VariGrow: Variational Architecture Growing for Task-Agnostic Continual Learning based on Bayesian Novelty

Randy Ardywibowo [1] **Zepeng Huo** [1] **Zhangyang Wang** [2] **Bobak Mortazavi** [1] **Shuai Huang** [3] **Xiaoning Qian** [1]

Abstract

Continual Learning (CL) is the problem of sequentially learning predictive models with varying data that may originate from different contexts. Many existing CL methods assume that the data stream is divided into a sequence of contexts, termed as **tasks**, with explicitly given transition boundaries. Unfortunately, many real-world CL scenarios have neither explicit task information nor context boundaries, motivating the study of task-agnostic CL. This paper proposes a variational architecture growing framework dubbed **VariGrow**. By interpreting dynamically growing neural networks as a Bayesian approximation, and defining flexible implicit variational distributions, VariGrow detects if a new task is arriving through an energy-based *novelty score*. If the novelty score is high and the sample is "detected" as a new task, VariGrow will grow a new expert module to be responsible for it. Otherwise, the sample will be assigned to one of the existing experts who is the most "familiar" with it (i.e., one with the lowest novelty score) to preserve all the acquired knowledge. We have tested VariGrow on several CIFAR and ImageNet-based benchmarks for the **strictly task-agnostic** CL setting without any task information during training or testing, which demonstrates its consistently superior or competitive performance. More interesting, VariGrow achieves comparable performance with task-aware CL methods.

1. Introduction

In conventional machine learning, data points are assumed to be identically and independently distributed (*iid*) and all available at once. In contrast, the regime of continual learning (CL) presents the new challenge of incrementally accumulating knowledge from past experiences, mimicking human's ability to learn with non-*iid* data streams in widely varying contexts. CL sequentially learns novel concepts to achieve reliable predictions without *catastrophically forgetting* previously learned knowledge. It can be applied to many real-world applications, such as robotics (Thrun & Mitchell, 1995), computer vision (Li et al., 2017), autonomous driving (Pierre, 2018), and healthcare monitoring (Ardywibowo et al., 2019; 2018; Ardywibowo, 2017; Jiang et al., 2019). To this end, many CL methods have been developed in attempting to solve the *stability-plasticity dilemma* (Aljundi et al., 2017; 2018; Lopez-Paz & Ranzato, 2017; Kirkpatrick et al., 2017; Rusu et al., 2016; Shin et al., 2017; Yoon et al., 2018; Yan et al., 2021b; Liu et al., 2021; Rajasegaran et al., 2019).

Many existing CL methods assume that the data stream is explicitly divided into a sequence of transiting contexts, termed as **tasks**, with task information given at both *training* and *testing* time. In real-world scenarios, however, there is no clear transition boundary between different contexts or tasks, limiting the application of these CL methods in practice (Lee et al., 2020). With this in mind, **task-agnostic CL** performs continual learning without requiring task IDs and their transitions. This new setting is challenging, dubbed as the *single-headed* setting, where existing task-agnostic CL methods have significantly lower performance compared to their task-aware counterparts (Rao et al., 2019; Aljundi, 2019; Aljundi et al., 2019; Zeno et al., 2018; He & Jaeger, 2018). In this paper, we focus on task-agnostic CL for classification problems.

Existing CL methods can be broadly categorized into 1) regularization-based, 2) memory-based, and 3) expansion-based (Parisi et al., 2019). While regularization- and memory-based methods focus on retaining the knowledge learned from the old tasks, expansion-based methods lean towards better absorbing new knowledge and circumvent the *capability saturation* (Sodhani et al., 2020). To the best

[1] Texas A&M University, College Station, Texas, USA [2] University of Texas, Austin, Texas, USA [3] University of Washington, Seattle, Washington, USA. Correspondence to: Randy Ardywibowo <randyardywibowo@tamu.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

of our knowledge, though, methods that tackle both catastrophic forgetting and capability saturation, under the task-agnostic CL paradigm, are lacking (Kaushik et al., 2021). Bayesian inference offers a promising way to reconcile this problem, with old data points naturally being summarized by a posterior distribution that can be sequentially updated. The inherent uncertainty quantification capability enables effective task-agnostic CL without needing task information explicitly (Barber, 2012). In particular, Bayesian nonparametrics offer a natural solution to the *stability-plasticity* dilemma by principally increasing model complexity as novel data arrives. However, the posterior distribution becomes intractable with large and complex datasets, and existing sequential variational approximations are not flexible enough to capture the complexity of these datasets (Blei & Jordan, 2006; Lin, 2013; Lee et al., 2020; Kessler et al., 2019). Promisingly, implicit variational inference enables flexible modeling of the posterior (Yin & Zhou, 2018; Titsias & Ruiz, 2019; Molchanov et al., 2019). However, their application to dynamically growing architectures for task-agnostic CL has not been previously explored.

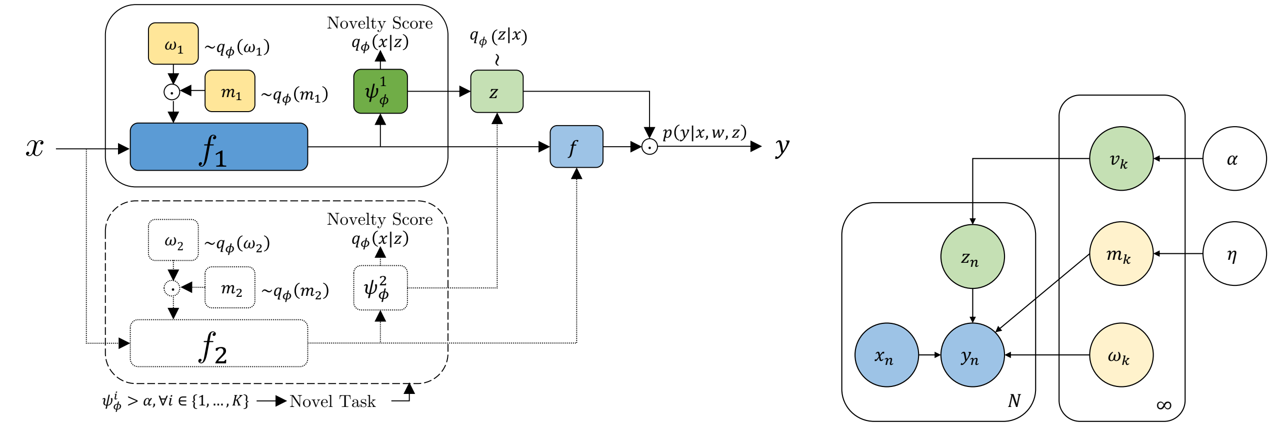
In this paper, we propose VariGrow, a **Vari**ational architecture **Grow**ing framework for task-agnostic continual learning. To accomplish this, we first formulate model or network growing in terms of Bayesian nonparametric distributions that define an infinite mixture of expert distributions, which can be considered as having an expansion-based backbone. This consists of an expert distribution for each mixture component and a mixing distribution selecting from which expert the data originate. We then approximate these distributions using flexible implicit variational distributions, allowing us to more accurately capture the posterior at each incremental step. Specifically, we design a mixing distribution using energy-based novelty scores to determine the mixture component to which each data point belongs (LeCun et al., 2006; Liu et al., 2020). This allows to dynamically decide whether to grow a new mixture component for novel instances, or to assign it to an existing one. Meanwhile, each component is handled by an expert distribution defined implicitly through Bayesian Neural Networks (BNNs) (Yin & Zhou, 2018; Titsias & Ruiz, 2019; Molchanov et al., 2019). By deriving tractable approximations to the Kullback-Leibler (KL) divergence, we optimize the Evidence Lower Bound (ELBO) of our formulation through stochastic gradient-based techniques along with a sparsification trick to ensure expressiveness. We have tested VariGrow on several CIFAR and ImageNet-based benchmarks for the **strict task-agnostic** (without using the 'label trick' (Zeno et al., 2018)) CL setting, which demonstrates its consistently competitive performance to existing task-agnostic CL methods. Interestingly, VariGrow even achieves comparable performances to task-aware counterparts.

