Energy-based Latent Aligner for Incremental Learning

K J Joseph[†‡] Salman Khan[‡⋆] Fahad Shahbaz Khan[‡⋄] Rao Muhammad Anwer[‡¶]
Vineeth N Balasubramanian[†]

[†]Indian Institute of Technology Hyderabad, India [‡]Mohamed bin Zayed University of AI, UAE
[⋆]Australian National University, Australia [⋄]Linköping University, Sweden [¶]Aalto University, Finland
{cs17m18p100001, vineethnb}@iith.ac.in, {salman.khan, fahad.khan, rao.anwer}@mbzuai.ac.ae

Abstract

*Deep learning models tend to forget their earlier knowledge while incrementally learning new tasks. This behavior emerges because the parameter updates optimized for the new tasks may not align well with the updates suitable for older tasks. The resulting latent representation mismatch causes forgetting. In this work, we propose ELI: **E**nergy-based **L**atent Aligner for **I**ncremental Learning, which first learns an energy manifold for the latent representations such that previous task latents will have low energy and the current task latents have high energy values. This learned manifold is used to counter the representational shift that happens during incremental learning. The implicit regularization that is offered by our proposed methodology can be used as a plug-and-play module in existing incremental learning methodologies. We validate this through extensive evaluation on CIFAR-100, ImageNet subset, ImageNet 1k and Pascal VOC datasets. We observe consistent improvement when ELI is added to three prominent methodologies in class-incremental learning, across multiple incremental settings. Further, when added to the state-of-the-art incremental object detector, ELI provides over 5% improvement in detection accuracy, corroborating its effectiveness and complementary advantage to the existing art. Code is available at: https://github.com/JosephKJ/ELI.*

1. Introduction

Learning experiences are dynamic in the real-world, requiring models to incrementally learn new capabilities over time. Incremental Learning (also called continual learning) is a paradigm that learns a model $\mathcal{M}^{\mathcal{T}_t}$ at time step t, such that it is competent in solving a continuum of tasks $\mathcal{T}_t = \{\tau_1, \tau_2, \cdots, \tau_t\}$ introduced to it during its lifetime. Each task τ_i contains instances from a disjoint set of classes. Importantly, the training data for the previous tasks $\{\tau_1, \cdots, \tau_{t-1}\}$ cannot be accessed while learning τ_t, due to privacy, memory and/or computational constraints.

We can represent an incremental model $\mathcal{M}^{\mathcal{T}_t}$, as a composition of a latent feature extractor $\mathcal{F}_{\boldsymbol{\theta}}^{\mathcal{T}_t}$ and a trailing network $\mathcal{F}_{\boldsymbol{\phi}}^{\mathcal{T}_t}$ that solves the task using the extracted features:

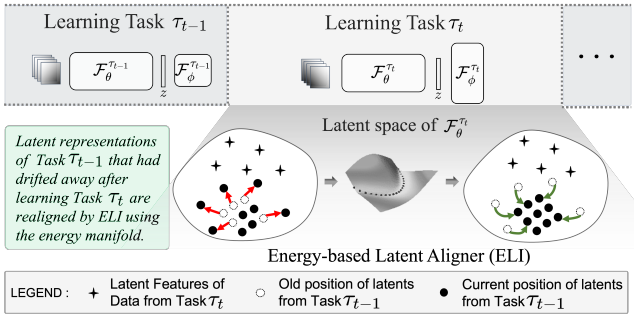


Figure 1. We illustrate an Incremental Learning model trained on a continuum of tasks in the top part of the figure. While learning the current task τ_t (zoomed-in), the latent representation of Task τ_{t-1} data gets disturbed, as shown by red arrows. ELI learns an energy manifold, and uses it to counteract this inherent representational shift, as illustrated by **green** arrows, thereby alleviating forgetting.

$\mathcal{M}^{\mathcal{T}_t}(\mathbf{x}) = (\mathcal{F}_{\boldsymbol{\phi}}^{\mathcal{T}_t} \circ \mathcal{F}_{\boldsymbol{\theta}}^{\mathcal{T}_t})(\mathbf{x})$; where $\mathbf{x} \in \mathcal{T}_t$. A naive approach for learning incrementally would be to use data samples from the current task τ_t to finetune the model trained until the previous task $\mathcal{M}^{\mathcal{T}_{t-1}}$. Doing so will bias the internal representations of the network to perform well on τ_t, inturn significantly degrading the performance on old tasks. This phenomenon is called catastrophic forgetting [13, 35].

The incremental learning problem requires accumulating knowledge over a long range of learning tasks without catastrophic forgetting. The main challenge is how to consolidate conflicting implicit representations across different training episodes to learn a generalized model applicable to all the learning experiences. To this end, existing approaches investigate regularization-based methods [2, 23, 24, 29, 40, 55] that constrain $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ such that the model performs well on all the tasks. Exemplar replay-based methods [7, 8, 21, 33, 41] retain a subset of datapoints from each task, and rehearse them to learn a continual model. Dynamically expanding models [34, 44, 45], enlarge $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ while learning incrementally.

Complementary to the existing methodologies, we introduce a novel approach which minimizes the representational shift in the latent space of an incremental model, using a learned energy manifold. The energy modeling offers a natural mechanism to deal with catastrophic forget-

ting which we build upon. Fig. 1 illustrates how our proposed methodology, ELI: **E**nergy-based **L**atent Aligner for **I**ncremental Learning, helps to alleviate forgetting. After learning the current task τ_t, the features from the feature extractor (referred to as *latents* henceforth), of the previous task data $\mathbf{z}^{\mathcal{T}_t} = \mathcal{F}_{\boldsymbol{\theta}}^{\mathcal{T}_t}(\mathbf{x})$, $\mathbf{x} \in \tau_{t-1}$ drift as shown by the red arrows. The *first* step in our approach is to learn an energy manifold where the latent representations from the model trained until the current task $\mathcal{M}^{\mathcal{T}_t}$ have higher energy, while the latents from the model trained till the previous task $\mathcal{M}^{\mathcal{T}_{t-1}}$ have lower energy. *Next*, the learned energy-based model (EBM) is used to transform the previous task latents $\mathbf{z}^{\mathcal{T}_t}$ (obtained via passing the previous task data through current model) which had drifted away, to alternate locations in the latent space such that the representational shift is undone (as shown by the **green** arrows). This helps alleviate forgetting in incremental learning. We explain how this transformation can be achieved in Sec. 3. We also present a proof-of-concept with MNIST (Fig. 3) which mimics the above setting. The latent space visualization and accuracy regain after learning the new task correlates with the illustration in Fig. 1, which reinforces our intuition.

A unique characteristic of our energy-based latent aligner is its ability to extend and enhance existing continual learning methodologies, without any change to their methodology. We verify this by adding ELI to three prominent class-incremental methods: iCaRL [41], LUCIR [20] and AANet [31] and the state-of-the-art incremental Object Detector: iOD [22]. We conduct thorough experimental evaluation on incremental versions of large-scale classification datasets like CIFAR-100 [25], ImageNet subset [41] and ImageNet 1k [9]; and Pascal VOC [12] object detection dataset. For incremental classification experiments, we consider two prominent setups: adding classes to a model trained with half of all the classes as first task, and the general incremental learning setting which considers equal number of classes for all tasks. ELI consistently improves performance across all datasets and on all methods in incremental classification settings, and obtains impressive performance gains on incremental Object Detection, compared to current state-of-the-art [22], by 5.4%, 7% and 3% while incrementally learning 10, 5 and a single class respectively. To summarize, the key highlights of our work are:

- We introduce a novel methodology ELI, which helps to counter the representational shift that happens in the latent space of incremental learning models.
- Our energy-based latent aligner can act as an add-on module to existing incremental classifiers and object detectors, without any changes to their methodology.
- ELI shows consistent improvement on over 45 experiments across three large scale incremental classification datasets, and improves the current state-of-the-art incremental object detector by over 5% mAP on average.

