Published in "Neurocomputing", 2020, vol. 404, pp. 381-400, which should be cited to refer to this work.
DOI: 10.1016/j.neucom.2020.02.115

# Lifelong Generative Modeling

Jason Ramapuram[a,*], Magda Gregorova[a], Alexandros Kalousis[a]

[a] *University of Geneva & University of Applied Sciences, HES-SO,
7 route de Drize, 1227 Carouge, Switzerland*

## Abstract

Lifelong learning [1, 2] is the problem of learning multiple consecutive tasks in a sequential manner where knowledge gained from previous tasks is retained and used for future learning. It is essential towards the development of intelligent machines that can adapt to their surroundings. In this work we focus on a lifelong learning approach to generative modeling where we continuously incorporate newly observed distributions into our learnt model. We do so through a student-teacher Variational Autoencoder[3] architecture which allows us to learn and preserve all the distributions seen so far without the need to retain the past data nor the past models. Through the introduction of a novel cross-model regularizer, inspired by a Bayesian update rule, the student model leverages the information learnt by the teacher, which acts as a summary of everything seen till now. The regularizer has the additional benefit of reducing the effect of catastrophic interference that appears when we learn over sequences of distributions. We demonstrate its efficacy in learning sequentially observed distributions as well as its ability to learn a common latent representation across a complex transfer learning scenario. We validate our model's performance on MNIST, FashionMNIST, SVHN and Celeb-A and demonstrate that our model mitigates the effects of catastrophic interference faced by neural networks in sequential learning scenarios. Our code is available: `https://github.com/jramapuram/LifelongVAE_pytorch`.

[*] Corresponding author
*Email address:* Jason.Ramapuram@etu.unige.ch (Jason Ramapuram)

*Keywords:* Lifelong Learning, Continual Learning, Variational Inference, Generative Modeling

## 1. Introduction

Deep unsupervised generative learning allows us to take advantage of the massive amount of unlabeled data available in order to build models that efficiently compress and learn an approximation of the true data distribution. It has wide ranging applications from image denoising [4, 5], inpainting [6, 7], super-resolution [8], structured prediction [9], clustering [10], and pre-training [11]. However, something that is lacking in the modern ML toolbox is an efficient way to learn these deep generative models in a lifelong setting. In a lot of real world scenarios we observe distributions sequentially; children in elementary school for example learn the alphabet letter-by-letter and in a sequential manner. Other real world examples include video data from sensors such as cameras and microphones or other similar sequential data. A system can also be resource limited wherein all of the past data or learnt models cannot be stored. The navigation of a resource limited robot in an unknown environment for instance, might require the robot to be able to inpaint images from a learnt generative model in a previous environment.

In the lifelong learning setting we sequentially observe a single distribution at a time from a possibly infinite set of distributions. Our objective is to learn a *single model* that is able to generate from *each* of the individual distributions *without the preservation of the observed data*[1]. We provide an example of such a setting in figure 1(a) using MNIST [12], where we sequentially observe three distributions. Since we only observe one distribution at a time we need to develop a strategy of retaining the previously learnt distributions and integrating it into future learning. To accumulate additional distributions in the current

[1] This setting is drastically different from the online learning setting; we touch upon this in Appendix 3.2

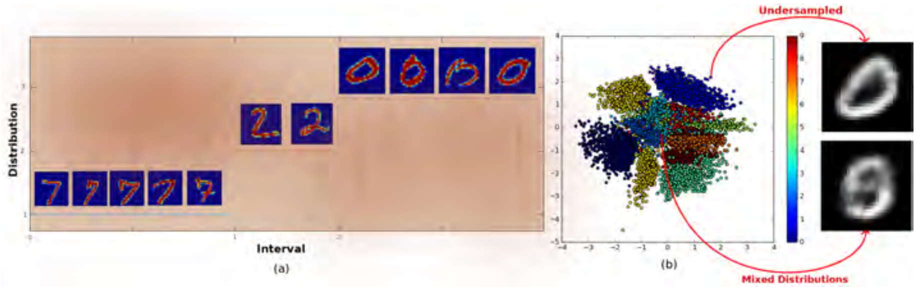


Figure 1: (a) Our problem setting where we sequentially observe samples from multiple unknown distributions and need to learn a *common generative model*; (b)Visualization of a learnt two-dimensional posterior of MNIST, evaluated with samples from the full test set. We depict the two generative shortcomings visually: 1) mixing of distributions which causes aliasing in a lifelong setting and 2) undersampling of distributions in a standard VAE posterior.

generative model we utilize a student-teacher architecture similar to those in distillation methods [13, 14]. The teacher contains a summary of all past distributions and is used to augment the data used to train the student model. The student model thus receives data samples from the currently observable distribution as well as synthetic data samples from previous distributions. Once a distribution shift occurs the existing teacher model is *discarded*, the student becomes the teacher and a new student is instantiated.

We introduce a novel regularizer in the form of a Bayesian update rule that allows us to bring the posterior of the student close to that of the teacher for the synthetic data generated by the teacher. This allows us to build upon and extend the teacher's inference model into the student each time the latter is re-instantiated (rather than re-learning it from scratch). By coupling this regularizer with an initial weight transfer [2] from the teacher to the student we also allow for faster convergence of the student model. We empirically show that this regularizer mitigates the effect of catastrophic interference [15]. It also ensures that even though our model evolves over time, it preserves the ability to

[2] This is enacted by simply copying the initial weights from the teacher to the student during a distribution transition.

generate samples from any of the previously observed distributions, a property we call *consistent sampling*.

While the model we present focuses on the generative, unsupervised setting it is possible to extend it to a classification or regression setting through a marginalization operand of the full joint distribution. However, since unsupervised generative modelling is an under-represented sub domain in lifelong learning, we focus our all our efforts on this setting. We choose to build our lifelong generative models using Variational Autoencoders (VAEs) [3] as they provide a mechanism for stable training (in contrast to Generative Adversarial Networks (GAN) [16] based methods [17]), simple generation (Section 4) and latent-variable posterior approximation: a requirement in many learning scenarios such as clustering [18], compression [19] and unsupervised representation learning [20]. In addition, GANs can suffer from low sample diversity [21] which can lead to compounding errors in a lifelong generative setting.

Using a standard VAE decoder to generate synthetic data for the student is problematic due to a couple of limitations of the VAE generative process as shown in Figure 1(b). 1) Sampling the prior can select a point in the latent space that is in between two separate distributions, causing generation of unrealistic synthetic data and eventually leading to loss of previously learnt distributions; 2) data points mapped to the posterior that are further away from the prior mean will be sampled less frequently resulting in an undersampling of some of the constituent distributions[3]. To address these sampling limitations we decompose the latent variable vector into a continuous and a discrete component (Section 4.3). The discrete component summarizes the discriminative information of the individual generative distributions while the continuous caters for the remaining sample variability (a nuisance variable [22]). By independently sampling the discrete and continuous components we preserve the distributional boundaries

[3] This is due to the fact that VAE's generate data by sampling their prior (generally an isotropic standard gaussian) and decoding the sample through the decoder neural network. Thus a posterior instance further from the prior mean is sampled less frequently.

and circumvent the two VAE limitations described above.

## 2. Related Work

The idea of learning in a continual, sequential manner have been explored extensively in machine learning, seeded by the seminal works of Never-Ending-Language-Learning (NELL) [23, 24] and Lifelong-Learning [1, 2, 25]. NELL was developed to accumulate and build semantic language relationships from an initial set of ontologies. It learns and builds it's representation through self-supervision [23]. In contrast, Lifelong Learning [1] was initially proposed as a framework to study lifelong concept learning, where each observed task tries to associate a particular concept/class using binary classification [26]. It was later extended to reinforcement learning [2] and neural networks [25] in a supervised setting. We extend the central tenants of Lifelong Learning proposed in [1, 2, 25] and focus our efforts on generative modeling with deep neural networks: an under-represented area within the domain.