2. Related Work

Continual Learning: Continual learning models aim to learn new knowledge without catastrophically forgetting previously learned information. Methods in this domain can be broadly categorized into three classes: 1) *memory-based* methods which store a subset of raw data or build a generative model to generate synthetic data for replay (Rebuffi et al., 2017; Shin et al., 2017; Lopez-Paz & Ranzato, 2017; Riemer et al., 2018), 2) *regularization-based* methods which focus on preserving old information when learning new ones by penalizing drastic changes to a model's parameters (Kirkpatrick et al., 2017; Aljundi et al., 2018; Titsias et al., 2019; Pomponi et al., 2020), and 3) *expansion-based* methods which grow and assign new model components for different tasks, keeping unrelated model parameters fixed. The expansion can be based on neurons (Wortsman et al., 2020), layers (Rusu et al., 2016; Schwarz et al., 2018), or independent networks (Yan et al., 2021b). Most CL methods require the task information during training and/or testing. For example, in the *multi-head* setting, models would only need to predict among the classes in one task, instead of the whole class set (Kaushik et al., 2021).

Task-agnostic continual learning: In many real-world applications, the current task information is usually not given (Lee et al., 2020; Kirichenko et al., 2021). Some methods proposed to tackle the task-agnostic setting, but only during testing (Kaushik et al., 2021; Yan et al., 2021b; Abati et al., 2020; Rajasegaran et al., 2020). There are recently developed methods assuming that task information is not given during training (Lee et al., 2020; Ebrahimi et al., 2020; Zeno et al., 2018) but their performances are much lower compared to their task-aware counterparts. Furthermore, the model training in these methods have various drawbacks. Rajasegaran et al. (2020) assumes that data in one batch comes from a single context (task), and assume that task labels are available during training. The training of UCB (Ebrahimi et al., 2020) is extremely slow due to their modified backpropagation formulation. In Lee et al. (2020), CN-DPM has the least assumptions on the data stream; however, their method requires performing density estimation through generative modeling, which can be intractable and unstable (Grathwohl et al., 2019; Lee et al., 2020; Mescheder et al., 2018; Nalisnick et al., 2018), causing their performance to be significantly lower than task-aware CL methods.

Variational Inference for CL with Anomaly Detection: Our VariGrow is motivated by the energy-based model (EBM), which maps an input to a single, non-probabilistic scalar called *energy* (Liu et al., 2020; LeCun et al., 2006). This energy score has shown to outperform the softmax confidence score for OoD detection (Hendrycks & Gimpel, 2016). Some other works on OoD detection either



(a)

(b)

Figure 1. VariGrow schematic: **(a)**: The dynamically growing construct illustrated for two expert components. The input $\boldsymbol{x}$ is passed into a Bayesian Neural Network f_1 with weights $\boldsymbol{\omega}_1 \sim q_{\boldsymbol{\phi}}(\boldsymbol{\omega}_1)$ multiplied by a binary mask $\boldsymbol{m}_1 \sim q_{\boldsymbol{\phi}}(\boldsymbol{m}_1)$, sparsifying the architecture. The output is used to compute an energy-based novelty score $\psi_{\boldsymbol{\phi}}^1$. As $\psi_{\boldsymbol{\phi}}^1(\boldsymbol{x})$ exceeds a threshold α, VariGrow expands and creates a second mixture component. The novelty scores are then used to construct the mixing distribution and sample $z \sim q_{\boldsymbol{\phi}}(z|x)$, determining which expert component is used to compute $p(y|\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{z})$. Through differentiable reparameterizations and approximations, gradient-based optimization can be performed to learn the variational parameters that optimize the Evidence Lower Bound (ELBO). **(b)**: The graphical model of the nonparametric distribution that we approximate, consisting of mixture assignments z_n for each data point according to prior probability $p(z_n = k) = v_k$, and a mixture distribution where the expert parameters $\boldsymbol{w}_k = \{\boldsymbol{\omega}_k, \boldsymbol{m}_k\}$ are sampled from.

develop deep generative models (Nalisnick et al., 2018), unify probabilistic and non-probabilistic models (Ranzato et al., 2007), or add background classes to enhance OoD detection (Mohseni et al., 2020). Kurle et al. (2019) analyzed non-stationary data using Bayesian neural networks and memory-based online variational Bayes by implementing 'Bayesian forgetting' to selectively forget knowledge not relevant to the current data distribution. Kessler et al. (2019) proposed a hierarchical Indian Buffet process (IBP) to allocate resources when learning new tasks. However, training would still require task information for online inference. Zeno et al. (2018) proposed Bayesian Gradient Descent to train neural networks, claiming that their closed-form update rule better fits task-agnostic training. However, a 'label trick' was used to implicitly infer new tasks from novel labels. Nguyen et al. (2018) proposed VCL, a variational Bayesian interpretation of CL using exemplar data points and a KL divergence penalty to retain previous information. But VCL is a *multi-head* formulation and requires task labels both during training and testing. Kirichenko et al. (2021) proposed using likelihood-based mixture models to handle the multi-modality of the different tasks. However, likelihood-based models fail on complex datasets and often assign higher likelihoods to OoD data (Nalisnick et al., 2018). For this, they resort to use a pretrained model to extract features for more complex datasets such as CIFAR100 (Krizhevsky et al., 2009) and ImageNet (Deng et al., 2009).

3. Methodology

Let $\{\mathcal{D}_t\}_{i=t}^T$ be a stream of datasets with each $\mathcal{D}_t$ having input-output pairs $(\boldsymbol{x}, y)$. Bayesian learning places a prior distribution $p(\boldsymbol{\theta})$ on the model parameters $\boldsymbol{\theta}$. In continual learning (CL), the posterior distribution after observing $t+1$ datasets is obtained using Bayes' rule:

$$p(\boldsymbol{\theta}|\mathcal{D}_{1:t+1}) \propto p(\mathcal{D}_{t+1}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D}_{1:t}). \tag{1}$$

Here, the posterior obtained in the previous step t is treated as a prior for the current step $t+1$. As we observe more novel data points, the complexity of the evolving posterior given our dataset increases. Hence it is important that our model scales accordingly (Hjort et al., 2010).

3.1. VariGrow

To this end, we design a dynamically growing model, VariGrow, parameterized as follows:

$$\begin{aligned} p(\boldsymbol{w}, \boldsymbol{z}|\mathcal{D}_{1:t+1}) &\propto p(\mathcal{D}_{t+1}|\boldsymbol{w}, \boldsymbol{z}_{t+1})p(\boldsymbol{w}, \boldsymbol{z}|\mathcal{D}_{1:t}), \\ p(\boldsymbol{w}, \boldsymbol{z}|\mathcal{D}_{1:t}) &= \prod_{i=1}^{t} p(\boldsymbol{w}_{\boldsymbol{z}_i}|\mathcal{D}_i)p(\boldsymbol{z}|\mathcal{D}_{1:t}). \end{aligned}$$

Here, $\boldsymbol{w}$ denote the parameters of an expert module such as a neural network, while $\boldsymbol{z}$ determines which expert mixture component $p(\boldsymbol{w}_{\boldsymbol{z}})$ to sample $\boldsymbol{w}$ from. This mixing strategy naturally enriches the model representation capacity when needed for continual learning. The schematic is shown in Figure. 1. When training with large and complex datasets, the posterior distribution is intractable and is typically approximated (Blei et al., 2017). It is important that the distributions we use to approximate the posterior via variational inference are flexible and expressive. For CL in particular, one must ensure that these variational distributions can be robustly updated without requiring the access to previously observed datasets (Nguyen et al., 2018). In our CL settings, one would expect to grow more components as we sequentially observe more data from novel tasks. There could be infinitely many expert mixture components, presenting additional challenges in inference (Hjort et al., 2010). To address the mentioned challenges, we define the following variational approximation to the above posterior:

$$q_{\boldsymbol{\phi}}(\boldsymbol{w}, \boldsymbol{z}|\boldsymbol{x}) = \prod_{i=1}^{t} q_{\boldsymbol{\phi}}(\boldsymbol{w}_{\boldsymbol{z}_i}) q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}). \tag{2}$$

To obtain an ideal approximate solution, it is crucial that both $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ and $q_{\boldsymbol{\phi}}(\boldsymbol{w}_{\boldsymbol{z}})$ are expressive and flexible. To this end, we will define these distributions implicitly. Also, note that we make $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ covariate-dependent, allowing us to assign individual data points to any mixture component. To deal with the potentially infinite number of expert modules, we can define $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ through a set of K expert components, while the other mixture components can be defined in relation to these main components. We describe these two distributions in detail in the following sections.