2. Related Work

Incremental Learning: In this setting a model consistently improves itself on new tasks, without compromising its performance on old tasks. One popular approach to achieve this behaviour is by constraining the parameters to not deviate much from previously tuned values [7, 10, 28, 32, 41, 52]. In this regard, knowledge distillation [19] has been used extensively to enforce explicit regularization in incremental classification [7, 28, 41] and object detection [15, 21, 46] settings. In replay based methods, typically a small subset of exemplars is stored to recall and retain representations useful for earlier tasks [6, 20, 24, 32, 41]. Another set of isolated parameter learning methods dedicate separate subsets of parameters to different tasks, thus avoiding interference *e.g.*, by new network blocks or gating mechanisms [1, 31, 38, 39, 44]. Further, meta-learning approaches have been explored to learn the update directions which are shared among multiple incremental tasks [22, 40, 43]. In contrast to these approaches, we propose to learn an EBM to align implicit feature distributions between incremental tasks. ELI can enhance these existing methods without any methodological modifications, by enforcing an implicit latent space regularization using the learned energy manifold.

Energy-based Models: EBMs [26] are a type of maximum likelihood estimation models that can assign low energies to observed data-label pairs and high energies otherwise [11]. EBMs have been used for out-of-distribution sample detection [30, 47], structured prediction [4, 5, 48] and improving adversarial robustness [11, 17]. Joint Energy-based Model (JEM) [14] shows that any classifier can be reinterpreted as a generative model that can model the joint likelihood of labels and data. While JEM requires alternating between a discriminative and generative objective, Wang *et al*. [49] propose an energy-based open-world softmax objective that can jointly perform discriminative learning and generative modeling. EBMs have also been used for synthesizing images [3, 53, 56, 57]. Xie *et al*. [54] represents EBM using a CNN and utilizes Langevin dynamics for MCMC sampling to generate realistic images. In contrast to these methods, we explore the utility of the EBMs to alleviate forgetting in a continual learning paradigm. Most of these methods operate in the data space, where sampling from the EBM would be expensive [56]. Differently, we learn the energy manifold with the latent representations, which is faster and effective in controlling the representational shift that affects incremental models. A recent unpublished work [27] proposes to replace the standard softmax layer of an incremental model with an energy-based classifier head. Our approach introduces an implicit regularization in the latent space using the learned energy manifold which is fundamentally different from their approach, scales well to harder datasets and diverse settings (classification and detection).

3. Energy-based Latent Aligner

Our proposed methodology ELI utilizes an Energy-based Model (EBM) [26] to optimally adapt the latent representations of an incremental model, such that it alleviates catastrophic forgetting. We refer to the intermediate feature vector extracted from the backbone network of the model as *latent* representations in our discussion. After a brief introduction to the problem setting in Sec. 3.1, we explain how the EBM is learned and used for aligning in Sec. 3.2. We conclude with a discussion on a toy experiment in Sec. 3.3.

3.1. Problem Setting

In the incremental learning paradigm, a set of tasks $\mathcal{T}_t = \{\tau_1, \tau_2, \cdots, \tau_t\}$ is introduced to the model over time. τ_t denotes the task introduced at time step t, which is composed of images $\mathbf{X}^{\tau_t}$ and labels $\mathbf{y}^{\tau_t}$ sampled from its corresponding task data distribution: $(\mathbf{x}_i^{\tau_t}, y_i^{\tau_t}) \sim p_{data}^{\tau_t}$. Each task τ_t, contains instances from a disjoint set of classes. We seek to build a model $\mathcal{M}^{\mathcal{T}_t}$, which is competent in solving all the tasks $\mathcal{T}_t$. Without loss of generality $\mathcal{M}^{\mathcal{T}_t}$ can be expressed as a composition of two functions: $\mathcal{M}^{\mathcal{T}_t}(\mathbf{x}) = (\mathcal{F}_{\boldsymbol{\phi}}^{\mathcal{T}_t} \circ \mathcal{F}_{\boldsymbol{\theta}}^{\mathcal{T}_t})(\mathbf{x})$, where $\mathcal{F}_{\boldsymbol{\theta}}^{\mathcal{T}_t}$ is a feature extractor and $\mathcal{F}_{\boldsymbol{\phi}}^{\mathcal{T}_t}$ is a classifier in the case of a classification model and a composite classification and localization branch for an object detector, solving all the tasks $\mathcal{T}_t$ introduced to it so far.

While training $\mathcal{M}^{\mathcal{T}_t}$ on current task τ_t, the model does not have access to all the data from previous tasks[1]. This imbalance between present and previous task data can bias the model to focus on the latest task, while catastrophically degrading its performance on the earlier ones. Making an incremental learner robust against such forgetting is a challenging research question. Regularization methods [2, 23], exemplar-replay methods [8, 33, 41] and progressive model expansion methods [34,44,45] have emerged as the standard ways to address forgetting. Our proposed methodology is complementary to all these developments in the field, and is generic enough to serve as an add-on to any such continual learning methodology, with minimal overhead.

3.2. Latent Aligner

We perform energy-based modeling in the latent space of continual learning models. Our latent aligner approach avoids the need to explicitly identify which latent representations should be adapted or retained to preserve knowledge across tasks while learning new skills. It implicitly identifies which representations are ideal to be shared between tasks, preserves them, and simultaneously adapts representations which negatively impact incremental learning.

Let us consider a concrete incremental learning setting where we introduce a new task τ_t to a model that is trained to perform well until the previous tasks $\mathcal{M}^{\mathcal{T}_{t-1}}$. Training data to learn the new task is sampled from the corresponding data distribution: $(\mathbf{x}_i^{\tau_t}, y_i^{\tau_t}) \sim p_{data}^{\tau_t}$. We may use any existing continual learning algorithm $\mathcal{A}$, to learn an incremental model $\mathcal{M}^{\mathcal{T}_t}$. The latent representations of $\mathcal{M}^{\mathcal{T}_{t-1}}$ would be optimized for learning τ_t, which causes degraded performance of $\mathcal{M}^{\mathcal{T}_t}$ on the previous tasks. Depending on the efficacy of $\mathcal{A}$, $\mathcal{M}^{\mathcal{T}_t}$ can have varying degrees of effectiveness in alleviating the inherent forgetting. Our proposed method helps to undo this representational shift that happens to previous task instances, when passed through $\mathcal{M}^{\mathcal{T}_t}$.

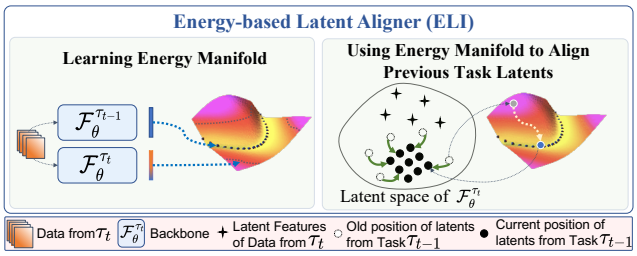


Figure 2. We learn an energy manifold using the latent representations of the current task data passed though the current model $\mathcal{F}_{\boldsymbol{\theta}}^{\mathcal{T}_t}$ and previous model $\mathcal{F}_{\boldsymbol{\theta}}^{\mathcal{T}_{t-1}}$. This manifold is used to align the latents from τ_{t-1} that were shifted while learning the new task.

