs have a significant improvement over the baseline methods on all the datasets, showing that EBMs forget less when updating models for new tasks.

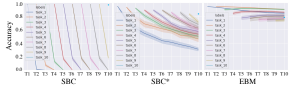

Figure 4: *Class-IL* testing accuracy of the standard classifier (SBC), classifier using our training objective (SBC*), and EBMs on each task on the permuted MNIST dataset.

EBM training objective on existing approaches. Our proposed energy-based training objective is simple and can also be directly applied to existing CL approaches. We test this by modifying the training objective of baseline models to that of our proposed energy objective, which computes the softmax normalization only over class labels in the current training batch. In Table 2, we find that our proposed objective significantly improves the performance of different CL methods. This is because the new training objective does not suppress the probability of old classes when improving the probability of new classes. Our training objective provides an orthogonal direction to tackle the CL problem and is simple to implement on existing CL approaches.

5.1.3. QUALITATIVE ANALYSIS

Most existing CL methods are based on the softmax-based classifiers (Pellegrini et al., 2019; Zenke et al., 2017). To better understand why EBMs suffer less from catastrophic forgetting, we qualitatively compare our EBMs and the standard SBC model shown in Figure 1 in this part.

Energy landscape. In Figure 2, we show the energy landscapes after training on task 9 and task 10 of the permuted MNIST dataset. For SBC, the energy is given by the negative of the predicted probability. Each datapoint has 100 energy values (EBM) or probabilities (SBC) corresponding to the 100 labels in the dataset. For each datapoint, these values are normalized over all 100 classes. Dark elements on the diagonal indicate correct predictions. After training on task T_9, SBC assigns high probabilities to classes from T_9 (80-90) for almost all the data from T_1 to T_9. After learning task T_{10}, the highest probabilities shift to classes from task T_{10} (90-100). SBC tends to assign high probabilities to new classes for both old and new data, indicating forgetting. In contrast, EBM has low energies across the diagonal, which means that after training on new tasks, EBM still assigns low energies to the true labels of data from previous tasks. This shows that EBM is better than SBC at learning new tasks without catastrophically forgetting of old tasks.

Predicted class distribution. In Figure 3, for the split MNIST dataset, we plot the proportional distribution of predicted classes. Only data from the tasks seen so far was used for this figure. Taking the second panel in the first row as an example, it shows the distribution of predicted labels on test data from the first two tasks after finishing training on the second task. This means that so far the model has seen

Table 4: Performance of EBM on CIFAR-10 with different label conditioning architectures.

Model architectures		Normalization types	
Beginning (V1)	13.69 ± 1.12	End Fix (V4)	34.30 ± 1.03
Middle(V2)	20.16 ± 1.05	End Fix Norm2 (V4)	33.91 ± 1.13
Middle(V3)	18.36 ± 0.97	End Fix Softmax (V4)	35.97 ± 1.09
End (V4)	38.13 ± 0.59	End Norm2 (V4)	37.23 ± 1.20
		End Softmax (V4)	**38.84 ± 1.08**

Table 5: Comparison of our EBM with baselines on different variants of the split CIFAR-100 protocol. Results of the *Class-IL* performance on the *boundary-aware* setting are reported.

Method	CIFAR-100 split up into: 5 tasks	10 tasks	20 tasks	50 tasks
SBC	14.74 ± 0.20	8.18 ± 0.10	4.46 ± 0.03	1.91 ± 0.02
EWC	14.78 ± 0.21	8.20 ± 0.09	4.46 ± 0.03	1.91 ± 0.02
SI	14.07 ± 0.24	9.24 ± 0.22	4.37 ± 0.04	1.88 ± 0.03
LwF	25.75 ± 0.14	10.71 ± 0.11	12.18 ± 0.16	7.68 ± 0.16
EBM	**34.88 ± 0.14**	**30.28 ± 0.28**	**25.04 ± 0.33**	**13.60 ± 0.50**

four classes: {1,2,3,4}. Since the number of test images from each class are similar, the ground truth proportional distribution should be uniform over those four classes. After training on the first task, the predictions of SBC are indeed roughly uniformly distributed over the first two classes (first panel). However, after learning new tasks, SBC only predicts classes from the most recent task, indicating that SBC fails to correctly memorize classes from previously seen tasks. In contrast, the predictions of EBM are substantially more uniformly distributed over all classes seen so far.