For the moment, let us assume that these two distributions are given. Optimizing the variational parameters $\boldsymbol{\phi}$ corresponds to minimizing the negative ELBO at each CL step t:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\phi}_{t+1}) &= \mathbb{E}_{q_{\boldsymbol{\phi}_{t+1}}(\boldsymbol{w}, \boldsymbol{z}|\boldsymbol{x})}[-\log p(\mathcal{D}_{t+1}|\boldsymbol{w}_{\boldsymbol{z}_{t+1}})] \\ &\quad + \mathrm{KL}(q_{\boldsymbol{\phi}_{t+1}}(\boldsymbol{w}, \boldsymbol{z}|\boldsymbol{x}) \parallel q_{\boldsymbol{\phi}_t}(\boldsymbol{w}, \boldsymbol{z}|\boldsymbol{x})). \end{aligned} \tag{3}$$

The expectation can be approximated using a single sample of $(\boldsymbol{w}, \boldsymbol{z}_{t+1}) \sim q_{\boldsymbol{\phi}_{t+1}}(\boldsymbol{w}, \boldsymbol{z}|\boldsymbol{x})$, and $\log p(\mathcal{D}_{t+1}|\boldsymbol{w}_{K+1}) = N\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{t+1}}[\log p(y|\boldsymbol{x}, \boldsymbol{w}_{K+1})]$ can be approximated using minibatches, with N being the number of data points in $\mathcal{D}_{t+1}$. When a novel task is detected by our variational formulation, $\boldsymbol{z}_{t+1} > K$, and a new expert mixture component $K+1$ is created. So the ELBO becomes

$$\begin{aligned} \mathcal{L}(\boldsymbol{\phi}_{t+1}) &= \mathbb{E}_{q_{\boldsymbol{\phi}_{t+1}}(\boldsymbol{w}, \boldsymbol{z}|\boldsymbol{x})}[-\log p(\mathcal{D}_{t+1}|\boldsymbol{w}_{K+1})] \\ &\quad + \mathrm{KL}(q_{\boldsymbol{\phi}_{t+1}}(\boldsymbol{w}_{K+1}) \parallel p(\boldsymbol{w})) \\ &\quad + \mathrm{KL}(q_{\boldsymbol{\phi}_{t+1}}(\boldsymbol{z}|\boldsymbol{x}) \parallel q_{\boldsymbol{\phi}_t}(\boldsymbol{z}|\boldsymbol{x})). \end{aligned} \tag{4}$$

Here, since $K+1$ indicates a new mixture component $q_{\boldsymbol{\phi}_t}(\boldsymbol{w}_{K+1}) = p(\boldsymbol{w})$, where $p(\boldsymbol{w})$ is the prior distribution on the expert parameters. Meanwhile when we observe data points assigned to an existing mixture component, $\boldsymbol{z}_{t+1} = k \in \{1, \ldots, K\}$, we have:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\phi}_{t+1}) &= \mathbb{E}_{q_{\boldsymbol{\phi}_{t+1}}(\boldsymbol{w}, \boldsymbol{z}|\boldsymbol{x})}[-\log p(\mathcal{D}_{t+1}|\boldsymbol{w}_k)] \\ &\quad + n_k \mathrm{KL}(q_{\boldsymbol{\phi}_{t+1}}(\boldsymbol{w}_k) \parallel q_{\boldsymbol{\phi}_t}(\boldsymbol{w}_k)) \\ &\quad + \mathrm{KL}(q_{\boldsymbol{\phi}_{t+1}}(\boldsymbol{z}|\boldsymbol{x}) \parallel q_{\boldsymbol{\phi}_t}(\boldsymbol{z}|\boldsymbol{x})), \end{aligned} \tag{5}$$

where n_k is the number of data points previously assigned to expert k. Intuitively, as more data points are assigned to expert k, we would expect the expert distribution $q_{\boldsymbol{\phi}}(\boldsymbol{w}_k)$ to approach the true corresponding posterior. Meanwhile, for new expert components, only a prior distribution is given, and the component is free to learn from novel data.

To evaluate and optimize the ELBO above, it is important that we define our variational distributions such that the KL terms defined above are easily computable, plus being flexible and expressive. So we adopt an energy-based mixing construct for the variational distribution $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ and define the expert weight distribution implicitly.

3.2. Energy-based Mixing Distribution

Here, we describe our specification for the expert mixing distribution $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$. By Bayes' rule, we have

$$q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) = \frac{q_{\boldsymbol{\phi}}(\boldsymbol{x}|\boldsymbol{z}) q_{\boldsymbol{\phi}}(\boldsymbol{z})}{\sum_{i=1}^{\infty} q_{\boldsymbol{\phi}}(\boldsymbol{x}|\boldsymbol{z}=i) q_{\boldsymbol{\phi}}(\boldsymbol{z}=i)}. \tag{6}$$

Although one would typically find $q_{\boldsymbol{\phi}}(\boldsymbol{x}|\boldsymbol{z})$ through density estimation, such as an agnostic method CN-DPM (Lee et al., 2020), this involves the difficult optimization process of training generative models which can be intractable and unstable to perform in practice (Grathwohl et al., 2019; Mescheder et al., 2018; Nalisnick et al., 2018). Instead of relying on density estimation for $q_{\boldsymbol{\phi}}(\boldsymbol{x}|\boldsymbol{z})$ we interpret it as an energy-based score function as in Liu et al. (2020).

In energy-based models, the system is optimized such that $\boldsymbol{x}$ belonging to a particular mixture component k will have low free energy $\psi_{\boldsymbol{\phi}}^{k}(\boldsymbol{x})$ relative to component k (LeCun et al., 2006; Liu et al., 2020). For example, in classification problems, the *Helmholtz free energy* relative to component k can be written w.r.t. the log-posterior predictive distribution $\ell_{\boldsymbol{\phi}}^{k}(\boldsymbol{x}, c) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{w}|\boldsymbol{z}=k)}[\log p(y=c|\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{z}=k)]$:

$$\psi_{\boldsymbol{\phi}}^{k}(\boldsymbol{x}) = -T \log \left(\sum_{c=1}^{C} \exp(\ell_{\boldsymbol{\phi}}^{k}(\boldsymbol{x}, c)/T) \right).$$

Here, C is all the known classes until current CL step, and T a temperature parameter of the free energy of component k. Note that $\ell_{\boldsymbol{\phi}}^{k}(\boldsymbol{x}, c)$ can be estimated using a single sample of $\boldsymbol{w}$. Similar energy functions can be derived for other tasks (LeCun et al., 2006). We would expect higher energy for data points $\boldsymbol{x}$ not belonging to component k, allowing us to assign data points into their respective mixture components. Specifically, for $k \in \{1, \ldots, K\}$, we have

$$q_{\boldsymbol{\phi}}(\boldsymbol{z}=k|\boldsymbol{x}) = \frac{\exp\left(-\psi_{\boldsymbol{\phi}}^{k}(\boldsymbol{x})\right)}{\sum_{i=1}^{K} \exp\left(-\psi_{\boldsymbol{\phi}}^{i}(\boldsymbol{x})\right) + e^{-\alpha}},$$

where α is a parameter controlling the concentration of the mixture components. Meanwhile, for $k > K$, we have

$$q_{\boldsymbol{\phi}}(\boldsymbol{z}=k|\boldsymbol{x}) = \frac{e^{-\alpha}}{2^{k-K}\left(\sum_{i=1}^{K}\exp\left(-\psi^{i}_{\boldsymbol{\phi}}(\boldsymbol{x})\right)+e^{-\alpha}\right)}.$$

Note that $q_{\boldsymbol{\phi}}(\boldsymbol{z} > K|\boldsymbol{x}) = e^{-\alpha}/(\sum_{i=1}^{K}\exp\left(-\psi^{i}_{\boldsymbol{\phi}}(\boldsymbol{x})\right) + e^{-\alpha})$. Here, the number of active mixture components K can dynamically grow as more data come in. Specifically, when $q_{\boldsymbol{\phi}}(\boldsymbol{z} > K|\boldsymbol{x}) > q_{\boldsymbol{\phi}}(\boldsymbol{z}_i|\boldsymbol{x}), \forall i \in \{1,\ldots,K\}$, we can allocate a new mixture component, growing a new expert module that implicitly defines this distribution. From an energy-based perspective, α can be seen as the average energy of all data points in the system. Such energy-based novelty scores allow us to identify and handle novel data points in a Bayesian fashion.