As illustrated in Fig. 2, in the *first* step, we learn an energy manifold using three ingredients: (i) images from the current task: $\mathbf{x} \sim p_{data}^{\tau_t}$, (ii) latent representations of $\mathbf{x}$ from the model trained till previous task: $\mathbf{z}^{\mathcal{T}_{t-1}} = \mathcal{F}_{\boldsymbol{\theta}}^{\mathcal{T}_{t-1}}(\mathbf{x})$ and (iii) latent representations of $\mathbf{x}$ from the model trained till the current task: $\mathbf{z}^{\mathcal{T}_t} = \mathcal{F}_{\boldsymbol{\theta}}^{\mathcal{T}_t}(\mathbf{x})$. An energy-based model E_ψ is learned to assign low energy values for $\mathbf{z}^{\mathcal{T}_{t-1}}$, and high energy values for $\mathbf{z}^{\mathcal{T}_t}$. *Next*, during inference, the learned energy manifold E_ψ is used to counteract the representational shift that happens to the latent representations of previous task instances when passed through the current model: $\mathbf{z}^{\mathcal{T}_t} = \mathcal{F}_{\boldsymbol{\theta}}^{\mathcal{T}_t}(\mathbf{x})$ where $\mathbf{x} \in \mathcal{T}_{t-1}$. Due to the representational shift in the latent space, $\mathbf{z}^{\mathcal{T}_t}$ will have higher energy values in the energy manifold. We align $\mathbf{z}^{\mathcal{T}_t}$ to alternate locations in latent space such that their energy on the manifold is minimized, as illustrated in right part of Fig. 2. These shifted latents demonstrate less forgetting, which we empirically verify through large scale experiments on incremental classification and object detection in Sec. 4.

It is interesting to note the following: 1) Our method adds implicit regularization in the latent space without making any changes to the incremental learning algorithm $\mathcal{A}$, which is used to learn $\mathcal{M}^{\mathcal{T}_t}$, 2) ELI does not require access to previous task data to learn the energy manifold. Current task data, passed though the model $\mathcal{F}_{\boldsymbol{\theta}}^{\mathcal{T}_{t-1}}$ indeed acts as a proxy for previous task data while learning the EBM.

3.2.1 Learning the Latent Aligner: EBMs provide a simple and flexible way to model data likelihoods [11]. We use continuous energy-based models, formulated using a neu-

[1] Such restricted memory is considered due to practical limitations such as bounded storage, computational budget and privacy issues.

ral network, which can generically model a diverse range of function mappings. Specifically, for a given latent feature vector $\mathbf{z} \in \mathbb{R}^D$ in ELI, we learn an energy function $E_\psi(\mathbf{z}) : \mathbb{R}^D \rightarrow \mathbb{R}$ to map it to a scalar energy value. An EBM is defined as Gibbs distribution $p_\psi(\mathbf{z})$ over $E_\psi(\mathbf{z})$:

$$p_\psi(\mathbf{z}) = \frac{\exp\left(-E_\psi(\mathbf{z})\right)}{\int_{\mathbf{z}} \exp\left(-E_\psi(\mathbf{z})\right) d\mathbf{z}}, \tag{1}$$

where $\int_{\mathbf{z}} \exp\left(-E_\psi(\mathbf{z})\right) d\mathbf{z}$ is an intractable partition function. EBM is trained by maximizing the data log-likelihood on a sample set drawn from the true distribution $p_{true}(\mathbf{z})$:

$$L(\psi) = \mathbb{E}_{\mathbf{z} \sim p_{true}}[\log p_\psi(\mathbf{z})]. \tag{2}$$

The derivative of the above objective is as follows [51]:

$$\partial_\psi L(\psi) = \mathbb{E}_{\mathbf{z} \sim p_{true}}[-\partial_\psi E_\psi(\mathbf{z})] + \mathbb{E}_{\mathbf{z} \sim p_\psi}[\partial_\psi E_\psi(\mathbf{z})]. \tag{3}$$

The first term in Eq. 3 ensures that the energy for a sample $\mathbf{z}$ drawn from the true data distribution p_{true} will be minimized, while the second term ensures that the samples drawn from the model itself, will have higher energy values. In ELI, p_{true} corresponds to the distribution of latent representations from the model trained till the previous task at any point in time. Sampling from $p_\psi(\mathbf{x})$ is intractable owing to the normalization constant in Eq. 1. Approximate samples are recursively drawn using Langevin dynamics [36, 50], which is a popular MCMC algorithm,

$$\mathbf{z}_{i+1} = \mathbf{z}_i - \frac{\lambda}{2} \partial_{\mathbf{z}} E_\psi(\mathbf{z}) + \sqrt{\lambda}\omega_i, \omega_i \sim \mathcal{N}(0, \mathbf{I}) \tag{4}$$

where λ is the step size and ω captures data uncertainty. Eq. 4 yields a Markov chain that stabilizes to an stationary distribution within few iterations, starting from an initial $\mathbf{z}_i$.

Algorithm 1 illustrates how the energy manifold is learned in ELI. The energy function E_ψ is realised by a multi-layer perceptron with a single neuron in the output layer, which quantifies energy of the input sample. It is Kaiming initialized in Line 1. Until a few number of iterations, we sample mini-batches from the current task data distribution $p_{data}^{\mathcal{T}_t}$. Next, the latent representation of the data in the mini-batch is retrieved from the model trained until the previous task $\mathcal{F}_{\boldsymbol{\theta}}^{\mathcal{T}_{t-1}}$ and the model trained till the current task $\mathcal{F}_{\boldsymbol{\theta}}^{\mathcal{T}_t}$, in Line 4 and 5 respectively. From here on, we prepare to compute the gradients according to Eq. 3, which is required for training the energy function. The first term in Eq. 3 minimizes the expectation over in-distribution energies, which is computed in Line 7, while the second term maximizes the expectation over out-of-distribution energies (Line 8). The Langevin sampling which is required to compute the out-of-distribution energies takes the latents from the current model as initial starting points of the Markov chain, as illustrated in Line 6. Finally, the loss is computed in Line 9 and the energy function E_ψ is optimized with RMSprop [18] optimizer in Line 10.