One of the key obstacles for a neural lifelong learner is the effect of catastrophic interference [15]. Model parameters of a neural network trained in a sequential manner tend to be biased towards the distribution of the latest observations, while forgetting what was learnt previously over data no longer accessible for training. Lifelong / continual learning aims to mitigate the effects of catastrophic interference using four major strategies: *transfer learning*, *replay mechanisms*, *parameter regularization* and *distribution regularization.*

**Transfer learning** : These approaches attempt to solve the problem of catastrophic interference by relaying previously learnt information to the current model. Methods such as Progressive Neural Networks [27] and Deep Block-Modular Neural Networks [28] for example, transfer a hidden layer representation from previously learnt models into the new model. The problem with transfer learning approaches is that they generally require the preservation of **all** previously learnt **model parameters** and thus do not scale with a large number of tasks.

**Replay mechanisms** : Recently there have been a few efforts to use generative replay in order to avoid catastrophic interference in a classification setting [29, 30]. These methods work by regenerating previous samples and using them (in conjunction with newly observed samples) in future learning. Neither of these however, leverage information from the previously trained models. Instead they simply re-learn each new joint task from scratch.

**Parameter regularization** : Methods such as Elastic Weight Consolidation (EWC) [31], Synaptic Intelligence [32] and Variational Continual Learning (VCL) [33] constrain the *parameters* of the new model to be close to the previous model through a predefined metric. EWC [31] for example, utilizes the Fisher Information matrix (FIM) to control the change of model parameters between two tasks. Intuitively, important parameters should not have their values changed, while non-important parameters are left unconstrained. The FIM is used as a weighting in a quadratic parameter difference regularizer under a Gaussianity assumption of the (parameter) posterior $P(\boldsymbol{\theta}|\boldsymbol{X})$. This assumption has been hypothesized [34] and later demonstrated empirically [35] to be sub-optimal for learnt neural network weights.

In addition to the rigid parameter-posterior restriction mentioned above, VCL also violates two of the requirements for a practical lifelong learning algorithm: a separate head network is added per task (reducing the solution to Progressive Neural Networks [27]) and a core-set of true data-samples is stored *per observed distribution*. Both of these requirements prevent the scalability of VCL to a truly lifelong setting due to the continual addition of extra parameters and extra data.

**Distribution regularization** : In contrast, methods such as distillation [13], ALTM [14] and Learning Without Forgetting (LwF) [36] constrain the outputs of models from different tasks to be similar. This can be interpreted as *distributional regularization* by generalizing the constraining metric (or semi-metric) to be a divergence on the output conditional distribution, i.e: $\mathcal{D}[P_{\boldsymbol{\theta}_i}(\boldsymbol{y}|\mathbf{x} = x_{<i}) \; || \; P_{\boldsymbol{\theta}_{<i}}(\boldsymbol{y}|\mathbf{x} = x_{<i})]$. One of the pitfalls of distribution regularization is that it necessitates the preservation of the previously observed data $x_{<i}$ which is a

violation of the lifelong learning setting where data from old distributions are no longer accessible.

Our work builds on the distribution regularization and replay strategies. In contrast to standard distribution regularization, where the constraint is applied on the output distribution, we apply our regularizer on the amortized, approximate posterior of the VAE (Section 4.1). In addition, we do not assume a parametric form for the distribution of the model's posterior $P(\boldsymbol{\theta}|\boldsymbol{X})$ as in EWC or VCL and allow the model to constrain the parameters between two tasks in a highly non-linear way (Section 4.2). By combining our replay mechanism with information transfer from the previous model, we **increase the training efficiency** in terms of the number of required training epochs (Experiment 5.1), while at the same time **not preserving any previous data** and only requiring **constant**[4] extra model storage.

## 3. Background

We consider an unsupervised setting where we observe a dataset $\mathbf{X}$ of $K \geq 1$ realizations $\mathbf{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(K)}\}$ from an unknown true distribution $P^*(\mathbf{x})$ with $\mathbf{x} \in \mathcal{R}^N$. We assume that the data is generated by a random process involving a non-observed random variable $\boldsymbol{z} \in \mathcal{R}^M$. In order to incorporate our prior knowledge, we posit a prior $P(\boldsymbol{z})$ over $\boldsymbol{z}$. Our objective is to approximate the true underlying data distribution by a model $P_{\boldsymbol{\theta}}(\mathbf{x})$ such that $P_{\boldsymbol{\theta}}(\mathbf{x}) \approx P^*(\mathbf{x})$.

Given a latent variable model $P_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})P(\mathbf{z})$ we obtain the marginal likelihood $P_{\boldsymbol{\theta}}(\mathbf{x})$ by integrating out the latent variable $\mathbf{z}$ from the joint distribution. The joint distribution can in turn be factorized using the conditional distribution $P_{\boldsymbol{\theta}}(\mathbf{x}|\boldsymbol{z})$ or the posterior $P_{\boldsymbol{\theta}}(\boldsymbol{z}|\mathbf{x})$:

$$P_{\boldsymbol{\theta}}(\mathbf{x}) = \int P_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{z})\delta \boldsymbol{z} = \int P_{\boldsymbol{\theta}}(\mathbf{x}|\boldsymbol{z})P(\boldsymbol{z})\delta \boldsymbol{z} \tag{1}$$

[4] We only require one teacher and student model as opposed to [27, 28]which require keeping all previous models

We model the conditional distribution $P_{\boldsymbol{\theta}}(\mathbf{x}|\boldsymbol{z})$ by a *decoder*, typically a neural network. Very often the marginal likelihood $P_{\boldsymbol{\theta}}(\mathbf{x})$ will be intractable because the integral in equation (1) does not have an analytical form nor an efficient estimator [37]. As a result, the respective posterior distribution, $P_{\boldsymbol{\theta}}(\boldsymbol{z}|\mathbf{x})$, is also intractable.

Variational inference side-steps the intractability of the posterior by approximating it with a tractable distribution $Q_{\boldsymbol{\phi}}(\boldsymbol{z}|\mathbf{x}) \approx P_{\boldsymbol{\theta}}(\boldsymbol{z}|\mathbf{x})$. VAEs use an *encoder* (generally a neural network) to model the approximate posterior $Q_{\boldsymbol{\phi}}(\boldsymbol{z}|\mathbf{x})$ and optimize the parameters $\boldsymbol{\phi}$ to minimize the reverse KL divergence $KL[Q_{\boldsymbol{\phi}}(\boldsymbol{z}|\mathbf{x})||P_{\boldsymbol{\theta}}(\boldsymbol{z}|\mathbf{x})]$ between the approximate posterior distribution $Q_{\boldsymbol{\phi}}(\boldsymbol{z}|\mathbf{x})$ and the true posterior $P_{\boldsymbol{\theta}}(\boldsymbol{z}|\mathbf{x})$. Given that $Q_{\boldsymbol{\phi}}(\boldsymbol{z}|\mathbf{x})$ is a powerful model (such that the KL divergence against the true posterior will be close to zero) we maximize the tractable Evidence Lower BOund (ELBO) to the intractable marginal likelihood $\mathcal{L}_{\boldsymbol{\theta}}(\mathbf{x}) \leq P_{\boldsymbol{\theta}}(\mathbf{x})$ (full derivation available in Appendix Section 9.6)

$$\text{ELBO:} \quad \mathcal{L}_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbb{E}_{Q_{\boldsymbol{\phi}}(\boldsymbol{z}|\mathbf{x})}[\log P_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{z}) - \log Q_{\boldsymbol{\phi}}(\boldsymbol{z}|\mathbf{x})] \tag{2}$$

By sharing the variational parameters $\boldsymbol{\phi}$ of the encoder across the data points (*amortized inference* [38]), variational autoencoders avoid per-data optimization loops typically needed by mean-field approaches.