Having specified our mixing distribution, we now show how to approximate its KL term. Instead of computing the KL term of this distribution directly, we derive a tractable upper bound to the KL term. Specifically, we can divide the KL term into two parts as follows:

$$\begin{aligned}
&\mathrm{KL}(q_{\boldsymbol{\phi}_{t+1}}(\boldsymbol{z}|\boldsymbol{x})||q_{\boldsymbol{\phi}_t}(\boldsymbol{z}|\boldsymbol{x})) = \\
&\quad\sum_{i=1}^{K} q_{\boldsymbol{\phi}_{t+1}}(\boldsymbol{z}=i|\boldsymbol{x})\log\frac{q_{\boldsymbol{\phi}_{t+1}}(\boldsymbol{z}=i|\boldsymbol{x})}{q_{\boldsymbol{\phi}_t}(\boldsymbol{z}=i|\boldsymbol{x})} \\
&\quad+\sum_{i=K+1}^{\infty} q_{\boldsymbol{\phi}_{t+1}}(\boldsymbol{z}=i|\boldsymbol{x})\log\frac{q_{\boldsymbol{\phi}_{t+1}}(\boldsymbol{z}=i|\boldsymbol{x})}{q_{\boldsymbol{\phi}_t}(\boldsymbol{z}=i|\boldsymbol{x})}.
\end{aligned} \tag{7}$$

Then, by applying Jensen's inequality to the second term:

$$\begin{aligned}
&\mathrm{KL}(q_{\boldsymbol{\phi}_{t+1}}(\boldsymbol{z}|\boldsymbol{x})||q_{\boldsymbol{\phi}_t}(\boldsymbol{z}|\boldsymbol{x})) \leq \\
&\quad\sum_{i=1}^{K} q_{\boldsymbol{\phi}_{t+1}}(\boldsymbol{z}=i|\boldsymbol{x})\log\frac{q_{\boldsymbol{\phi}_{t+1}}(\boldsymbol{z}=i|\boldsymbol{x})}{q_{\boldsymbol{\phi}_t}(\boldsymbol{z}=i|\boldsymbol{x})} \\
&\quad+ q_{\boldsymbol{\phi}_{t+1}}(\boldsymbol{z}>K|\boldsymbol{x})\log\frac{q_{\boldsymbol{\phi}_{t+1}}(\boldsymbol{z}>K|\boldsymbol{x})}{q_{\boldsymbol{\phi}_t}(\boldsymbol{z}>K|\boldsymbol{x})}
\end{aligned} \tag{8}$$

In other words, by defining the following $K+1$ categorical distribution $q'_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$:

$$q'_{\boldsymbol{\phi}}(\boldsymbol{z}=k|\boldsymbol{x}) = q_{\boldsymbol{\phi}}(\boldsymbol{z}=k|\boldsymbol{x}),\ \ k \in \{1,\ldots,K\} \tag{9}$$

$$q'_{\boldsymbol{\phi}}(\boldsymbol{z}=K+1|\boldsymbol{x}) = q_{\boldsymbol{\phi}}(\boldsymbol{z}>K|\boldsymbol{x}), \tag{10}$$

we can use the KL divergence of this distribution as a tractable upper bound to the KL divergence of our original distribution:

$$\mathrm{KL}(q_{\boldsymbol{\phi}_{t+1}}(\boldsymbol{z}|\boldsymbol{x})||q_{\boldsymbol{\phi}_t}(\boldsymbol{z}|\boldsymbol{x})) \leq \mathrm{KL}(q'_{\boldsymbol{\phi}_{t+1}}(\boldsymbol{z}|\boldsymbol{x})||q'_{\boldsymbol{\phi}_t}(\boldsymbol{z}|\boldsymbol{x})).$$

3.3. Implicit Expert Distribution

We now define the expert distribution $q_{\boldsymbol{\phi}}(\boldsymbol{w}|\boldsymbol{z}) = q_{\boldsymbol{\phi}}(\boldsymbol{w}_{\boldsymbol{z}})$. For each $\boldsymbol{z}$, we define $\boldsymbol{w}_{\boldsymbol{z}}$ as the parameters of a Bayesian Neural Network (BNN). To regularize the BNN ensuring that it does not overfit and to maintain a small memory size, we posit that only a small percentage of weights in the BNN are nonzero so here we apply a sparsification method. Indeed, this is inline with recent findings on neural network model compression (Frankle & Carbin, 2018), as well as existing variational inference methods for encourgaging sparse activations in BNNs (Ardywibowo et al., 2022a;b; 2020; Boluki et al., 2020). So we separate $\boldsymbol{w}$ into two parameter groups $\boldsymbol{w} = \{\boldsymbol{\omega}, \boldsymbol{m}\}$. Here $\boldsymbol{\omega}$ are the weights and biases commonly found in a standard BNN. For a Bayesian treatment of $\boldsymbol{\omega}$, approximate Bayesian inference techniques for neural networks, such as Monte Carlo dropout and its variants can be used (Gal & Ghahramani, 2016; Gal et al., 2017; Kingma et al., 2015; Boluki et al., 2020; Kumar et al., 2021). Alternatively, one can employ MAP estimation of $\boldsymbol{\omega}$ and place Gaussian priors to induce $\mathcal{L}^2$ weight decay regularization (Krogh & Hertz, 1992; Vladimirova et al., 2019).

On the other hand, $\boldsymbol{m}$ is a learnable mask parameter that decide which weights and biases of $\boldsymbol{\omega}$ are active or set to zero. With this, we can define a sparsifying prior for $\boldsymbol{m}$. Specifically, let m_k be stochastic binary variables that determine whether weight k is used. To simplify our exposition, we will remove the subscript k and introduce them later for conciseness. For each weight k, we define a prior distribution $p(m)$ for each binary variable as

$$p(m) = \mathtt{Bern}(e^{-\eta}). \tag{11}$$

Here, η is a parameter controlling the shape of the prior. We then define a variational distribution $q(m) = q_{\phi}(m)$ with parameters $\phi \in (-\infty, \infty)$ by transforming random variables from an explicit distribution $\epsilon \sim p(\epsilon)$ using a reparameterizable transformation as follows:

$$\epsilon \sim p(\epsilon), \quad m = \zeta(\phi, \epsilon) \quad \Rightarrow \quad m \sim q_{\boldsymbol{\phi}}(m). \tag{12}$$

Here, $\zeta(\phi, \epsilon)$ outputs a binary random variable that determines whether the weight corresponding to m is used, where $m = \zeta(\phi, \epsilon)$. By defining $p(\epsilon)$ as the logistic distribution with probability density $f(\epsilon)$, and $\zeta(\phi, \epsilon)$ through the sigmoid function $\sigma(\phi) \in (0, 1)$ as follows:

$$\begin{aligned}
&\epsilon \sim p(\epsilon), \qquad f(\epsilon) = \frac{e^{-\epsilon}}{(1+e^{-\epsilon})^2}, \\
&\zeta(\phi,\epsilon) \;=\; \mathbb{1}\left[\log\left(\frac{\sigma(\phi)}{1-\sigma(\phi)}\right) + \epsilon > 0\right],
\end{aligned}$$

we have that $q_{\phi}(m = 1) = \mathbb{E}_{p(\epsilon)}[\zeta(\phi, \epsilon)] = \sigma(\phi)$. In practice, $\epsilon \sim p(\epsilon)$ can be sampled as $\epsilon = \log u - \log(1 - u)$, where $u \sim \mathtt{Unif}(0, 1)$. With this, the KL divergence between the prior and posterior can be computed as

$$\mathrm{KL}(q(\boldsymbol{m})||p(\boldsymbol{m})) = \sum_{k=1}^{K}\mathrm{KL}(q_{\boldsymbol{\phi}}(m_k)||p(m_k)),$$

Table 1. Results on CIFAR100-B0 benchmark (averaged over three runs). Parameters are counted by millions. *Dashes indicate results were not reported by the authors.