Algorithm 1 Algorithm LEARNEBM

Input: Feature extractor of model trained till current task: $\mathcal{F}_{\boldsymbol{\theta}}^{\mathcal{T}_t}$; Feature extractor of model trained till previous task: $\mathcal{F}_{\boldsymbol{\theta}}^{\mathcal{T}_{t-1}}$; Data distribution of the current task: $p_{data}^{\mathcal{T}_t}$

1: $E_\psi \leftarrow$Initialize the Energy function.
2: **while** until required iterations **do**
3: $\quad \mathbf{x} \sim p_{data}^{\mathcal{T}_t}$ ▷ *Sample a mini-batch*
4: $\quad \mathbf{z}^{\mathcal{T}_{t-1}} \leftarrow \mathcal{F}_{\boldsymbol{\theta}}^{\mathcal{T}_{t-1}}(\mathbf{x})$
5: $\quad \mathbf{z}^{\mathcal{T}_t} \leftarrow \mathcal{F}_{\boldsymbol{\theta}}^{\mathcal{T}_t}(\mathbf{x})$
6: $\quad \mathbf{z}_{sampled}^{\mathcal{T}_t} \leftarrow$ Sample from EBM with $\mathbf{z}^{\mathcal{T}_t}$ as starting points. ▷ *Refer Equation 4*
7: $\quad$ in_dist_energy $\leftarrow E_\psi(\mathbf{z}^{\mathcal{T}_{t-1}})$
8: $\quad$ out_of_dist_energy $\leftarrow E_\psi(\mathbf{z}_{sampled}^{\mathcal{T}_t})$
9: $\quad Loss \leftarrow (-$in_dist_energy $+$ out_of_dist_energy$)$ ▷ *Refer Equation 3*
10: $\quad$ Optimize E_ψ with $Loss$.
11: **return** E_ψ

3.2.2 Alignment using ELI: After learning a task τ_t in an incremental setting, we use Algorithm 1 to learn the energy manifold. This manifold is used to align the latent representations of previous task instances from the current model $\mathcal{M}^{\mathcal{T}_t}$ using Algorithm 2. The gradient of energy function E_ψ with respect to the latent representation $\mathbf{z}$ is computed (Line 2). These latents are then successively updated to reduce their energy (Line 3). We repeat this for L_{steps} number of Langevin iterations. The aligner assumes that a high-level task information is available during inference i.e., whether a latent belongs to the current task or not.

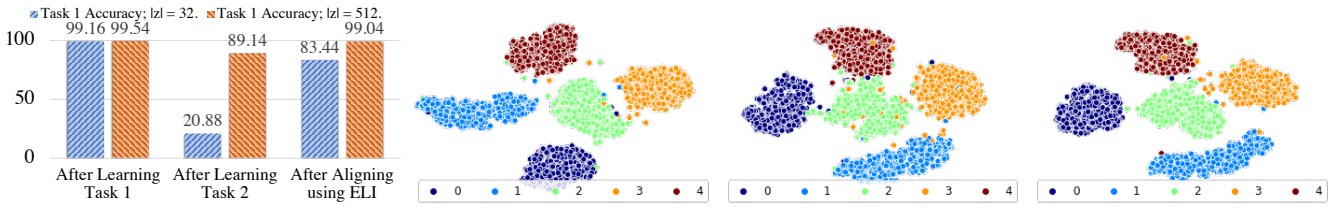
Algorithm 2 Algorithm ALIGNLATENTS

Input: Latent vector to be adapted: $\mathbf{z}$; EBM: E_ψ; Number of Langevin steps: L_{steps}; Learning rate: λ

1: **while** until L_{steps} iterations **do**
2: $\quad grad \leftarrow \nabla_{\mathbf{z}} E_\psi(\mathbf{z})$
3: $\quad \mathbf{z} \leftarrow \mathbf{z} - \lambda * grad$
4: **return** $\mathbf{z}$

3.3. Toy Example

Our methodology is build on a key premise that latent representations of an incremental learning model will get disturbed after training on new tasks, and that an energy-based manifold can aid in successfully mitigating this unwarranted representational shift in a post-hoc fashion. In Fig. 3, we present a proof-of-concept that our hypothesis indeed holds. We consider a two task experiment with incremental MNIST, where the first task is to learn the first 5 classes, while the second is to learn the rest: $\mathcal{T}_1 = \{\tau_0 \ldots \tau_4\}$ and $\mathcal{T}_2 = \{\tau_5 \ldots \tau_9\}$. We first learn $\mathcal{M}^{\mathcal{T}_1}(\mathbf{x}) = (\mathcal{F}_{\boldsymbol{\phi}}^{\mathcal{T}_1} \circ \mathcal{F}_{\boldsymbol{\theta}}^{\mathcal{T}_1})(\mathbf{x})$, where $\mathbf{x} \in \mathcal{T}_1$, and then incrementally update it to $\mathcal{M}^{\mathcal{T}_2}(\mathbf{x}) = (\mathcal{F}_{\boldsymbol{\phi}}^{\mathcal{T}_2} \circ \mathcal{F}_{\boldsymbol{\theta}}^{\mathcal{T}_2})(\mathbf{x})$, where



(a) Performance of Task 1 after learning Task 1, Task 2 and after adaptation using ELI.

(b) Visualizing Task 1 latents after learning Task 1.

(c) Visualizing Task 1 latents after learning Task 2.

(d) Visualizing Task 1 latents after aligning using ELI.

Figure 3. A key hypothesis that we base our methodology is that while learning a new task, the latent representations will get disturbed, which will in-turn cause catastrophic forgetting of the previous task, and that an energy manifold can be used to align these latents, such that it alleviates forgetting. Here, we illustrate a proof-of-concept that our hypothesis is indeed true. We consider a two task experiment on MNIST; $\mathcal{T}_1 = \{\tau_0, \tau_1, \tau_2, \tau_3, \tau_4\}, \mathcal{T}_2 = \{\tau_5, \tau_6, \tau_7, \tau_8, \tau_9\}$. After learning the second task, the accuracy on $\mathcal{T}_1$ test set drops to 20.88%, while experimenting with a 32 dimensional latent space. The latent aligner in ELI provides 62.56% improvement in test accuracy to 83.44%. The visualization of a 512 dimensional latent space after learning $\mathcal{T}_2$ in sub-figure (c), indeed shows cluttering due to representational shift. ELI is able to align the latents as shown in sub-figure (d), which alleviates the drop in accuracy from 89.14% to 99.04%.

$\mathbf{x} \in \mathcal{T}_2$. When evaluating the Task 1 classification accuracy using $(\mathcal{F}_{\boldsymbol{\phi}}^{\mathcal{T}_1} \circ \mathcal{F}_{\boldsymbol{\theta}}^{\mathcal{T}_2})(\mathbf{x})$, where $\mathbf{x} \in \mathcal{T}_1^{test}$, we see catastrophic forgetting in action. There is a significant drop in performance from 99.2% to 20.9%, when we use a 32 dimensional latent space. Let $\mathcal{F}_{\psi}^{\text{ELI}}$ represent our proposed latent aligner. While re-evaluating the classification accuracy using $(\mathcal{F}_{\boldsymbol{\phi}}^{\mathcal{T}_1} \circ \mathcal{F}_{\psi}^{\text{ELI}} \circ \mathcal{F}_{\boldsymbol{\theta}}^{\mathcal{T}_2})(\mathbf{x})$, where $\mathbf{x} \in \mathcal{T}_1^{test}$, we see an improvement of 62.6% to 83.4%. We also try increasing the latent space dimension to 512. Consistent to our earlier observation, we observe a drop in accuracy from 99.54% to 89.14%. ELI helps to improve it to 99.04%. The absolute drop in performance due to forgetting is lower than the 32 dimensional latent space because of the larger capacity of the model. The visualization of latent space in sub-figure (c) also suggests more cluttering. Sub-figure (d) explicitly reinforces the utility of ELI to realign the latents. Specifically, note how `Class 3` latents which were intermingled with `Class 2` latents are now nicely moved around in the latent space by ELI. These results strongly motivate the utility of our method. By making $\mathcal{F}_{\boldsymbol{\theta}}^{\mathcal{T}_2}$ more stronger using mainstream incremental learning methodologies, we would improve the performance further. We illustrate this on harder datasets for class-incremental learning and incremental object detection setting in Sec. 4.1 and Sec. 4.2 respectively.