In the Supplement, we provide further analyses comparing the continual learning behavior of SBC and EBMs, including confusion matrices in Supplement Section B.1, an analysis of model capacity in Supplement Section B.2, a comparison of parameter importance in Supplement Section B.3, and an evaluation of the interference with previously learned tasks in Supplement Section B.4.

5.1.4. Is the strong performance of EBMs due to the Energy Training Objective or due to the Label Conditioning?

Effect of energy training objective. We conduct an experiment on the CIFAR-100 dataset to investigate how different EBM training objectives influence the CL results. We compare three different strategies for selecting the negative samples as described in Section 4.3.1. The first strategy uses all seen classes so far as negative labels (**All Neg Seen**), which is most similar to the way that the traditional classifier is optimized. The second one takes all the classes in the current batch as negative labels (**All Neg Batch**). The last one randomly selects one class from the current batch as the negative as we mentioned in Equation 7 (**1 Neg Batch**). In Table 3, we find using only one negative sample generates the best result, and using negatives sampled from classes in the current batch is better than from all seen classes. Since our EBM training objective aims at improving the energy of negative samples while decreasing the energy of positive ones, sampling negatives from the current batch has less interference with previous classes than sampling from all seen classes. Using a single negative causes the minimum suppression on negative samples and thus has the best result.

Effect of label conditioning. Next, we test whether the label conditioning in our EBMs is important for their performance in Table 2. As mentioned in Section 5.1.2, we modify baseline models using our training objective. EBMs still outperform the modified baselines, implying that the label conditioning architecture also contributes to why EBMs suffer less from catastrophic forgetting.

We further show the testing accuracy of each task as the training progresses in Figure 4. We compare the standard classifier (SBC), classifier using our training objective (SBC*), and our EBMs on permuted MNIST. We find that the accuracy of old tasks in SBC drop sharply when learning new tasks, while the EBM training objective used in SBC* is better. The curve on EBMs drops even slower than SBC*, implying our EBMs can mitigate the forgetting problem.

To summarize, we show that the strong performance of our EBMs is due to both the EBM training objective and the label conditioning architecture. Moreover, these results indicate that surprisingly, and counterintuitively, directly optimizing the cross-entropy loss used by existing approaches may not be the best way to approach continual learning.

5.1.5. Comparison of different architectures

EBMs allow flexibility in integrating data information and label information in the energy function. To investigate where and how to combine the information from data $\mathbf{x}$ and label y, we conduct a series of experiments on CIFAR-10. Table 4 shows four model architectures (V1-V4) that combine $\mathbf{x}$ and y in the early, middle, and late stages, respectively (see Supplement Section C for more details). We find combining $\mathbf{x}$ and y in the late stage (V4) performs the best. We note that instead of learning a feature embedding of label y, we can use a fixed projection matrix which is randomly sampled from the uniform distribution $\mathcal{U}(0,1)$. Even using this fixed random projection can already generates better results than most baselines in Table 1. Note further that the number of trainable parameters in the "Fix" setting is much lower than that of the baselines. Using a learned feature embedding of y can further improve the result. We may also apply different normalization methods over the feature channel of y. We find that Softmax (**End Softmax (V4)**) is better than the L2 normalization (**End Norm2 (V4)**) and no normalization (**End (V4)**).

5.1.6. Effect of different numbers of tasks

To test the generality of our proposed EBMs, in Table 5 we repeat the boundary-aware experiments on CIFAR-100 for different number of classes per task. In Table 1, the CIFAR-100 dataset was split up into 10 tasks, resulting in 10 classes per task. Here we additionally split CIFAR-100 up into 5 tasks (i.e., 20 classes per task), 20 tasks (i.e., 5 classes per task) and 50 tasks (i.e., 2 classes per task). Our EBM substantially outperforms the baselines on all settings.

Table 6: Comparisons of the softmax-based classifier and EBMs using exact replay on four datasets. Results of *Class-IL* on the *boundary-aware* setting are reported. k is the memory budget size.