Method	5 Steps		10 Steps		20 Steps		50 Steps	
	Params.	Acc. (%)	Params.	Acc. (%)	Params.	Acc. (%)	Params.	Acc. (%)
Bound	11.2	80.40	11.2	80.41	11.2	81.49	11.2	81.74
iCaRL (Rebuffi et al., 2017)	11.2	71.14	11.2	65.27	11.2	61.20	11.2	56.08
UCIR (Hou et al., 2019)	11.2	62.77	11.2	58.66	11.2	58.17	11.2	56.86
BiC (Hou et al., 2019)	11.2	73.10	11.2	68.80	11.2	66.48	11.2	62.09
WA (Zhao et al., 2020)	11.2	72.81	11.2	69.46	11.2	67.33	11.2	64.32
PODNet (Douillard et al., 2020)	11.2	66.70	11.2	58.03	11.2	53.97	11.2	51.19
AANets (Liu et al., 2021)	11.2	67.59	11.2	65.66	-	-	-	-
RPSNet (Rajasegaran et al., 2019)	60.6	70.50	56.5	68.60	-	-	-	-
DER (Yan et al., 2021a)	2.89	75.55	4.96	74.64	7.21	73.98	10.15	72.05
CN-DPM (Lee et al., 2020) (Agnostic)	19.2	20.34	19.2	17.60	19.2	18.79	19.2	19.70
VariGrow (Agnostic)	2.97	75.50	4.88	75.04	7.30	74.03	10.25	72.21

$$\begin{aligned}\mathrm{KL}(q_{\boldsymbol\phi}(m_k)\|p(m_k)) = &-H[q_{\boldsymbol\phi}(m_k)] + \eta q_{\boldsymbol\phi}(m_k=1)\\ &-\log\left(1-e^{-\eta}\right) q_{\boldsymbol\phi}(m_k=0),\end{aligned}$$

where $H[q_{\boldsymbol\phi}(m_k)]$ is the entropy of $q_{\boldsymbol\phi}(m_k)$. For sufficiently large η, $\log\left(1-e^{-\eta}\right)\approx 0$. We achieve this by scaling η with N, $\eta = N\lambda$. We can then scale the negative ELBO with the number of samples N without changing the optima.

$$\mathrm{KL}(q_{\boldsymbol\phi}(m_k)\|p(m_k)) \approx -\frac{1}{N}H[q_{\boldsymbol\phi}(m_k)] + \frac{N\lambda}{N}q_{\boldsymbol\phi}(m_k=1).$$

For large N, the entropy term vanishes, leaving us with

$$\mathrm{KL}(q_{\boldsymbol\phi}(m_k)\|p(m_k)) \approx \lambda q_{\boldsymbol\phi}(m_k=1) = \lambda\sigma(\phi_k).$$

Here the KL term penalizes the model for using too many weights on average, enabling sparsity of each expert. The discrete nature of the selection variables makes it not immediately amenable to gradient-based optimization through reparameterization. To deal with this challenge, we adopt the Gumbel-softmax reparameterization trick (Jang et al., 2016; Maddison et al., 2016) to relax the discrete random variables. This amounts to replacing the indicator function in $\zeta(\phi,\epsilon)$ with a sigmoid scaled by temperature τ, $\zeta(\phi,\epsilon) \approx \tilde{\zeta}(\phi,\epsilon) = \sigma\Big(\big(\log\big(\frac{\sigma(\phi)}{1-\sigma(\phi)}\big) + \epsilon\big)/\tau\Big)$. Meanwhile, during testing, we can use $\zeta(\phi,\epsilon)$ directly and remove the stochasticity. The masking parameters seemingly take more resources but in experiments we show that they can reduce and sparsify the BNN nicely.

4. Experiments

In this section, we conduct extensive experiments to validate the effectiveness of VariGrow. We evaluate our method on 3 datasets: CIFAR-100 (Rebuffi et al., 2017), ImageNet-100 (Rebuffi et al., 2017), and ImageNet-1000 (Rebuffi et al., 2017), using two commonly used benchmark protocols. After detailing our experimental setups and implementation details in Section 4.1, we present and discuss experimental results on the CIFAR-100 dataset and both ImageNet-100 and ImageNet-1000 datasets in Sections 4.2 and 4.3.

4.1. Experimental Setups

Datasets: CIFAR-100 (Krizhevsky et al., 2009) consists of 60,000 32x32 pixel color images ranging over 100 classes. The dataset is divided into 50,000 training images with 500 images per class, and 10,000 images for evaluation with 100 images per class. ImageNet-1000 (Deng et al., 2009) is a large-scale dataset consisting of 1,000 classes, including about 1.2 million RGB images for training and 50,000 images for validation. ImageNet-100 (Rebuffi et al.,

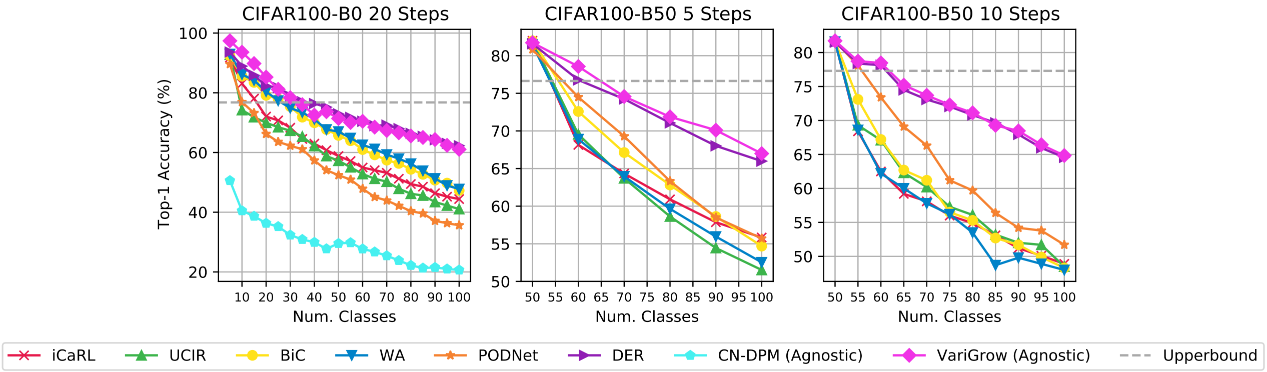


Figure 2. Class-incremental performance comparisons at each step for the CIFAR-100 dataset.

Table 2. Results on CIFAR100-B50 (averaged over three runs). Parameters are counted by millions.

Method	2 Steps		5 Steps		10 Steps	
	Params.	Acc. (%)	Params.	Acc. (%)	Params.	Acc. (%)
Bound	11.2	77.22	11.2	79.89	11.2	79.91
iCaRL (Rebuffi et al., 2017)	11.2	71.33	11.2	65.06	11.2	58.59
UCIR (Hou et al., 2019)	11.2	67.21	11.2	64.28	11.2	59.92
BiC (Hou et al., 2019)	11.2	72.47	11.2	66.62	11.2	60.25
WA (Zhao et al., 2020)	11.2	71.43	11.2	64.01	11.2	57.86
PODNet (Douillard et al., 2020)	11.2	71.30	11.2	67.25	11.2	64.04
DER (Yan et al., 2021a)	3.90	74.57	6.13	72.60	8.79	72.45
VariGrow (Agnostic)	**3.63**	**74.62**	**6.01**	**73.97**	**8.55**	**72.75**

2017; Hou et al., 2019; Yan et al., 2021b) is a subset of it by selecting 100 classes from the ImageNet-1000 dataset.