4. Experiments and Results

We conduct extensive experiments with incremental classifiers and object detectors to evaluate ELI. To the best of our knowledge, ours is the first methodology, which works across both these settings *without* any modification.

Protocols: In both problem domains, we study class-incremental setting where a group of classes constitutes an incremental task. For class-incremental learning of classifiers, we experiment with two prominent protocols that exist in the literature: **a)** train with half the total number of classes as the first task [20, 31], and equal number of classes per task thereafter, **b)** ensure that each task (including the first) has equal number of classes [7, 24, 38, 41]. The former tests extreme class incremental learning setting, where in the 25 task setting we incrementally add only two classes at each stage for a dataset with 100 classes. It has the advantage of learning a strong initial classifier as it has access to half of the dataset in Task 1. The later setting has a uniform class distribution across tasks. Both these settings test different plausible dynamics of an incremental classifier. For incremental object detection, similar to existing works [22, 37, 46], we follow a two task setting where the second task contains 10, 5 or a single incremental class.

Datasets and Evaluation Metrics: Following existing works [7, 20, 22, 31, 41, 46] we use incremental versions of CIFAR-100 [25], ImageNet subset [41], ImageNet 1k [9] and Pascal VOC [12] datasets. CIFAR-100 [25] contains 50k training images, corresponding to 100 classes, each with spatial dimensions of 32×32. ImageNet-subset [41] contains 100 randomly selected classes from ImageNet datasets. We also experiment with the full ImageNet 2012 dataset [9] which contains 1000 classes. In contrast to CIFAR-100, there are over 1300 images per class with 224×224 size in both ImageNet-subset and ImageNet-1k. Pascal VOC 2007 [12] contains 9963 images, where each object instance is annotated with its class label and location in an image. Instances from 20 classes are annotated in Pascal VOC. Average accuracy across tasks [31, 41] and mean average precision (mAP) [12] is used as the evaluation metric for incremental classification and detection, respectively.

Implementation Details: Following the standard practice [31, 41], we use ResNet-18 [16] for CIFAR-100 experiments and ResNet-32 [16] for ImageNet experiments. We use a batch size of 128 and train for 160 epochs. We start with an initial learning rate of 0.1, which is decayed by 0.1 after 80^{th} and 120^{th} epochs. The EBM is a three layer neural network with 64 neurons in the first two layers and sin-

Table 1. The table shows class-incremental learning results when our latent aligner ELI is added to three prominent and top-performing incremental approaches [20, 31, 41]. ELI is able to provide additional latent space regularization to these methods, consistently improving them across all the settings. The green subscript highlights the relative improvement. Refer to Sec. 4.1 for detailed analysis.

Settings →		Half of all the classes is used to learn the first task						Same number of classes for each task					
Datasets →		CIFAR-100			ImageNet subset			CIFAR-100			ImageNet subset		
Methods	Venue	5 Tasks	10 Tasks	25 Tasks	5 Tasks	10 Tasks	25 Tasks	5 Tasks	10 Tasks	20 Tasks	5 Tasks	10 Tasks	20 Tasks
iCaRL [41]	CVPR 17	56.97	53.28	50.98	58.24	51.6	49.02	61.59	60.05	57.81	71.46	65.25	60.21
iCaRL + ELI		63.68 + 6.71	58.92 + 5.64	54.00 + 3.02	68.94 + 10.73	61.48 + 9.88	56.11 + 7.08	**70.13** + 8.54	**67.81** + 7.75	**63.06** + 5.25	**78.51** + 7.04	**71.66** + 6.41	**66.77** + 6.56
LUCIR [20]	CVPR 19	64.37	62.57	59.91	71.38	68.99	64.65	62.01	58.95	54.2	74.22	67.97	62.2
LUCIR + ELI		66.06 + 1.69	63.50 + 0.93	60.30 + 0.39	**74.58** + 3.21	71.62 + 2.61	66.35 + 1.71	64.55 + 2.49	59.51 + 0.56	54.98 + 0.78	75.38 + 1.16	70.28 + 2.31	65.51 + 3.31
AANet [31]	CVPR 21	67.53	66.25	64.28	70.84	70.3	69.07	63.89	60.94	56.88	65.86	54.13	44.96
AANet + ELI		**68.78** + 1.25	**66.62** + 0.37	**64.72** + 0.44	73.54 + 2.73	**71.82** + 1.52	**70.32** + 1.25	66.36 + 2.47	61.72 + 0.78	57.65 + 0.77	67.43 + 1.57	55.47 + 1.34	46.93 + 1.97

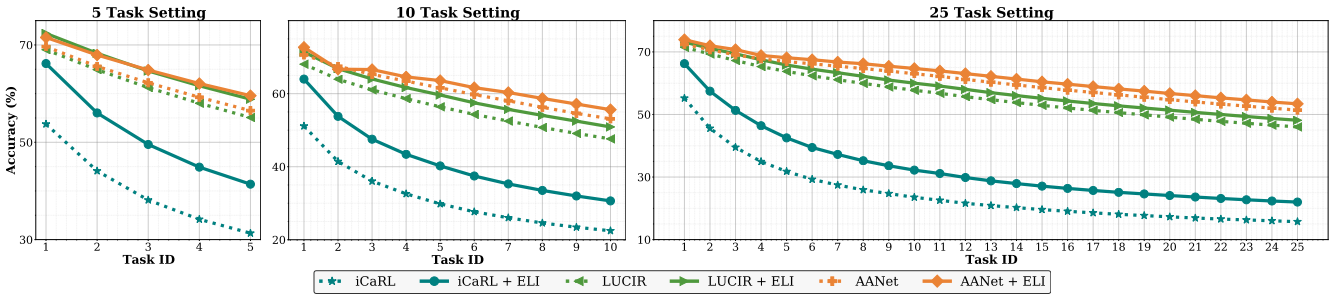


Figure 4. Here, we plot the average accuracy after learning each incremental task on ImageNet 1k dataset. ELI is able to consistently improve iCaRL [41], LUCIR [20] and AANet [31] on 5 task, 10 task and 25 task setting. On average, we see 8.17%, 3.05% and 2.53% improvement to the three base methods. *(Best viewed in color)*

gle neuron in the last layer. The features that are passed on to the final softmax classifier of the base network, are used for learning the EBM. It is trained for 1500 iterations with mini-batches of size 128. The learning rate is set to 0.0001. We use 30 langevin iterations to sample from the EBM. We found that keeping an exponential moving average of the EBM model was effective. The implementations of the three prominent class-incremental methodologies (iCaRL [41], LUCIR [20] and AANet [31]) follows the official code from AANet [31] authors, released under an MIT license. They use an exemplar store of 20 images per class. Note that our latent aligner does not use exemplars. The iCaRL inference is modified to use fully-connected layers following Castro *et al*. [7]. All results are mean of three runs. We use an incremental version of Faster R-CNN [42] for object detection experiments, following iOD [22]. The 2048 dimensional penultimate feature vector from the RoI Head is used to learning the EBM.