### *3.1. Lifelong Generative Modeling*

The standard setting in maximum-likelihood generative modeling is to estimate the set of parameters $\boldsymbol{\theta}$ that will maximize the marginal likelihood, $P_i(\mathbf{x}; \boldsymbol{\theta})$, for dataset $\mathbf{X}_i$, generated IID from a single true data distribution $P_i^*(\mathbf{x})$. Latent variable models $P_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) = P_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})P(\mathbf{z})$ (such as VAEs) capture the complex structures in $P^*(\mathbf{x})$ by conditioning the observed variables $\mathbf{x}$ on the latent variables $\mathbf{z}$ and combining these in (possibly infinite) mixtures $P_{\boldsymbol{\theta}}(\mathbf{x}) = \int P_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})P(\mathbf{z})\delta\boldsymbol{z}$.

Our sequential setting is vastly different from the standard approach described above. We receive a sequence of (possibly infinite) datasets $\mathbf{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_i, \ldots\}$

where each dataset $\mathbf{X}_i = \{\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}, \ldots, \mathbf{x}_i^{(K_i)}\}$ originates from a disparate distribution $P_i^*(\mathbf{x})$ . At any given time we observe the latest dataset $\mathbf{X}_i$ (observed as a set of minibatch samples) generated from a single distribution $P_i^*(\mathbf{x})$ without access to any of the previous observed datasets $\mathbf{X}_{<i}$. As depicted in Figure 1(a), our goal is to learn a *single model* that is able to generate samples from *each of the observed distributions* $\{P_1^*(\mathbf{x}), \ldots, P_i^*(\mathbf{x}), \ldots\}$, without the addition of an approximation model $P_i(\mathbf{x}; \boldsymbol{\theta}) \approx P_i^*(\mathbf{x})$ per observed distribution.

### *3.2. Contrast to streaming / online methods*

Our method has similarities to streaming methods such as Streaming Variational Bayes (SVB) [39] and Incremental Bayesian Clustering methods [40, 41] in that we estimate and refine posteriors through time. In general this can be done through the following Bayesian update rule that states that the lastest posterior is proportional to the current likelihood times the previous posterior:

$$P(\boldsymbol{z}|\mathbf{X}_1, ..., \mathbf{X}_t) \propto P(\mathbf{X}_t|\boldsymbol{z})P(\boldsymbol{z}|\mathbf{X}_1, ..., \mathbf{X}_{t-1}) \tag{3}$$

SVB computes the intractable posterior, $P(\boldsymbol{z}|\mathbf{X}_1, ..., \mathbf{X}_t)$, utilizing an approximation, $\mathcal{A}_t$, that accepts as input the current dataset, $\mathbf{X}_t$, along with the previous posterior $\mathcal{A}_{t-1}$ :

$$P(\boldsymbol{z}|\mathbf{X}_1, ..., \mathbf{X}_t) \approx \mathcal{A}_t(\mathbf{X}_t, \mathcal{A}_{t-1}) \tag{4}$$

The first posterior input ($\mathcal{A}_{t=0}$) to the approximating function is the prior $P(\boldsymbol{z})$. The objective of SVB and other streaming methods is to model the posterior of the currently observed data in the best possible manner. Our setting differs from this in that we want to retain information from *all previously observed distributions* (sometimes called a knowledge store [1]). This can be useful in scenarios where a distribution is seen once, but only used much later down the road. Rather than creating a posterior update rule, we recompute the posterior via Equation 3, leveraging the fact that we can re-generate $\mathbf{X}_{<t} \approx \hat{\mathbf{X}}_{<t}$

through the generative process. This allows us to recompute a more appropriate posterior re-using all of the (generated) data, rather than using the previously computed (approximate) posterior $\mathcal{A}_{t-1}$:

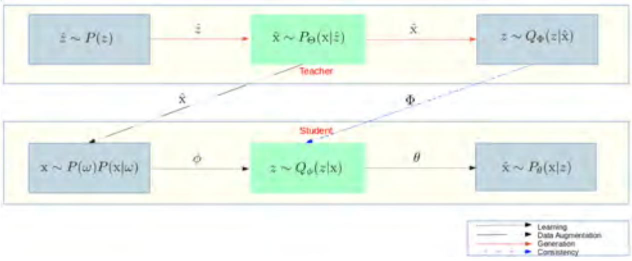
$$P(\boldsymbol{z}|\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_t) \propto P(\mathbf{X}_t|\boldsymbol{z})P(\boldsymbol{z}|\hat{\mathbf{X}}_1, ..., \hat{\mathbf{X}}_{t-1}) \tag{5}$$

Coupling this generative replay strategy with the Bayesian update regularizer introduced in Section 4.1, we demonstrate that not only do we learn an updated poster as in Equation 5, but also allow for a natural transfer of information between sequentially learnt models: a fundamental tenant of lifelong learning [1, 2].

Finally, another key difference between lifelong learning and online methods is that lifelong learning aims to learn from a sequence of *tentatively different* [26] tasks while still retaining and accumulating knowledge; online learning generally assumes that the true underlying distribution comes from a single distribution [42]. There are some exceptions to this where online learning is applied to the problem of domain adaptation, eg: [43, 40]. In our experiments we analyze both scenarios: one where there is a small change in distributional semantics (Experiment 5.1) as well as a scenario in which the new distribution is vastly different (Experiment 5.2). We also focus explicitly on utilizing deep neural networks for generation, a setting that necessitates the mitigation of Catastrophic interference.

## 4. Model

To enable lifelong generative learning we propose a dual model architecture based on a student-teacher model. The teacher and the student have rather different roles throughout the learning process: the teacher's role is to preserve the memory of the previously learnt distributions and to pass this knowledge onto the student; the student's role is to learn the distributions over the new incoming data while accommodating for the knowledge obtained from the teacher. The dual model architecture is summarized in figure 2.



```
Algorithm 1 Data Flow
Teacher:
    Sample Prior: $\hat{\boldsymbol{z}} \sim P(\boldsymbol{z})$
    Decode: $\hat{\mathbf{x}} \sim P_{\boldsymbol{\Theta}}(\mathbf{x}|\hat{\boldsymbol{z}})$
    Encode: $\boldsymbol{z} \sim Q_{\boldsymbol{\Phi}}(\boldsymbol{z}|\hat{\mathbf{x}})$
Student:
    Sample : $\mathbf{x} \sim P(\boldsymbol{\omega})P(\mathbf{x}|\boldsymbol{\omega})$
    Encode : $\boldsymbol{z} \sim Q_{\boldsymbol{\phi}}(\boldsymbol{z}|\mathbf{x})$
    Decode: $\hat{\mathbf{x}} \sim P_{\boldsymbol{\theta}}(\mathbf{x}|\boldsymbol{z})$
```

Figure 2: Shown above is the relationship of the teacher and the student generative models and the algorithm for the flow of data. Data generated from the teacher model is used to augment the student model's training data and consistency is applied between posteriors. Best viewed in color.