Benchmark Protocols: For the **CIFAR-100**, we test our methods on two widely used protocols: 1) CIFAR100-B0 (Rebuffi et al., 2017; Yan et al., 2021b): a protocol which divides all 100 classes into 5, 10, 20, and 50 incremental steps with a fixed memory size of 2,000 exemplars over batches; 2) CIFAR100-B50 (Hou et al., 2019; Yan et al., 2021b): a protocol which starts from a model trained on 50 classes, while the remaining 50 classes are divided into splits of 2, 5, and 10 incremental steps with 20 examples as memory per class. We compare the top-1 average incremental accuracy, which takes the average of the accuracy for each step. We follow similar protocols for **ImageNet-100**: 1) ImageNet100-B0 (Rebuffi et al., 2017; Yan et al., 2021b): the protocol trains the model in batches of 10 classes from scratch with a fixed memory size 2,000 over batches; 2) ImageNet100-B50 (Hou et al., 2019; Yan et al., 2021b): the protocol starts from a model trained on 50 classes while the remaining 50 classes come in 10 steps with 20 exemplars per class. For fair comparisons, we use the same ImageNet subset and class order done by Rebuffi et al. (2017), Hou et al. (2019), and Yan et al. (2021b). For **ImageNet-1000**, we evaluate our method on the ImageNet1000-B0 benchmark (Rebuffi et al., 2017; Yan et al., 2021b), that trains the model in batches of 100 classes with 10 steps in total and set a fixed memory size as 20,000 exemplars, with the same class order by Rebuffi et al. (2017) for ImageNet-1000. For both ImageNet-100 and ImageNet-1000, we compare the top-1 and top-5 average incremental accuracy, as well as the last step accuracy. For the task-agnostic setting during *training* and *testing* we hide task IDs, which is called *single-head* setting. The task-aware setting (i.e. *multi-head*) is using one prediction head for each task and effectively predicting the labels within a task instead the whole label pool. For baselines, we compare against various state-of-the-art (i) **task-aware** CL methods: iCaRL (Rebuffi et al., 2017) is memory-based that picks exemplars by balancing the number of class labels. UCIR (Hou et al., 2019), uses normalized feature vectors for prediction. BiC (Hou et al., 2019) trains a bias correction layer on a validation set. WA (Zhao et al., 2020) corrects biased weights by aligning the norm of the weight vectors of new classes to weight vectors of old classes. PODNet (Douillard et al., 2020) uses a spatial distillation loss penalizing parameter changes. TPCIL (Tao et al., 2020) attempts to preserve the topology of the latent feature space, AANets (Liu et al., 2021) attempt to solve the stability-plasticity dilemma by proposing stable and plastic blocks, RPSNet (Rajasegaran et al., 2019) is a path selection algorithm that progressively chooses optimal paths as sub-network for the new classes. DER (Yan et al., 2021a) dynamically grows the network

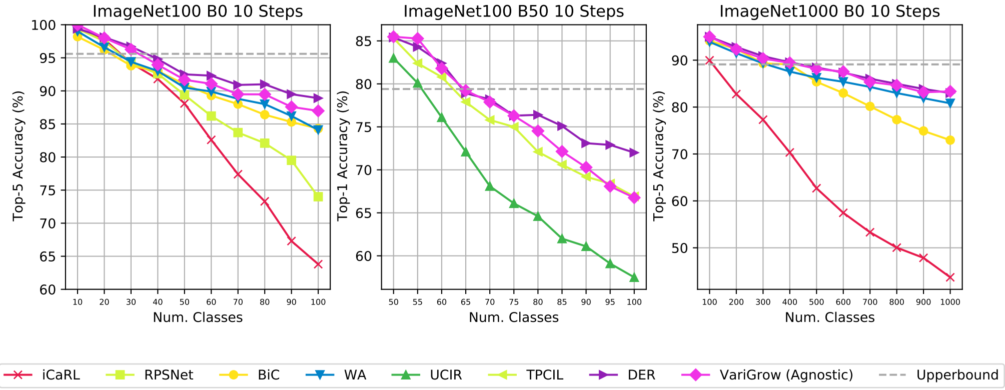


Figure 3. Class-incremental performance comparisons at each step for the ImageNet-100 and ImageNet-1000.

Table 3. Results on ImageNet-B0 (averaged over three runs). Parameters are counted by millions. Avg is the average accuracy (%) over steps. Last is the accuracy (%) evaluate on each task for the model at the last incremental step. *Dashes indicate results were not reported.

Method	ImageNet100-B0					ImageNet1000-B0				
	Params.	Top-1		Top-5		Params.	Top-1		Top-5	
		Avg	Last	Avg	Last		Avg	Last	Avg	Last
Bound	11.2	-	-	-	95.1	11.2	89.27	-	-	-
iCaRL (Rebuffi et al., 2017)	11.2	-	-	83.6	63.8	11.2	38.4	22.7	63.7	44.0
BiC (Hou et al., 2019)	11.2	-	-	90.6	84.4	11.2	-	-	84.0	73.2
WA (Zhao et al., 2020)	11.2	-	-	91.0	84.1	11.2	65.67	55.6	86.6	81.1
RPSNet (Rajasegaran et al., 2019)	-	-	-	87.9	74.0	-	-	-	-	-
AANets (Liu et al., 2021)	11.2	75.58	-	-	-	11.2	64.85	-	-	-
DER (Yan et al., 2021a)	7.67	76.12	66.06	92.79	88.38	14.52	66.73	58.62	87.08	81.89
VariGrow (Agnostic)	7.82	76.04	65.87	92.51	88.17	14.89	66.58	58.47	86.88	81.70

using given task labels. We also benchmark against (ii) **task-agnostic** CL methods: CN-DPM (Lee et al., 2020), a hybrid expansion- and memory-based method, and UCB (Ebrahimi et al., 2020), a regularization-based BNN model. We were not able to reproducible UCB's results reliably, with an accuracy of only 40.34% on the CIFAR10/100 testing protocol. One agnostic method (Aljundi et al., 2019) was not considered due to the capacity can only handle smaller dataset.

Implementation Details: For all datasets, we adopt ResNet-18 (He et al., 2016) as the architecture of our expert modules, following RPSNet (Rajasegaran et al., 2019) and DER (Yan et al., 2021b). We run experiments on three different class orders and report the average of the results. In these experiments, we treat the exemplars variationally, following Nguyen et al. (2018) and select new exemplars (i.e. coreset) as novel mixture components are encountered based on the herding selection strategy (Welling, 2009). We also use these exemplars along with OoD datasets to further calibrate our energy-based novelty score, following the selection of Liu et al. (2020). We use Tiny-ImageNet (Le & Yang) and LSUN (Yu et al., 2015) as OoD datasets for CIFAR-100 and ImageNet experiments respectively. We perform MAP estimation of the neural network weights $\boldsymbol{\omega}$ using Gaussian priors, equivalent to adding a 5×10^{-4} weight decay coefficient. We set $\lambda = 1$ for the prior distribution of $\boldsymbol{m}$, and set $T = 1$, and $\alpha = 18$ for the energy-based novelty scores. We optimize our formulation using SGD with a learning rate of 0.1, batch size of 128 for CIFAR-100, and 256 for ImageNet. We train for 120 epochs and decay the learning rate by 0.1 after 30, 60, and 90 epochs.

Table 4. Results on ImageNet-B50 (averaged over three runs). Parameters are counted by millions. *Dashes indicate results were not reported by the authors.

Method	ImageNet100-B50				
	Params.	Top-1		Top-5	
		Avg	Last	Avg	Last
Bound	11.2	81.20	81.5	-	-
UCIR (Hou et al., 2019)	11.2	68.09	57.3	-	-
PODNet (Douillard et al., 2020)	11.2	74.33	-	-	-
TPCIL (Tao et al., 2020)	11.2	74.81	66.91	-	-
DER (Yan et al., 2021a)	8.87	77.73	72.06	94.01	91.64
VariGrow (Agnostic)	8.94	77.64	71.48	92.84	89.95

4.2. Evaluation on CIFAR100

Quantitative Results: Table 1 and Figure 2 (left) show the results for CIFAR100-B0. We can see that, without needing task labels nor task switching information, our method is competitive with state-of-the-art CL methods which are task-aware. Meanwhile, ours significantly outperforms CN-DPM, a task-agnostic online learning formulation, with an improvement of over +50%. Moreover, the margin between our method and CN-DPM continuously increases, indicating that our method performs better over longer continual learning episodes with fewer parameters with our sparsification. Note also that we are getting very close to the offline multi-task learning baseline (Bound). This demonstrates that VariGrow is able to learn from a non-*iid* data stream without much decrease in performance despite not having access to the entire dataset.