4.1. Incremental Classification Results

We augment three popular class-incremental learning methods: iCaRL [41], LUCIR [20] and AANet [31] with our proposed latent aligner. Table 1 showcases the results on CIFAR-100 [25] and ImageNet subset [41] datasets. As explained earlier, we conduct experiments on the setting where half of the classes are learned in the first task, and when all tasks has equal number of classes. In the former, we group 10, 5 and 2 classes each to create 5, 10 and 25 learning tasks respectively, after training the model on 50 initial classes. In the second setting, we group 20, 10 and 5 classes each to create 5, 10 and 20 incremental tasks. We see consistent improvement across all these settings when we add ELI to the corresponding base methodology. In both the settings, the improvement is more pronounced on harder datasets. LUCIR [20] and AANet [31] use an explicit latent space regularizer in their methodology. ELI is able to improve them further. Simpler methods like iCaRL [41] benefit more from the implicit regularization that ELI offers (this aspect is explored further in Sec. 5.1). In Fig. 4, we plot the average accuracy after learning each task in 5 task, 10 task and 25 task settings on ImageNet 1k. We see a similar trend, but with larger improvements on this harder dataset. When added to iCaRL [41], LUCIR [20] and AANet [31], ELI provides 8.17%, 3.05% and 2.53% improvement on average in ImageNet 1k experiments, respectively.

When we consider adding same number of classes in each incremental task, simple logit distillation provided by iCaRL [41], along with our proposed latent aligner outperforms complicated methods by a significant margin. This is because the feature learning that happens with half of the classes in the first task, is a major prerequisite for good performance of approaches like LUCIR [20] and AANet [31].

Table 2. Incremental Object Detection is evaluated in a two task setting with Pascal VOC 2007 dataset [12]. We consider adding 10, 5 and one class (highlighted in color) to a detector trained on the rest of the classes. When added to the state-of-the-art incremental Object Detector iOD [22], ELI provide a competitive improvement of 5.4%, 7% and 3% mAP in $10+10$, $15+5$ and $19+1$ settings respectively.

10 + 10 Setting	aero	cycle	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	bike	person	plant	sheep	sofa	train	tv	mAP
All 20	79.4	83.3	73.2	59.4	62.6	81.7	86.6	83	56.4	81.6	71.9	83	85.4	81.5	82.7	49.4	74.4	75.1	79.6	73.6	75.2
First 10	78.6	78.6	72	54.5	63.9	81.5	87	78.2	55.3	84.4	-	-	-	-	-	-	-	-	-	-	73.4
Std Training	35.7	9.1	16.6	7.3	9.1	18.2	9.1	26.4	9.1	6.1	57.6	57.1	72.6	67.5	73.9	33.5	53.4	61.1	66.5	57	37.3
Shmelkov *et al.* [46]	69.9	70.4	69.4	54.3	48	68.7	78.9	68.4	45.5	58.1	59.7	72.7	73.5	73.2	66.3	29.5	63.4	61.6	69.3	62.2	63.1
Faster ILOD [37]	72.8	75.7	71.2	60.5	61.7	70.4	83.3	76.6	53.1	72.3	36.7	70.9	66.8	67.6	66.1	24.7	63.1	48.1	57.1	43.6	62.2
ORE [21]	63.5	70.9	58.9	42.9	34.1	76.2	80.7	76.3	34.1	66.1	56.1	70.4	80.2	72.3	81.8	42.7	71.6	68.1	77	67.7	64.6
iOD [22]	76	74.6	67.5	55.9	57.6	75.1	85.4	77	43.7	70.8	60.1	66.4	76	72.6	74.6	39.7	64	60.2	68.5	60.5	66.3
iOD + ELI	78.5	81.6	73.8	65.5	63.2	80.2	87.7	82.5	52.4	81.2	55.5	73.1	80.5	76.5	80.4	42.2	68.8	66	72.6	70.8	**71.7**
15 + 5 Setting	aero	cycle	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	bike	person	plant	sheep	sofa	train	tv	mAP
All 20	79.4	83.3	73.2	59.4	62.6	81.7	86.6	83	56.4	81.6	71.9	83	85.4	81.5	82.7	49.4	74.4	75.1	79.6	73.6	75.2
First 15	78.1	82.6	74.2	61.8	63.9	80.4	87	81.5	57.7	80.4	73.1	80.8	85.8	81.6	83.9	-	-	-	-	-	53.2
Std Training	12.7	0.6	9.1	9.1	3	0	8.5	9.1	0	3	9.1	0	3.3	2.3	9.1	37.6	51.2	57.8	51.5	59.8	16.8
Shmelkov *et al.* [46]	70.5	79.2	68.8	59.1	53.2	75.4	79.4	78.8	46.6	59.4	59	75.8	71.8	78.6	69.6	33.7	61.5	63.1	71.7	62.2	65.9
Faster ILOD [37]	66.5	78.1	71.8	54.6	61.4	68.4	82.6	82.7	52.1	74.3	63.1	78.6	80.5	78.4	80.4	36.7	61.7	59.3	67.9	59.1	67.9
ORE [21]	75.4	81	67.1	51.9	55.7	77.2	85.6	81.7	46.1	76.2	55.4	76.7	86.2	78.5	82.1	32.8	63.6	54.7	77.7	64.6	68.5
iOD [22]	78.4	79.7	66.9	54.8	56.2	77.7	84.6	79.1	47.7	75	61.8	74.7	81.6	77.5	80.2	37.8	58	54.6	73	56.1	67.8
iOD + ELI	80.1	85.8	73.6	68.8	66.3	85.2	87.5	84.1	59.9	81.2	74.6	83.7	85.3	77.9	80.3	45.2	63.4	66.2	77.6	69.5	**74.8**
19 + 1 Setting	aero	cycle	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	bike	person	plant	sheep	sofa	train	tv	mAP
All 20	79.4	83.3	73.2	59.4	62.6	81.7	86.6	83	56.4	81.6	71.9	83	85.4	81.5	82.7	49.4	74.4	75.1	79.6	73.6	75.2
First 19	76.3	77.3	68.4	55.4	59.7	81.4	85.3	80.3	47.8	78.1	65.7	77.5	83.5	76.2	77.2	46.6	71.4	65.8	76.5	-	67.5
Std Training	16.6	9.1	9.1	9.1	9.1	8.3	35.3	9.1	0	22.3	9.1	9.1	9.1	13.7	9.1	9.1	23.1	9.1	15.4	50.7	14.3
Shmelkov *et al.* [46]	69.4	79.3	69.5	57.4	45.4	78.4	79.1	80.5	45.7	76.3	64.8	77.2	80.8	77.5	70.1	42.3	67.5	64.4	76.7	62.7	68.3
Faster ILOD [37]	64.2	74.7	73.2	55.5	53.7	70.8	82.9	82.6	51.6	79.7	58.7	78.8	81.8	75.3	77.4	43.1	73.8	61.7	69.8	61.1	68.6
ORE [21]	67.3	76.8	60	48.4	58.8	81.1	86.5	75.8	41.5	79.6	54.6	72.8	85.9	81.7	82.4	44.8	75.8	68.2	75.7	60.1	68.9
iOD [22]	78.2	77.5	69.4	55	56	78.4	84.2	79.2	46.6	79	63.2	78.5	82.7	79.1	79.9	44.1	73.2	66.3	76.4	57.6	70.2
iOD + ELI	84.7	79.2	73.7	60.1	61.8	82.8	85.4	82.9	51.3	82.7	64.5	82.3	82.9	75.9	78.7	50.7	73.9	74.7	76.7	59.2	**73.2**

Figure 5. In these qualitative results of incremental Object Detection, instances of `plant`, `sheep`, `sofa`, `train` and `tvmonitor` were introduced to a detector trained on the rest. We detect instances of old and new classes alike. More results are in supplementary materials.