The top row represents the teacher model. At any given time the teacher contains a summary of all previous distributions within the learnt parameters $\boldsymbol{\Phi}$ of the encoder $Q_{\boldsymbol{\Phi}}(\boldsymbol{z}|\mathbf{x})$ and the learnt parameters $\boldsymbol{\Theta}$ of the decoder $P_{\boldsymbol{\Theta}}(\mathbf{x}|\boldsymbol{z})$. The teacher is used to generate synthetic samples $\hat{\mathbf{x}}$ from these past distributions by decoding samples from the prior $\hat{\boldsymbol{z}} \sim P(\boldsymbol{z})$ through the decoder $\hat{\mathbf{x}} \sim P_{\boldsymbol{\Theta}}(\mathbf{x}|\hat{\boldsymbol{z}})$. The generated synthetic samples $\hat{\mathbf{x}}$ are passed onto the student model as a form of knowledge transfer about the past distributions.

The bottom row of figure 2 represents the student, which is responsible for updating the parameters of the encoder $Q_{\boldsymbol{\phi}}(\boldsymbol{z}|\mathbf{x})$ and decoder $P_{\boldsymbol{\theta}}(\mathbf{x}|\boldsymbol{z})$ over the newly observed data. The student is exposed to a set of learning instances $\mathbf{x}$ sampled from $\mathbf{x} \sim P(\boldsymbol{\omega})P(\mathbf{x}|\boldsymbol{\omega})$, $\boldsymbol{\omega} \sim \mathrm{Ber}(\pi)$; it sees synthetic instances generated by the teacher $P(\mathbf{x}|\boldsymbol{\omega} = 0) = P_{\boldsymbol{\Theta}}(\mathbf{x}|\boldsymbol{z})$, and real ones sampled from the currently active training distribution $P(\mathbf{x}|\boldsymbol{\omega} = 1) = P_i^*(\mathbf{x})$. The mean $\pi$ of the Bernouli distribution controls the sampling proportion of the previously learnt distributions to the current one and is set based on the number of observed datasets. If we have seen k datasets (and thus k distributions) prior to the current one then $\pi = \frac{1}{k+1}$. This ensures that all the current and past distributions are equally represented in the training set used by the student model. Once a new distribution is signalled, the old teacher is dropped, the student model is frozen and becomes the new teacher ($\boldsymbol{\phi} \rightarrow \boldsymbol{\Phi}, \boldsymbol{\theta} \rightarrow \boldsymbol{\Theta}$), and a new student is initiated with the latest weights $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ from the previous student (the new teacher).

### *4.1. Teacher-student consistency*

Our central objective is to learn a *single set of parameters* $[\boldsymbol{\phi}, \boldsymbol{\theta}]$ such that we are able to generate samples from all observed distributions $\{P_1^*(\mathbf{x}), \ldots, P_i^*(\mathbf{x})\}$. Given that we can generate samples $\{\hat{\boldsymbol{X}}_1, \ldots, \hat{\boldsymbol{X}}_{i-1}\}$ for all previous $i-1$ observed distributions via the teacher model, our objective can be formalized as the maximization of the augmented ELBO, $\boldsymbol{\mathcal{L}}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x}_{<i}, \mathbf{x}_i) \approx \hat{\boldsymbol{\mathcal{L}}}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\hat{\mathbf{x}}_{<i}, \mathbf{x}_i)$, under the assumption that $\hat{\mathbf{x}}_{<i} \perp\!\!\!\perp \mathbf{x}_i$:

$$\begin{aligned}\hat{\boldsymbol{\mathcal{L}}}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\hat{\mathbf{x}}_{<i}, \mathbf{x}_i) &= \mathbb{E}_{Q_{\boldsymbol{\phi}}(\boldsymbol{z}|\hat{\mathbf{x}}_{<i}, \mathbf{x}_i)}\Big[\log P_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{<i}, \mathbf{x}_i|\boldsymbol{z}) - KL[Q_{\boldsymbol{\phi}}(\boldsymbol{z}|\hat{\mathbf{x}}_{<i}, \mathbf{x}_i)||P(\boldsymbol{z})]\Big] \\ &= \mathbb{E}_{Q_{\boldsymbol{\phi}}(\boldsymbol{z}|\hat{\mathbf{x}}_{<i}, \mathbf{x}_i)}\Big[\log P_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{<i}, \mathbf{x}_i|\boldsymbol{z}) - KL[Q_{\boldsymbol{\phi}}(\boldsymbol{z}|\mathbf{x}_i)||P(\boldsymbol{z})] - KL[Q_{\boldsymbol{\phi}}(\boldsymbol{z}|\hat{\mathbf{x}}_{<i})||P(\boldsymbol{z})]\Big]\end{aligned} \tag{6}$$

The ELBO $\hat{\boldsymbol{\mathcal{L}}}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\hat{\mathbf{x}}_{<i}, \mathbf{x}_i)$, in Equation 6 is approximate due to the fact that we use the generations $\hat{\boldsymbol{X}}_{<i} \sim P_{\boldsymbol{\Theta}}(\hat{\mathbf{x}}|\boldsymbol{z})$ instead of the true data $\boldsymbol{X}_{<i}$ [5]. Rather than naively shrinking the full posterior to the prior via the KL divergence in Equation 6 we introduce a posterior regularizer $KL[Q_{\boldsymbol{\phi}}(\boldsymbol{z}|\hat{\mathbf{x}}_{<i})||Q_{\boldsymbol{\Phi}}(\boldsymbol{z}|\hat{\mathbf{x}}_{<i})]$ that distills the teacher's learnt representation into the student over the generated data $\hat{\boldsymbol{X}}_{<i}$ [6]. We will now show how this regularizer can be perceived as a natural extension of the VAE learning objective across the combined dataset $\{\hat{\boldsymbol{X}}_{<i}, \boldsymbol{X}_i\}$ through the lens of a Bayesian update of the student posterior.

**Lemma 1.** *For random variables* $\hat{\mathbf{x}}$ *and* $\boldsymbol{z}$ *with conditionals* $Q_{\boldsymbol{\Phi}^*}(\boldsymbol{z}|\hat{\mathbf{x}})$ *and* $Q_{\boldsymbol{\phi}}(\boldsymbol{z}|\hat{\mathbf{x}})$*, both distributed as a categorical or gaussian and parameterized by* $\boldsymbol{\Phi}^*$ *and* $\boldsymbol{\phi}$ *respectively, the KL divergence between the distributions is:*

$$KL[Q_{\boldsymbol{\phi}}(\boldsymbol{z}|\hat{\mathbf{x}})||Q_{\boldsymbol{\Phi}^*}(\boldsymbol{z}|\hat{\mathbf{x}})] = KL[Q_{\hat{\boldsymbol{\phi}}}(\boldsymbol{z}|\hat{\mathbf{x}})||P(\boldsymbol{z})] + C(\boldsymbol{\Phi}^*) \tag{7}$$

[5] Note that the ELBO described in Equation 6 is still a single sample ELBO of $\mathbf{x} = [\hat{\mathbf{x}}_{<i}, \mathbf{x}_i]$; we overload this notation here to imply that we can regenerate many samples similar to the true dataset using the decoder network.

[6] While it is also possible to apply a similar cross-model regularizer to the reconstruction term, i.e: $KL[P_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{<i}|\boldsymbol{z}) \;||\; P_{\boldsymbol{\Theta}}(\hat{\mathbf{x}}_{<i}|\boldsymbol{z})]$, we observe that doing so hurts performance (Appendix 9.2).