Method	splitMNIST	permMNIST	CIFAR-10	CIFAR-100
	k=1000	k=1000	k=1000	k=2000
SBC ER	90.65 ± 0.45	93.70 ± 0.09	42.07 ± 0.64	28.57 ± 0.35
EBM ER	91.13 ± 0.35	94.59 ± 0.09	44.76 ± 0.73	34.07 ± 0.55

5.1.7. Can EBMs use Replay?

Although EBMs already achieve good performance on *Class-IL* without using any replay, we found that EBMs are flexible enough to be combined with replay-based approaches to further improve the performance. In Table 6, we show the results of SBC and EBM using exact replay. When training a new task, we mix the new data with data sampled from a memory buffer that stores examples of previously learned tasks to train the models. The examples stored in the buffer are randomly selected from the classes encountered so far, and the available memory budget k is equally divided over all classes encountered so far. We provide more details in the Supplement Section D.

We report the results on 4 datasets. Note these numbers might be slightly different from results reported in some existing works because of the usage of different model architectures, different memory sizes, and different ways to split the datasets. In our experiments, we control the baselines and EBMs to have similar model architectures with similar number of model parameters and the same buffer sizes on each dataset. After using extra memory, both SBC and EBM have improvements. EBMs still outperform SBC, especially on more challenging datasets, e.g., CIFAR-10 and CIFAR-100. Interestingly, on CIFAR-100, we find that EBMs without using replay (30.28%; Table 1) perform better than SBC with using replay (28.57%; Table 6).

Again our focus is try to address the CL problems without using replay or stored data. Our results show that the proposed EBM formulation provides an orthogonal direction to tackle the CL problems. This approach can be further combined with existing CL methods.

5.2. Experiments on Boundary-Agnostic Setting

When applying continual learning in real life, boundaries are not usually well defined between different tasks. However, most existing continual learning methods rely on the presence of sharp boundaries between tasks to determine when to consolidate the knowledge. We show that EBMs are able to flexibly perform CL across different setups, and perform well on the *boundary-agnostic* setting as well.

Table 7: Evaluation of class-incremental learning performance on the *boundary-agnostic* setting. Each experiment is performed 5 times with different random seeds, average test accuracy is reported as the mean ± SEM over these runs. Note that our comparison is restricted to methods that do not replay stored or generated data.

Method	splitMNIST	permMNIST	CIFAR-10	CIFAR-100
SBC	24.03 ± 0.59	21.42 ± 0.11	23.30 ± 0.81	9.85 ± 0.02
Online EWC	39.62 ± 0.14	41.37 ± 0.04	22.53 ± 0.41	9.57 ± 0.02
SI	28.79 ± 0.24	35.71 ± 0.11	26.26 ± 0.72	10.42 ± 0.01
BGD	21.65 ± 1.15	26.15 ± 0.22	17.03 ± 0.82	8.50 ± 0.02
EBM	**81.78 ± 1.22**	**92.35 ± 0.11**	**49.47 ± 1.25**	**34.39 ± 0.24**

5.2.1. Datasets and evalution protocols

For the *boundary-agnostic* setting described in Section 2.1, we use the same datasets as the *boundary-aware* setting in Section 5.1.1. We use the code of "continuous task-agnostic learning" proposed by (Zeno et al., 2018) to generate a continually changing data stream. In this setting, the frequency of each subsequent class increases and decreases linearly. All experiments are performed according to the *Class-IL* scenario.

5.2.2. Comparison with existing methods

We focus on the investigation of the performance of different model architectures with similar model size and memory footprint, and thus do no compare with replay-based methods. Since there is no knowledge on the number of tasks, many methods of continual learning that rely on task boundaries are generally inapplicable. One trivial adaptation is to take the core action after every batch step instead of every task. However, doing such adaptation is impractical for most algorithms, such as EWC, because of the large computational complexity. However, we managed to run the Online EWC, SI, and BGD baselines in this way. All compared methods used similar model architectures as in the *boundary-aware* setting. Each experiment was performed 5 times with different random seeds, the results are reported as the mean ± SEM over these runs.

The results are shown in Table 7. We observe that EBMs have a significant improvement on all the datasets. The experiments show that EBMs have good generalization ability for different continual learning problems as EBMs can naturally handle data streams with and without task boundaries.

6. Discussion and Future Work

In this paper, we show that energy-based models are a promising class of models in a variety of different continual learning settings. We demonstrate that EBMs exhibit many desirable characteristics to prevent catastrophic forgetting in continual learning, and we experimentally show that EBMs obtain strong performance on the challenging class-incremental learning scenario on multiple benchmarks, both on the boundary-aware and boundary-agnostic settings. Additionally, we found that EBMs have larger model capacity than other models which might be important for future