4.2. Incremental Object Detection Results

Following the standard evaluation protocol [22, 46] for incremental object detection, we group classes from Pascal VOC 2007 [12] into two tasks. Three different task combinations are considered here. We initially learn 10, 15 or 19 classes, and then introduce 10, 5 or one class as the second task, respectively. Table 2 shows the results of this experiment. The first two rows in each section give the upper-bound and the accuracy after learning the first task. The 'Std Training' row shows how the performance on previous classes deteriorate when simply finetuning the model on the new class instances. The next three rows titled Shmelkov *et al.* [46], Faster ILOD [37] and ORE [21] show how existing methods help to address catastrophic forgetting. We add ELI to iOD [22], the current state-of-the-art method, to improve its mAP by 5.4%, 7% and 3% while adding 10, 5 and one class respectively, to a detector trained on the rest. This improvement can be attributed to the effectiveness of ELI in aligning the latent representations to reduce forgetting. These results also demonstrate that ELI is an effective plug-and-play method to reduce forgetting, across classification and detection tasks. Fig. 5 shows our qualitative results.

5. Discussions and Analysis

5.1 ELI as an Implicit Regularizer: To showcase the effectiveness of the implicit regularization that ELI offers, we remove the explicit latent regularization term (referred to as ER in Fig. 6) from our top performing method

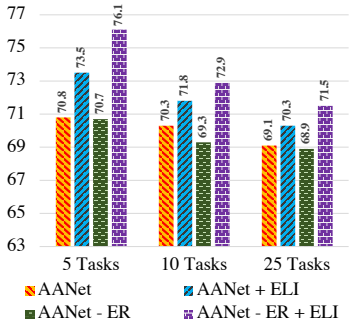


Figure 6. ELI as an Implicit Regularizer on ImageNet subset.

Table 3. We vary the number of Langevin steps L_{steps}, required to sample from the EBM. The latents get aligned even within a few steps.

# of steps	5 Tasks	10 Tasks	25 Tasks
5	63.30	58.61	52.81
10	63.66	58.85	53.49
20	63.63	58.90	53.76
30	63.68	58.92	54.00
60	63.73	59.01	54.07
90	63.79	58.97	54.04

Table 4. We change the number of iterations for training the EBM in Algo. 1. The EBM converges within 1k iterations, with moderate improvement thereafter.

# of iterations	5 Tasks	10 Tasks	25 Tasks
10	56.90	53.85	49.02
100	60.53	57.08	50.41
1000	63.60	58.88	53.66
1500	63.68	58.92	54.00
2000	63.80	58.97	54.03
3000	63.67	58.85	54.06

Table 5. We vary the architecture of the EBM here. i and o refers to input and output layer, while the values in-between represent the number of neurons in each layer.

Architecture	5 Tasks	10 Tasks	25 Tasks
i - o	60.97	57.52	53.92
i - 64 - o	63.72	59.02	54.59
i - 64 - 64 - o	63.68	58.92	54.00
i - 64 - 64 - 64 - o	63.71	58.9	54.44
i - 256 - 256 - o	63.53	58.68	54.16
i - 512 - 512 - o	63.66	58.66	54.00

AANet [31] on ImageNet subset [41] experiments. There is a consistent drop in accuracy when ER is removed from the base method (**green** bars). ELI is able to improve the performance of such a model by 5.41%, 3.58% and 2.57% on 5, 10 and 25 task experiments respectively (**violet** bars). We note that the gain is more significant when we compare with adding ELI to AANet with explicit regularization, corroborating the effectiveness of our implicit regularizer.

5.2 Aligning the Final Layer Logits: ELI aligns the latent representations from the feature extractor $\mathbf{z} = \mathcal{F}_{\boldsymbol{\theta}}^{\mathcal{T}_t}(\mathbf{x})$. An alternative would be to align the final logits $\mathcal{F}_{\boldsymbol{\phi}}^{\mathcal{T}_t}(\mathcal{F}_{\boldsymbol{\theta}}^{\mathcal{T}_t}(\mathbf{x}))$. We re-evaluate incremental CIFAR-100 experiments in this setting. We find that latent space alignment is more effective than aligning the logit space (referred to as '+ Logit Aligner' in Tab. 6). This is because the logits are specific to the end task while the latent representations model generalizable features across tasks.

5.3 Aligning across Different-sized Latent Spaces: ELI can align latent representations of varied dimensions. Our toy experiment on MNIST uses 32 and 512 dimensional latent space, while CIFAR-100 experiments use a 64 dimensional space. ImageNet and Pascal VOC experiments uses a latent space of 512 and 2048 dimensions each.

5.4 Sensitivity to Hyper-parameters: We alter parameters that can affect the ELI performance in Tab. 3, 4 and 5. The experiments are on CIFAR-100 in 'iCaRL + ELI' setting. The highlighted rows represent the default configuration.

Number of Langevin Steps: In Tab. 3, we experiment with changing the number of Langevin steps L_{steps} required to sample from EBM in Algo. 2. ELI is able to align latents with very few number of steps as the energy manifold is adept in guiding the alignment of latent representations.

Number of Iterations Required: While training the EBM using Algo. 1, we change the number of iterations required, and report the accuracy in Tab. 4. At around 1000 iterations, the EBM converges. Increasing the number of iterations further, does not lead to significant improvement.

Architecture: We experiment with EBM models of different capacities in Tab. 5. We find that using a smaller architecture or a significantly larger architecture does not help. We see this as a desirable characteristic since we learn the energy manifold of the latent space and not the data space.

5.5 Compute and Memory: We record the compute, memory and time requirements of ELI for CIFAR-100. We use a single Nvidia Tesla K80 GPU for these metrics. The EBM, which is a two layer network with 64 neurons each, has 8.385K parameters and takes 1.057M flops when trying to learn 64 dimensional latent features. It takes 0.039 ± 0.003 secs to sample from this EBM when aligning 64 dimensional latents. We use 30 Langevin iterations for sampling. Mini-batch size is 128 for both the experiments.

Table 6. Latent representations alignment is more effective than aligning logits. Subscripts show change in accuracy from baseline.

Method	5 Tasks	10 Tasks	25 Tasks
iCaRL [41]	56.97	53.28	50.98
iCaRL [41] + Logit Aligner	57.97 $_{+ 1.00}$	54.42 $_{+ 1.14}$	51.49 $_{+ 0.51}$
iCaRL [41] + ELI	63.68 $_{+ 6.71}$	58.92 $_{+ 5.64}$	54.00 $_{+ 3.02}$
LUCIR [20]	64.37	62.57	59.91
LUCIR [20] + Logit Aligner	62.50 $_{- 1.87}$	61.67 $_{- 0.9}$	59.22 $_{- 0.69}$
LUCIR [20] + ELI	66.06 $_{+ 1.69}$	63.50 $_{+ 0.93}$	60.30 $_{+ 0.39}$
AANet [31]	67.53	66.25	64.28
AANet [31] + Logit Aligner	66.16 $_{- 1.37}$	65.29 $_{- 0.96}$	63.81 $_{- 0.47}$
AANet [31] + ELI	68.78 $_{+ 1.25}$	66.62 $_{+ 0.37}$	64.72 $_{+ 0.44}$

6. Conclusion

We demonstrate the use of energy-based models (EBMs) as a promising solution for incremental learning, by extending their natural mechanism to deal with representational shift. This is achieved by modeling the likelihoods in the latent feature space, measuring the distributional shifts experienced across learning tasks and in-turn realigning them to optimize learning across all the tasks. Our proposed approach ELI, is complementary to existing methods and can be used as an add-on module without modifying their base pipeline. ELI offers consistent improvement to three prominent class-incremental classification methodologies when evaluated across multiple settings. Further, on the harder incremental object detection task, our methodology provides significant improvement over state-of-the-art.