*where $\hat{\boldsymbol{\phi}} = f(\boldsymbol{\phi}, \boldsymbol{\Phi}^*)$ depends on the parametric form of Q, and C is only a function of $\boldsymbol{\Phi}^*$.*

We prove Lemma 1 for the relevant distributions (under some mild assumptions) in Appendix 9.1. Using Lemma 1 and the assumption that $\hat{\mathbf{x}}_{<i} \perp\!\!\!\perp \mathbf{x}_i$, Equation 6 can be interpreted as a standard VAE ELBO under a reparameterization $\hat{\boldsymbol{\phi}} = f(\boldsymbol{\phi}, \boldsymbol{\Phi}^*)$:

$$\hat{\mathcal{L}}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\hat{\mathbf{x}}_{<i}, \mathbf{x}_i) = \mathbb{E}_{Q_{\boldsymbol{\phi}}(\boldsymbol{z}|\hat{\mathbf{x}}_{<i},\mathbf{x}_i)}\Big[\log P_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{<i}, \mathbf{x}_i|\boldsymbol{z}) - KL[Q_{\boldsymbol{\phi}}(\boldsymbol{z}|\mathbf{x}_i)||P(\boldsymbol{z})] - KL[Q_{\hat{\boldsymbol{\phi}}}(\boldsymbol{z}|\hat{\mathbf{x}}_{<i})||P(\boldsymbol{z})]\Big] \tag{8}$$

where the last term $C(\boldsymbol{\Phi}^*)$ is constant with respect to $\boldsymbol{\phi}$ and thus not included in Equation 8. Recasting the problem in such a manner allows us to see that transitioning the ELBO to a sequential setting involves: 1) $KL[Q_{\hat{\boldsymbol{\phi}}}(\boldsymbol{z}|\hat{\mathbf{x}}_{<i})||P(\boldsymbol{z})]$, a term bringing the student posterior (as a function of both itself and the teacher parameters) close to the prior for previously observed data and 2) $KL[Q_{\boldsymbol{\phi}}(\boldsymbol{z}|\mathbf{x}_i)||P(\boldsymbol{z})]$, the standard term in the ELBO that attempts to bring the student posterior close to the prior for the current data.

Naively evaluating the student ELBO using $\hat{\mathbf{x}}_{<i}$, the synthetic teacher data and $\mathbf{x}_i$, the real current data, results in equation 6. While the change seems minor, it omits the introduction of $f(\boldsymbol{\phi}, \boldsymbol{\Phi}^*)$ which allows for a transfer of information between models. In practice, we analytically evaluate $KL[Q_{\boldsymbol{\phi}}(\boldsymbol{z}|\hat{\mathbf{x}}_{<i})\, ||Q_{\boldsymbol{\Phi}^*}(\boldsymbol{z}|\hat{\mathbf{x}}_{<i})]$, the KL divergence between the teacher and the student posteriors, instead of deriving the functional form of $f(\boldsymbol{\phi}, \boldsymbol{\Phi}^*)$ for each different distribution pair.

### *4.2. Contrast To EWC*

Our distribution regularizer $KL[Q_{\boldsymbol{\phi}}(\boldsymbol{z}|\hat{\mathbf{x}}_{<i})||Q_{\boldsymbol{\Phi}}(\boldsymbol{z}|\hat{\mathbf{x}}_{<i})]$ affects the same parameters $\boldsymbol{\phi}$ as parameter regularizer methods such as EWC. However, it does so in a non-linear manner dependent on the underlying network structure as opposed to the fixed functional form of the distance metric in EWC. We will demonstrate in our experiments that our proposed method does no worse than EWC in the worst case (i.e. when the EWC constraint is a valid distance metric

assumption as in Experiment 5.1), but drastically outperforms EWC in the case when this is not true (Experiment 5.2).

| EWC $\min_{\boldsymbol{\phi}} d[P(\boldsymbol{\phi}\|\mathbf{x})\|\|P(\boldsymbol{\Phi}\|\mathbf{x})]$ | Lifelong ( Isotropic Gaussian Posterior ) $\min_{\boldsymbol{\phi}} d[Q_{\boldsymbol{\phi}}(\boldsymbol{z}\|\mathbf{x})\|\|Q_{\boldsymbol{\Phi}}(\boldsymbol{z}\|\mathbf{x})]$ |
|---|---|
| $\approx \frac{\gamma}{2}(\boldsymbol{\phi}-\boldsymbol{\Phi})^T F(\boldsymbol{\phi}-\boldsymbol{\Phi})$ | $= 0.5\Big[tr(\boldsymbol{\Sigma}_{\boldsymbol{\Phi}}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{\phi}}) + (\boldsymbol{\mu}_{\boldsymbol{\Phi}} - \boldsymbol{\mu}_{\boldsymbol{\phi}})^T\boldsymbol{\Sigma}_{\boldsymbol{\Phi}}^{-1}(\boldsymbol{\mu}_{\boldsymbol{\Phi}} - \boldsymbol{\mu}_{\boldsymbol{\phi}}) - C + log\Big(\frac{\|\boldsymbol{\Sigma}_{\boldsymbol{\Phi}}\|}{\|\boldsymbol{\Sigma}_{\boldsymbol{\phi}}\|}\Big)\Big]$ |

In the above table we examine the distance metric $d$, used to minimize the effects of catastrophic inference in both EWC and our proposed Lifelong method. While our method can operate over any distribution that has a tractable KL-divergence, for the purposes of demonstration we examine the simple case of an isotropic gaussian *latent-variable posterior*. EWC directly enforces a quadratic constraint on the model parameters $\boldsymbol{\phi}$, while our method indirectly affects the same parameters through a regularization of the posterior distribution $Q_{\boldsymbol{\phi}}(\boldsymbol{z}|\mathbf{x})$. For a given input $\mathbf{x}$ in the Lifelong case, the only freedom the model has is to change $\boldsymbol{\phi}$; it does so in a non-linear[7] way such that the analytical KL shown above is minimized.

### *4.3. Latent variable*

A critical component of our model is the synthetic data generation by the teacher's model $P(\boldsymbol{z}) \mapsto P_{\boldsymbol{\Theta}}(\hat{\mathbf{x}}|\boldsymbol{z})$. The synthetic samples need to be representative of all the previously observed distributions in order to provide the student with ample information about the learning history. Considering only the case of teacher generated samples $\hat{\mathbf{x}} \sim P(\mathbf{x}|\boldsymbol{\omega} = \mathbf{0})$: the minibatch of N samples received by the student after k distribtions $\{\hat{\mathbf{x}}_1 \sim P_{\boldsymbol{\Theta}}(\hat{\mathbf{x}}|\boldsymbol{z}), ..., \hat{\mathbf{x}}_N \sim P_{\boldsymbol{\Theta}}(\hat{\mathbf{x}}|\boldsymbol{z})\}$ should contain approximately $\frac{N}{k}$ samples from each of the k observed distributions in order to prevent catastrophic forgetting.

A simple unimodal prior distribution $P(\boldsymbol{z})$, such as the isotropic Gaussian typically used in classical VAEs (see Figure 1(b)), results in an undersampling of distributions in the posterior that are further away from the prior mean. This in turn leads to catastrophic forgetting of the undersampled distributions in the student model. We circumvent this problem by decomposing the posterior

[7] This is because the parameters of the distribution are modeled by a deep neural network.

distribution into a conditionally independent discrete and a continuous component $Q_{\boldsymbol{\phi}}(\boldsymbol{z}_c, \boldsymbol{z}_d|\mathbf{x}) = Q_{\boldsymbol{\phi}}(\boldsymbol{z}_c|\mathbf{x})Q_{\boldsymbol{\phi}}(\boldsymbol{z}_d|\mathbf{x})$ [8]. We assume a uniform multivariate discrete prior $\boldsymbol{z}_d \sim Cat(\frac{1}{J})$ for the discrete component and a multivariate standard normal prior $\boldsymbol{z}_c \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ for the continuous component. The discrete component $\boldsymbol{z}_d$ is used to summarize the most discriminative information about each of the true generating distributions $P_i^*(\mathbf{x})$, while the continuous component attends to the sample variability (a nuisance variable [22]). This split allows us to directly generate $\lfloor \frac{N}{k} \rfloor$ synthetic samples per observed distribution by fixing a discrete component $\boldsymbol{z}_d$, varying the continuous sample $\boldsymbol{z}_c^i \sim P(\boldsymbol{z}_c),\ i \in \{1..\lfloor \frac{N}{k} \rfloor\}$ and decoding through the decoder network $P_{\boldsymbol{\Theta}}(\hat{\mathbf{x}}|\boldsymbol{z}_d, \boldsymbol{z}_c^i)$.