We further compare the performance of VariGrow on the CIFAR100-B50 benchmark in Table 2 and Figure 2 (middle, right), again showing that our method is competitive with task-aware continual learning methods. We note that DER is the most competitive task-aware method in our benchmarks but it has to grow the network architecture with given task switching information.

To further banchmark the efficacy of our method in handling task-agnosticism, we study the effects of two settings where task-agnosticism can occur. One setting involves removing

Table 5. Results on different task-agnostic settings on the CIFAR100-B50 benchmark.

Setting	Accuracy (%)	
	5 Steps	10 Steps
Baseline	73.97	72.45
Lookback Old Tasks	71.21	70.98
Fuzzy Boundaries	70.03	69.19

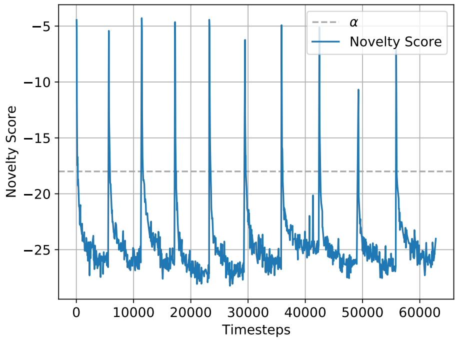


Figure 4. Novelty score values for each iteration on the CIFAR100-B50 10-step protocol.

the clear task boundaries, and instead gradually introducing data from novel tasks. This setting is similar to the fuzzy task boundary experiment conducted by Lee et al. (2020). The other setting assumes that data from previous tasks can be observed again in between task switches. Thus, our model needs to be able to distinguish these instances and correctly assign them to an existing expert instead of growing a new one. We denote both of these experiments as fuzzy and lookback respectively, and the performance of our method in these settings on the CIFAR-100 dataset can be seen in Table 5. As seen in Table 5, our method suffers only a slight degradation in accuracy in these settings compared to the traditional setting. We hypothesize that this is caused by stray datapoints being incorrectly assigned to the wrong expert.

Qualitative view on energy-based Novelty Score: We show the energy-based novelty score at each timestep for the CIFAR100-B50 10-step protocol in Figure 4. Here, we see that our novelty score clearly helps detect task changes, with significantly increased novelty scores after a new task is observed. Note also that VariGrow is able to correctly detect that there are 10 tasks with 10 observed peaks.

4.3. Evaluation on ImageNet

We show results for the ImageNet-100 and ImageNet-1000 datasets in Tables 3, 4, and Figure 3. We see that our VariGrow is again competitive with task-aware methods for all splits on these two datasets, which are more complex compared to CIFAR100. We note that the gap in top-5 accuracy is smaller. We believe that this is because the top-5 accuracy is more tolerant to slightly inaccurate predictions and thus less sensitive to catastrophic forgetting.

5. Conclusions

We have presented **VariGrow**, a variational architecture growing formulation to solve strict task-agnostic continual learning. VariGrow defines an implicit variational construct to approximate the nonparametric posterior at each incremental CL step, giving a Bayesian interpretation of growing networks. Using energy-based novelty detection, we are able to dynamically grow the CL prediction model only when needed and reliably assign data points and thereafter corresponding tasks into different expert mixture components, where each component can be handled by expert distributions defined implicitly by neural networks. Our extensive performance evaluation experiments on both CIFAR and ImageNet show that VariGrow significantly outperforms existing task-agnostic CL methods and is competitive even against task-aware CL methods.

Acknowledgments

This project is supported in part by the Defense Advanced Research Projects Agency (DARPA) under grant FA8750-18-2-0027. X. Qian has been supported in part by the National Science Foundation (NSF) Awards 1553281, 1812641, 1835690, 1934904, 1956219, and 2119103.

References

Abati, D., Tomczak, J., Blankevoort, T., Calderara, S., Cucchiara, R., and Bejnordi, B. E. Conditional channel gated networks for task-aware continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3931–3940, 2020.

Aljundi, R. Continual learning in neural networks. *arXiv preprint arXiv:1910.02718*, 2019. URL `https://arxiv.org/abs/1910.02718`.

Aljundi, R., Chakravarty, P., and Tuytelaars, T. Expert gate: Lifelong learning with a network of experts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3366–3375, 2017.

Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., and Tuytelaars, T. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 139–154, 2018.

Aljundi, R., Lin, M., Goujaud, B., and Bengio, Y. Gradient based sample selection for online continual learning. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 11816–11825. Curran Associates, Inc., 2019.

Ardywibowo, R. *Analyzing Daily Behavioral Data for Personalized Health Management*. PhD thesis, 2017.

Ardywibowo, R., Huang, S., Gui, S., Xiao, C., Cheng, Y., Liu, J., and Qian, X. Switching-state dynamical modeling of daily behavioral data. *Journal of Healthcare Informatics Research*, 2(3):228–247, 2018.

Ardywibowo, R., Zhao, G., Wang, Z., Mortazavi, B., Huang, S., and Qian, X. Adaptive activity monitoring with uncertainty quantification in switching Gaussian process models. In *The 22nd International Conference on Artificial Intelligence and Statistics (AISTATS 2019)*, 2019.

Ardywibowo, R., Boluki, S., Gong, X., Wang, Z., and Qian, X. NADS: Neural architecture distribution search for uncertainty awareness. In *International Conference on Machine Learning*, pp. 356–366. PMLR, 2020.

Ardywibowo, R., Boluki, S., Wang, Z., Mortazavi, B. J., Huang, S., and Qian, X. Vfds: Variational foresight dynamic selection in bayesian neural networks for efficient human activity recognition. In *International Conference on Artificial Intelligence and Statistics*, pp. 1359–1379. PMLR, 2022a.

Ardywibowo, R., Dayana, V. R. K., and Hwang, H. Dynamic quantization for energy efficient deep learning, March 31 2022b. US Patent App. 17/488,261.

Barber, D. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

Blei, D. M. and Jordan, M. I. Variational inference for dirichlet process mixtures. *Bayesian analysis*, 1(1):121–143, 2006.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

Boluki, S., Ardywibowo, R., Dadaneh, S. Z., Zhou, M., and Qian, X. Learnable Bernoulli dropout for Bayesian deep learning. In Chiappa, S. and Calandra, R. (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 3905–3916. PMLR, 26–28 Aug 2020. URL http://proceedings.mlr.press/v108/boluki20a.html.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Douillard, A., Cord, M., Ollion, C., Robert, T., and Valle, E. PODNet: Pooled outputs distillation for small-tasks incremental learning. In *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, 2020. URL https://www.ecva.net/papers/eccv_2020/papers_ECCV/papers/123650086.pdf.

Ebrahimi, S., Elhoseiny, M., Darrell, T., and Rohrbach, M. Uncertainty-guided continual learning with bayesian neural networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HklUCCVKDB.

Frankle, J. and Carbin, M. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

Gal, Y. and Ghahramani, Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pp. 1050–1059. PMLR, 2016.

Gal, Y., Hron, J., and Kendall, A. Concrete dropout. *Advances in Neural Information Processing Systems*, 30, 2017.

Grathwohl, W., Wang, K.-C., Jacobsen, J.-H., Duvenaud, D., Norouzi, M., and Swersky, K. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations*, 2019.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

He, X. and Jaeger, H. Overcoming catastrophic interference using conceptor-aided backpropagation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B1al7jg0b.

Hendrycks, D. and Gimpel, K. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

Hjort, N. L., Holmes, C., Müller, P., and Walker, S. G. *Bayesian nonparametrics*, volume 28. Cambridge University Press, 2010.

Hou, S., Pan, X., Loy, C. C., Wang, Z., and Lin, D. Learning a unified classifier incrementally via rebalancing. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with Gumbel-Softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Jiang, Z., Ardywibowo, R., Samereh, A., Evans, H. L., Lober, W. B., Chang, X., Qian, X., Wang, Z., and Huang, S. A roadmap for automatic surgical site infection detection and evaluation using user-generated incision images. *Surgical infections*, 20(7):555–565, 2019.