Acknowledgements

We thank Yaoyao Liu for his prompt clarifications on AANET [31] code. KJJ thanks TCS Research for their PhD fellowship. VNB thanks DST, Govt of India, for partially supporting this work through the IMPRINT and ICPS programs.

References

[1] Davide Abati, Jakub Tomczak, Tijmen Blankevoort, Simone Calderara, Rita Cucchiara, and Babak Ehteshami Bejnordi. Conditional channel gated networks for task-aware continual learning. In *CVPR*, pages 3931–3940, 2020. 2

[2] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 139–154, 2018. 1, 3

[3] Michael Arbel, Liang Zhou, and Arthur Gretton. Generalized energy based models. In *ICLR*, 2021. 2

[4] Anton Bakhtin, Yuntian Deng, Sam Gross, Myle Ott, Marc'Aurelio Ranzato, and Arthur Szlam. Residual energy-based models for text. *Journal of Machine Learning Research*, 22(40):1–41, 2021. 2

[5] David Belanger and Andrew McCallum. Structured prediction energy networks. pages 983–992. PMLR, 2016. 2

[6] Eden Belouadah and Adrian Popescu. Il2m: Class incremental learning with dual memory. In *ICCV*, pages 583–592, 2019. 2

[7] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *ECCV*, pages 233–248, 2018. 1, 2, 5, 6

[8] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. In *ICLR*, 2019. 1, 3

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2, 5

[10] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *ICCV*, pages 86–102. Springer, 2020. 2

[11] Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *NeurIPS*, volume 32, 2019. 2, 3

[12] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 2, 5, 7

[13] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999. 1

[14] Will Grathwohl, Kuan-Chieh Wang, Joern-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *ICLR*, 2019. 2

[15] Akshita Gupta, Sanath Narayan, KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Mubarak Shah. Ow-detr: Open-world detection transformer. *CVPR*, 2021. 2

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016. 5

[17] Mitch Hill, Jonathan Craig Mitchell, and Song-Chun Zhu. Stochastic security: Adversarial defense using long-run dynamics of energy-based models. In *ICLR*, 2021. 2

[18] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8):2, 2012. 4

[19] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015. 2

[20] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *CVPR*, pages 831–839, 2019. 2, 5, 6, 8

[21] KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Towards open world object detection. In *CVPR*, pages 5830–5840, 2021. 1, 2, 7

[22] KJ Joseph, Jathushan Rajasegaran, Salman Khan, Fahad Khan, and Vineeth N Balasubramanian. Incremental object detection via meta-learning. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, Nov 2021. 2, 5, 6, 7

[23] 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. 1, 3

[24] Joseph KJ and Vineeth Nallure Balasubramanian. Meta-consolidation for continual learning. *NeurIPS*, 33, 2020. 1, 2, 5

[25] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Citeseer*, 2009. 2, 5, 6

[26] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006. 2, 3

[27] Shuang Li, Yilun Du, Gido M van de Ven, and Igor Mordatch. Energy-based models for continual learning. *arXiv preprint arXiv:2011.12216*, 2020. 2

[28] Zhizhong Li and Derek Hoiem. Learning without forgetting. *PAMI*, 40(12):2935–2947, 2017. 2

[29] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2018. 1

[30] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *NeurIPS*, 2020. 2

[31] Yaoyao Liu, Bernt Schiele, and Qianru Sun. Adaptive aggregation networks for class-incremental learning. In *CVPR*, pages 2544–2553, 2021. 2, 5, 6, 8

[32] Yaoyao Liu, Yuting Su, An-An Liu, Bernt Schiele, and Qianru Sun. Mnemonics training: Multi-class incremental learning without forgetting. In *CVPR*, pages 12245–12254, 2020. 2

[33] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *NeurIPS*, pages 6467–6476, 2017. 1, 3

[34] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *CVPR*, pages 7765–7773, 2018. 1, 3

[35] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989. 1

[36] Radford M Neal et al. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2, 2011. 4

[37] Can Peng, Kun Zhao, and Brian C Lovell. Faster ilod: Incremental learning for object detectors based on faster rcnn. *Pattern Recognition Letters*, 140:109–115, 2020. 5, 7

[38] Jathushan Rajasegaran, Munawar Hayat, Salman Khan, Fahad Shahbaz Khan, and Ling Shao. Random path selection for incremental learning. *NeurIPS*, 2019. 2, 5

[39] Jathushan Rajasegaran, Munawar Hayat, Salman Khan, Fahad Shahbaz Khan, Ling Shao, and Ming-Hsuan Yang. An adaptive random path selection approach for incremental learning. *arXiv preprint arXiv:1906.01120*, 2019. 2

[40] Jathushan Rajasegaran, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Mubarak Shah. itaml: An incremental task-agnostic meta-learning approach. In *CVPR*, pages 13588–13597, 2020. 1, 2

[41] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, pages 2001–2010, 2017. 1, 2, 3, 5, 6, 8

[42] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *NeurIPS*, 28:91–99, 2015. 6

[43] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, , and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *ICLR*, 2019. 2

[44] 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. 1, 2, 3

[45] Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. pages 4548–4557, 2018. 1, 3

[46] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. Incremental learning of object detectors without catastrophic forgetting. In *Proceedings of the IEEE international conference on computer vision*, pages 3400–3409, 2017. 2, 5, 7

[47] Francesco Tonin, Arun Pandey, Panagiotis Patrinos, and Johan A. K. Suykens. Unsupervised energy-based out-of-distribution detection using stiefel-restricted kernel machine. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2021. 2

[48] Lifu Tu and Kevin Gimpel. Learning approximate inference networks for structured prediction. In *ICLR*, 2018. 2

[49] Yezhen Wang, Bo Li, Tong Che, Kaiyang Zhou, Ziwei Liu, and Dongsheng Li. Energy-based open-world uncertainty modeling for confidence calibration. In *ICCV*, pages 9302–9311, 2021. 2

[50] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. pages 681–688, 2011. 4

[51] Oliver Woodford. Notes on contrastive divergence. *Department of Engineering Science, University of Oxford, Tech. Rep*, 2006. 4

[52] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *CVPR*, pages 374–382, 2019. 2

[53] Zhisheng Xiao, Karsten Kreis, Jan Kautz, and Arash Vahdat. Vaebm: A symbiosis between variational autoencoders and energy-based models. In *ICLR*, 2021. 2

[54] Jianwen Xie, Yang Lu, Song-Chun Zhu, and Yingnian Wu. A theory of generative convnet. pages 2635–2644. PMLR, 2016. 2

[55] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. pages 3987–3995, 2017. 1

[56] Yang Zhao and Changyou Chen. Unpaired image-to-image translation via latent energy transport. In *CVPR*, pages 16418–16427, 2021. 2

[57] Yang Zhao, Jianwen Xie, and Ping Li. Learning energy-based generative models via coarse-to-fine expanding and sampling. In *ICLR*, 2020. 2