We empirically validate the benefit of this posterior decomposition in Experiment 6.1 where we compare against multiple VAE models trained on various typical posteriors such as the isotropic-gaussian, bernouilli and discrete distributions. We demonstrate that this decomposition learns a better disentangled representation than all of the other baselines.

### *4.4. Information restricting regularizer*

In order to enforce that $P_{\boldsymbol{\Theta}}(\hat{\mathbf{x}}|\boldsymbol{z}_d)$ carries the most relevant information for the generative process (which in turn allows us to easily generate samples from an individual previously observed distribution), we introduce a negative information gain regularizer between the continuous representation $\boldsymbol{z}_c$ and the generated data $\hat{\mathbf{x}}$ : $I(\boldsymbol{z}_c; \hat{\mathbf{x}}) = H(\boldsymbol{z}_c) - H(\boldsymbol{z}_c|\hat{\mathbf{x}})$. $H(\boldsymbol{z}_c)$ is used to denote the marginal entropy of $\boldsymbol{z}_c$ and $H(\boldsymbol{z}_c|\hat{\mathbf{x}})$ denotes the conditional entropy of $\boldsymbol{z}_c$ given $\hat{\mathbf{x}}$. This prevents the model from primarily using the continuous representation, while disregarding the discrete one and therefore the pathological $P_{\boldsymbol{\Theta}}(\hat{\mathbf{x}}|\boldsymbol{z}_d, \boldsymbol{z}_c) = P_{\boldsymbol{\Theta}}(\hat{\mathbf{x}}|\boldsymbol{z}_c)$. We utilize a lower bound for this term in a similar manner as done in InfoGAN [44, 16]:

[8] A similar idea is employed in work parallel to our own [21].

$$
\begin{aligned}
I(\boldsymbol{z}_c, \hat{\mathbf{x}}) &= H(\boldsymbol{z}_c) - \mathbb{E}_{\hat{\mathbf{x}} \sim P_{\boldsymbol{\theta}}(\hat{\mathrm{x}}|\boldsymbol{z}_d, \boldsymbol{z}_c)} H(\boldsymbol{z}_c | \mathbf{x} = \hat{\mathbf{x}}) \\
&= H(\boldsymbol{z}_c) - \mathbb{E}_{\hat{\mathbf{x}} \sim P_{\boldsymbol{\theta}}(\hat{\mathrm{x}}|\boldsymbol{z}_d, \boldsymbol{z}_c)} \mathbb{E}_{\boldsymbol{z}_c \sim P(\boldsymbol{z}_c|\mathbf{x})} \log P(\boldsymbol{z}_c | \mathbf{x} = \hat{\mathbf{x}}) \\
&\geq L_I(\boldsymbol{z}_c, \hat{\mathbf{x}}) \\
&= \mathbb{E}_{\hat{\mathbf{x}} \sim P_{\boldsymbol{\theta}}(\hat{\mathrm{x}}|\boldsymbol{z}_d, \boldsymbol{z}_c)} \mathbb{E}_{\boldsymbol{z}_c \sim Q_{\boldsymbol{\phi}}(\boldsymbol{z}_c|\mathbf{x})} \log Q_{\boldsymbol{\phi}}(\boldsymbol{z}_c | \mathbf{x} = \hat{\mathbf{x}})
\end{aligned}
\tag{9}
$$

Rather than maximizing the mutual information between $\boldsymbol{z}_d$ and $\hat{\mathbf{x}}$ (as in InfoGAN) and introduce a min-max optimization problem, we instead minimize the information of the continuous component as an equivalent problem formulation. Since our model doesn't utilize skip connections, information from the input data has to flow through the latent variables $\boldsymbol{z} = [\boldsymbol{z}_c, \boldsymbol{z}_d]$ to reach the decoder. Minimizing the information gain between $\boldsymbol{z}_c$ and the generated decoded sample $\hat{\mathbf{x}}$ forces the model to dominantly use $\boldsymbol{z}_d$.

In contrast to InfoGAN, VAEs already estimate the posterior $Q_{\boldsymbol{\phi}}(\boldsymbol{z}_c|\mathbf{x})$ and thus do not need the introduction of any extra parameters $\boldsymbol{\phi}$ for the approximation. Finally, as opposed to InfoGAN, which uses the variational bound (twice) on the mutual information [45], our regularizer has a clear interpretation: it restricts information through a specific latent variable within the computational graph. We observe that this constraint is essential for empirical performance of our model and empirically validate this in our ablation study in Experiment 6.2.

### *4.5. Contrast to VASE*

The recent work of Life-Long Disentangled Representation Learning with Cross-Domain Latent Homologies (VASE) [46] extend upon our work [46, p. 7], but take a more empirical route by incorporating a classification-based heuristic for their posterior distribution. In contrast, we show (Section 4.1) that our objective naturally emerges in a sequential learning setting for VAEs, allowing us to learn the full joint distribution $P(\mathbf{x}, \boldsymbol{z})$ in an unsupervised manner. Due to the incorporation of direct supervised class information [46] also observe that regularizing the decoding distribution $P_{\boldsymbol{\theta}}(\mathbf{x}|\boldsymbol{z})$ aids in the learning process, something that we observe to fail in a purely unsupervised generative setting (Appendix Section 9.2). Finally, in contrast to [46], we include an information restricting regularizer (Section 4.4) which allows us to directly control the interpretation and flow of information of the learnt latent variables.

### *4.6. Learning Objective*

The final learning objective for each of the student models is the maximization of the augmented ELBO discussed in Section 4.1 and the negative information gain term proposed in Section 4.4.

$$\begin{aligned}\underbrace{\mathbb{E}_{Q_{\boldsymbol{\phi}}}[log\ P_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{<i}, \mathbf{x}_i|\boldsymbol{z}_d, \boldsymbol{z}_c)] - KL[Q_{\boldsymbol{\phi}}(\boldsymbol{z}_d, \boldsymbol{z}_c|\hat{\mathbf{x}}_{<i}, \mathbf{x}_i)||P(\boldsymbol{z}_c, \boldsymbol{z}_d)]}_{\text{Augmented ELBO}} \\ - \underbrace{KL[Q_{\boldsymbol{\phi}}(\boldsymbol{z}_d|\hat{\mathbf{x}}_{<i})||Q_{\boldsymbol{\Phi}}(\boldsymbol{z}_d|\hat{\mathbf{x}}_{<i})]}_{\text{Consistency Regularizer}} \\ - \underbrace{\lambda L_I(\boldsymbol{z}_c, \hat{\mathbf{x}}_{<i}, \mathbf{x}_i)}_{\text{Information Gain}}\end{aligned} \tag{10}$$

The $\lambda$ hyper-parameter controls the importance of the information gain regularizer. Too large a value for $\lambda$ causes a lack of sample diversity, while too small a value causes the model to not use the discrete latent distribution. We did a random hyperparameter search and determined $\lambda = 0.01$ to be a reasonable choice for all of our experiments. This is in line with the $\lambda$ used in InfoGAN [44] for continuous latent variables. We empirically validate the necessity of both terms proposed in Equation 10 in our ablation study in Experiment 6.2. We also validate the benefit of the latent variable factorization in Experiment 6.1.