Kaushik, P., Gain, A., Kortylewski, A., and Yuille, A. Understanding catastrophic forgetting and remembering in continual learning with optimal relevance mapping. *arXiv preprint arXiv:2102.11343*, 2021.

Kessler, S., Nguyen, V., Zohren, S., and Roberts, S. Hierarchical indian buffet neural networks for bayesian continual learning. *arXiv preprint arXiv:1912.02290*, 2019.

Kingma, D. P., Salimans, T., and Welling, M. Variational dropout and the local reparameterization trick. *Advances in neural information processing systems*, 28:2575–2583, 2015.

Kirichenko, P., Farajtabar, M., Rao, D., Lakshminarayanan, B., Levine, N., Li, A., Hu, H., Wilson, A. G., and Pascanu, R. Task-agnostic continual learning with hybrid probabilistic models. In *ICML Workshop on Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models*, 2021.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. Overcoming catastrophic forgetting in neural networks. *Proc. of the national academy of sciences*, 2017. URL https://www.pnas.org/content/pnas/114/13/3521.full.pdf.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Krogh, A. and Hertz, J. A. A simple weight decay can improve generalization. In *Advances in neural information processing systems*, pp. 950–957, 1992.

Kumar, A., Chatterjee, S., and Rai, P. Bayesian structural adaptation for continual learning. In *International Conference on Machine Learning*, pp. 5850–5860. PMLR, 2021.

Kurle, R., Cseke, B., Klushyn, A., van der Smagt, P., and Günnemann, S. Continual learning with bayesian neural networks for non-stationary data. In *International Conference on Learning Representations*, 2019.

Le, Y. and Yang, X. Tiny imagenet visual recognition challenge.

LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., and Huang, F. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Lee, S., Ha, J., Zhang, D., and Kim, G. A neural Dirichlet process mixture model for task-free continual learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJxSOJStPr.

Li, L., Jun, Z., Fei, J., and Li, S. An incremental face recognition system based on deep learning. In *2017 Fifteenth IAPR international conference on machine vision applications (MVA)*, pp. 238–241. IEEE, 2017.

Lin, D. Online learning of nonparametric mixture models via sequential variational approximation. *Advances in Neural Information Processing Systems*, 26:395–403, 2013.

Liu, W., Wang, X., Owens, J., and Li, Y. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 33, 2020.

Liu, Y., Schiele, B., and Sun, Q. Adaptive aggregation networks for class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2544–2553, 2021.

Lopez-Paz, D. and Ranzato, M.-A. Gradient episodic memory for continual learning. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 6467–6476. Curran Associates, Inc., 2017.

Maddison, C. J., Mnih, A., and Teh, Y. W. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

Mescheder, L., Geiger, A., and Nowozin, S. Which training methods for gans do actually converge? In *International conference on machine learning*, pp. 3481–3490. PMLR, 2018.

Mohseni, S., Pitale, M., Yadawa, J., and Wang, Z. Self-supervised learning for generalizable out-of-distribution detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5216–5223, 2020.

Molchanov, D., Kharitonov, V., Sobolev, A., and Vetrov, D. Doubly semi-implicit variational inference. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2593–2602. PMLR, 2019.

Nalisnick, E., Matsukawa, A., Teh, Y. W., Gorur, D., and Lakshminarayanan, B. Do deep generative models know what they don't know? In *International Conference on Learning Representations*, 2018.

Nguyen, C. V., Li, Y., Bui, T. D., and Turner, R. E. Variational continual learning. In *International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1710.10628.

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

Pierre, J. M. Incremental lifelong deep learning for autonomous vehicles. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 3949–3954. IEEE, 2018.

Pomponi, J., Scardapane, S., Lomonaco, V., and Uncini, A. Efficient continual learning in neural networks with embedding regularization. *Neurocomputing*, 397:139–148, 2020.

Rajasegaran, J., Hayat, M., Khan, S. H., Khan, F. S., and Shao, L. Random path selection for incremental learning. *CoRR*, abs/1906.01120, 2019. URL http://arxiv.org/abs/1906.01120.

Rajasegaran, J., Khan, S., Hayat, M., Khan, F. S., and Shah, M. iTAML: An incremental task-agnostic meta-learning approach. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13588–13597, 2020.

Ranzato, M., Boureau, Y.-L., Chopra, S., and LeCun, Y. A unified energy-based framework for unsupervised learning. In *Artificial Intelligence and Statistics*, pp. 371–379. PMLR, 2007.

Rao, D., Visin, F., Rusu, A. A., Teh, Y. W., Pascanu, R., and Hadsell, R. Continual unsupervised representation learning, 2019. URL https://arxiv.org/pdf/1910.14481.pdf.

Rebuffi, S.-A., Kolesnikov, A., Sperl, G., and Lampert, C. H. ICARL: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017. URL https://arxiv.org/abs/1611.07725.

Riemer, M., Cases, I., Ajemian, R., Liu, M., Rish, I., Tu, Y., and Tesauro, G. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *International Conference on Learning Representations*, 2018.

Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., and Hadsell, R. Progressive neural networks. *ArXiv e-prints*, jun 2016. URL https://arxiv.org/abs/1606.04671.

Schwarz, J., Luketina, J., Czarnecki, W. M., Grabska-Barwinska, A., Teh, Y. W., Pascanu, R., and Hadsell, R. Progress & compress: A scalable framework for continual learning. In *ICML*, 2018. URL https://arxiv.org/abs/1805.06370.

Shin, H., Lee, J. K., Kim, J., and Kim, J. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, pp. 2990–2999, 2017. URL https://arxiv.org/abs/1705.08690.

Sodhani, S., Chandar, S., and Bengio, Y. Toward training recurrent neural networks for lifelong learning. *Neural computation*, 32(1):1–35, 2020.

Tao, X., Chang, X., Hong, X., Wei, X., and Gong, Y. Topology-preserving class-incremental learning. In *European Conference on Computer Vision*, pp. 254–270. Springer, 2020.

Thrun, S. and Mitchell, T. M. Lifelong robot learning. In *The biology and technology of intelligent autonomous agents*, pp. 165–196. Springer, 1995. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.71.3723&rep=rep1&type=pdf.

Titsias, M. K. and Ruiz, F. Unbiased implicit variational inference. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 167–176. PMLR, 2019.

Titsias, M. K., Schwarz, J., Matthews, A. G. d. G., Pascanu, R., and Teh, Y. W. Functional regularisation for continual learning with Gaussian processes. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1901.11356.

Vladimirova, M., Verbeek, J., Mesejo, P., and Arbel, J. Understanding priors in bayesian neural networks at the unit level. In *International Conference on Machine Learning*, pp. 6458–6467. PMLR, 2019.

Welling, M. Herding dynamical weights to learn. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 1121–1128, 2009.

Wortsman, M., Ramanujan, V., Liu, R., Kembhavi, A., Rastegari, M., Yosinski, J., and Farhadi, A. Supermasks in superposition. *Advances in Neural Information Processing Systems*, 33, 2020.

Yan, S., Xie, J., and He, X. DER: Dynamically expandable representation for class incremental learning, 2021a.

Yan, S., Xie, J., and He, X. DER: Dynamically Expandable Representation for class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3014–3023, 2021b.

Yin, M. and Zhou, M. Semi-implicit variational inference. In *International Conference on Machine Learning*, pp. 5660–5669. PMLR, 2018.

Yoon, J., Yang, E., Lee, J., and Hwang, S. J. Lifelong learning with dynamically expandable networks. In *International Conference on Learning Representations*, 2018.

Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., and Xiao, J. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv e-prints*, pp. arXiv–1506, 2015.

Zeno, C., Golan, I., Hoffer, E., and Soudry, D. Task agnostic continual learning using online variational Bayes, 2018. URL https://arxiv.org/pdf/1803.10123.pdf.

Zhao, B., Xiao, X., Gan, G., Zhang, B., and Xia, S.-T. Maintaining discrimination and fairness in class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13208–13217, 2020.