## 5. Experiments

In all of our experiments we focus on the performance benefits our architecture and augmented learning objective brings into the lifelong learning setting which is the main motivation of our work. To do this we divide our experiments into three distinct problems, namely *sequential learning of similar distributions*

(Experiments 5.1, 6.2), *sequential learning of disparate distributions* (Experiment 5.2) and finally two *complex transfer learning problems* (Experiment 5.3, 5.4). Lifelong learning over similar distributions allows us to examine the reusability of the learnt feature representation; on the other hand lifelong learning over disparate distributions and the complex transfer learning settings allow us to explore the extent to which our model can accomodate new information without forgetting previously learnt representations. We evaluate our model and the baselines over *standard datasets* used in other state of the art lifelong / continual learning literature [33, 32, 29, 30, 31, 27]. While these datasets are simple in a classification setting, transitioning to a *lifelong-generative* setting scales the problem complexity substantially. We give details specific to each experiment in their individual sections. Some of the commonalities between the experiments are described below.

In Experiment 5.1 and Experiment 5.2 we compare our model to a set of EWC baselines. For comparability, we use the same student-teacher architecture as in our model, but instead of our consistency regularizer we augment the VAE ELBO by the EWC distance metric between the student ($\boldsymbol{\xi} = [\boldsymbol{\phi}, \boldsymbol{\theta}]$) and teacher ($\boldsymbol{\Xi} = [\boldsymbol{\Phi}, \boldsymbol{\Theta}]$) models. Since we do not have access to the true log-likelihood we estimate the diagonal Fisher information matrix from the ELBO:

| EWC Learning Objective | Fisher Approximation |
|---|---|
| $\underbrace{\mathbb{E}_{Q_\phi}[log\ P_{\boldsymbol{\theta}}(\mathbf{x}\vert\boldsymbol{z})] - KL[Q_{\boldsymbol{\phi}}(\boldsymbol{z}\vert\mathbf{x})\vert\vert P(\boldsymbol{z})]}_{\text{Student VAE ELBO}} - \underbrace{\frac{\gamma}{2}(\boldsymbol{\xi} - \boldsymbol{\Xi})^T F(\boldsymbol{\xi} - \boldsymbol{\Xi})}_{EWC}$ | $F = \mathrm{diag}\left(\nabla_{\boldsymbol{\Xi}} \mathbb{E}_{Q_{\boldsymbol{\Phi}}(\boldsymbol{z}\vert\mathbf{x})}\left[\log\ \frac{P_{\boldsymbol{\Theta}}(\mathbf{x},\boldsymbol{z})}{Q_{\boldsymbol{\Phi}}(\boldsymbol{z}\vert\mathbf{x})}\right]^2\right)$ |

We utilize two major performance metrics for our experiments : the *negative* test ELBO and the Frechet distance[9] (as proposed in [47]). The negative test ELBO provides a lower bound to the test log-likelihood of the true data distribution, while the Frechet distance gives us a quantification of the quality and diversity of the generated samples. Note that lower values are better for both metrics. In both Experiment 5.1 and Experiment 5.2, we run each model five times each and report the mean and standard deviations. We utilize both fully convolutional (-C-) and fully dense architectures (-D-) and list the top

[9] More details about this metric are provided in Appendix Section 9.10

performing models and baselines. We provide the entire set of analyzed baselines in Appendix Section 9.4. In addition, all *network architectures* and other optimization details are provided in Appendix Section 9.3 as well our our git repository.

### *5.1. Lifelong Learning of Similar Distributions*

In this experiment, we demonstrate the performance benefit our architecture and augmented learning objective from Equation 10 bring to the continual learning of a set of related distributions. The hypothesis for working over similar distributions is that models should leverage previously learnt features and use it for future learning. We compare our method (*lifelong-[$\lambda$]*) (where $\lambda$ is the mutual information hyperparameter) against a standard VAE (*vanilla-vae*), a VAE that observes all the data (*full-vae*), a VAE that observes all the data upto (inclusively) the current distribution (*upto-vae*) and finally to a set of EWC baselines (*ewc-[$\gamma$]*) where gamma is the EWC hyperparameter value. The *lifelong*, *ewc* and *vanilla* models only observe one dataset $\boldsymbol{X}_i$ at a time and do not have access to any of the previous true datasets $\boldsymbol{X}_{<i}$. In order to fairly evaluate the test ELBO,we utilize the *same* graphical model (i.e. the discrete and continuous latent variables) for all models.

We use Fashion MNIST [48][10] to simulate our continual learning setting. We treat each object as a different distribution and present the model with samples drawn from a single distribution at a time. Each individual distribution contains 10,000 training samples indicative of the current fashion object (such as shirts, shoes, etc) and a corresponding 1,000 test samples. We sequentially progress over the ten available distributions and report performance metrics on the test set of all distributions seen up to the current point at the end of each training run (quantified by an early-stopping criterion). Note that, the test set is *incrementally increased*, eg: at the second distribution the test set contains samples from the first and second test datasets; the training set on the other

[10] We also report MNIST results for the same experiment in Appendix 9.5

hand only contains samples from the currently observed distribution, $\boldsymbol{X}_i^{\text{train}}$. Since the cardinality of the test set increases, we will observe an increase in the negative test ELBO and Frechet distance. This is due to the fact that the model needs to be able to not only reconstruct (or generate in the case of the Frechet metric) the dataset $\boldsymbol{X}_i^{\text{test}}$ that it just observed, but also all previous test sets $\boldsymbol{X}_{<i}^{\text{test}}$.

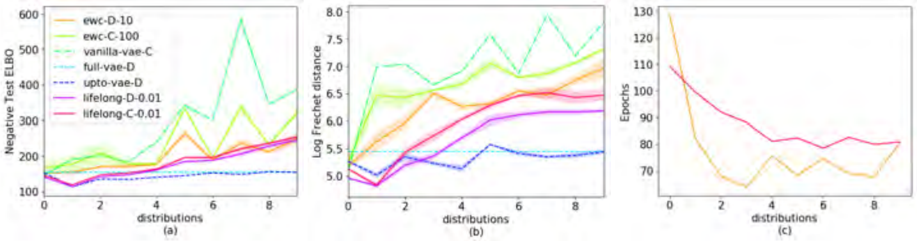


Figure 3: (a)Negative Test ELBO. (b) *Log-Frechet* distance (c) Epochs to trigger early stopping

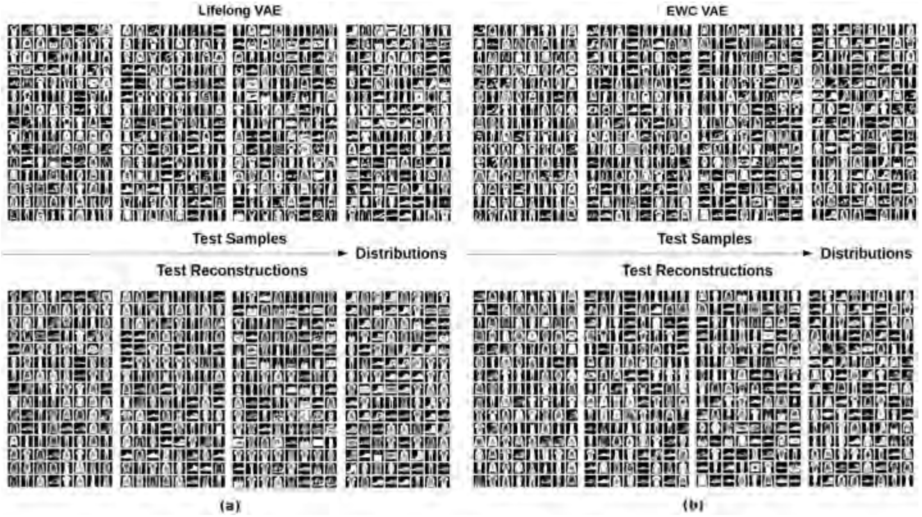


Figure 4: Top row: test-samples; bottom row: reconstructions. We visualize an increasing number of accumulated distributions from left to right and show test samples and reconstructions from the last 4 student models in a sequence of 10 distributions. (a) Lifelong VAE model (b) EWC VAE model

The *full-vae* and *upto-vae* models present the best attainable performance as they have access to all the previous data at all times, and thus do not suffer from catastrophic interference. The *vanilla-vae* on the other displays the worst performance since it catastrophically forgets all previous distributions. Since the *full-vae* is exposed to all data, $\boldsymbol{X}_{i=\{1..10\}}^{\text{train}}$, in a traditional batch setting, it describes the best possible lower bound performance metric on the full test dataset, $\boldsymbol{X}_{i=\{1..10\}}^{\text{test}}$. The *upto-vae* on the other hand differs from the *ewc-vae*, *vanilla-vae* and *lifelong-vae* models in that it has access to $\boldsymbol{X}_{\leq i}^{\text{train}} = \{\boldsymbol{X}_1^{\text{train}}, ..., \boldsymbol{X}_i^{\text{train}}\}$, the training data upto (inclusively) the current datset $\boldsymbol{X}_i^{\text{train}}$. In contrast to the *full-vae*, the *upto-vae* is evaluated on $\boldsymbol{X}_{\leq i}^{\text{test}} = \{\boldsymbol{X}_1^{\text{test}}, ..., \boldsymbol{X}_i^{\text{test}}\}$ in a similar manner as the *ewc* and *lifelong* models. Due to this, we observe (in terms of both the Frechet distance and the negative ELBO) that the *upto-vae* performs better than the *full-vae* up until the fifth distribution. Once we observe all ten distributions, the performance of *upto-vae* is equivalent to that of the *full-vae*. While *upto-vae* and *full-vae* deliver the best performance, they are not viable solutions in a lifelong setting since they need to use all the previous training data. Interestingly enough, we observe that the *lifelong-vae* performs better (in terms of the Frechet ditance) than the *upto-vae* up until the third distribution. We attribute this to the mutual information regularizer presented in Section 4.4 and that at early training, the replay mechanism does not suffer from degraded generations.

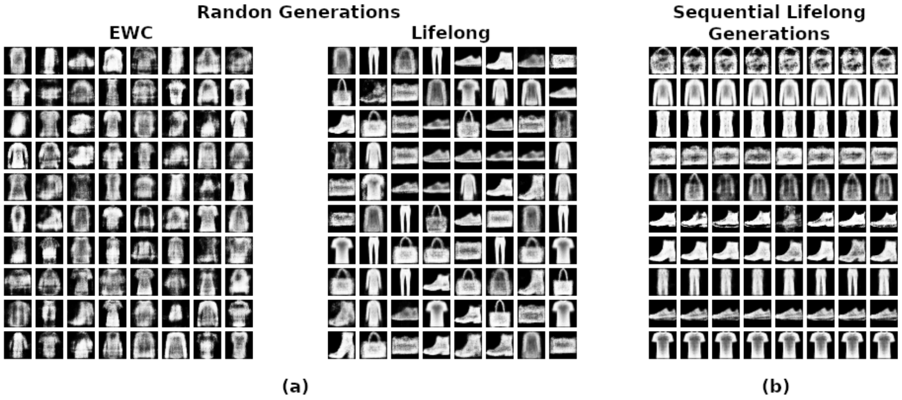


Figure 5: Generations of models after training with 10 distributions of FashionMNIST; (a) Random generations of EWC VAE vs. Lifelong VAE. (b) Sequential generation of Lifelong VAE; left to right are generated by fixing $\boldsymbol{z}_d$ and randomly sampling $\boldsymbol{z}_c$; top to bottom uniformly varies $\boldsymbol{z}_d$. The EWC VAE does not generate meaningful sequential generations.

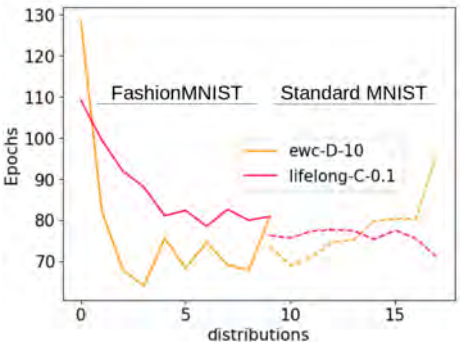
We observe that our *lifelong-vae* model does at least as well as EWC with respect to the test ELBO (Figure 3a), while outperforming it with respect to the *log*-Frechet distance (Figure 3b). This is further validated by respectively visualizing image reconstructions (Figure 3(a)) and image generations (Figure 5). We surmise the improvement with regards to sample generation is because our model can generate distinct, high quality samples (Figure 5) while avoiding mixing or under-sampling due to the joint interaction of the consistency regularizer (Section 4.1) and information gain regularizer (Section 4.3).

Even though each trial is performed five times, we observe large increases in terms of the negative ELBO and Frechet distance at distribution five and seven. This is attributed to the models being exposed to a drastically different distribution: specifically the “sandals” and “sneakers” distributions. Since these distributions are vastly different from previous distributions such as “T-shirt”, it poses challenging for the models to accomodate the new varied representation.

Finally, we evaluate the number of epochs needed to train both an *ewc* and a *lifelong* model in Figure 3(c). We quantify the convergence time as the number of epochs it takes a model to trigger an early stopping criterion on the validation set. If a model converges faster, we surmise that it is efficiently using information from previous learning. We observe that the *ewc* model initially converges faster than our *lifelong* model, but does so at the expense of significantly worse sample generation (Figure 3b, Figure 5a-left), minimizing it's usefulness in a lifelong generative setting. The *lifelong* model consecutively requires fewer and fewer epochs for convergence and finally reaches the same number of epochs as *ewc*. We extend this setting to a much larger set of similar distributions in Section 5.1.1 and observe that our model does outperform *ewc* in such scenarios. This confirms the benefits our new objective formulation brings to the lifelong setting, where previous knowledge is retained and used in future learning.

### *5.1.1. Extending Number Of Sequential Distributions*

In order to observe how our *lifelong* model compares with *ewc* with regards to training epoch convergence time, we extended Experiment 5.1 to a set of 20 distributions. The first ten observed distributions are the FashionMNIST objects as in Experiment 5.1 and the latter ten are the MNIST digits separated into their individual digit-distributions. We observe that in this setting our *lifelong* model outperforms the *ewc* model as demonstrated on the left.



The analysis of convergence time based on an early stopping criterion on the negative ELBO is generally noisy due to varying sources such as the noise introduced from minibatch sampling. However, we observe an average decrease in the number of epochs required for the *lifelong* model in contrast to the *ewc* model. We also note that model convergence does not guarentee a good solution and this is evident from the generations of EWC from Experiment 5.1.

### *5.2. Lifelong Learning of Different Distributions*

In this experiment we examine the capability of our model to generate and recall completely different distributions. This setting differs from Experiment 5.1 in that the models cannot leverage previously learnt feature representations for future learning. We apply a set of unique fixed image permutations ($G = \{G_1, ...G_N\}$ where $\forall i, j \in \{1..N\}, i \neq j,\ \ G_i \neq G_j$) to the entire MNIST dataset. We create 5 such datasets $\{\boldsymbol{X}, G_1\boldsymbol{X}, ...G_4\boldsymbol{X}\}$ and sequentially progress over them in a similar manner as Experiment 5.1. We use an unpermuted version of the MNIST dataset $\boldsymbol{X}$ to simulate the first distribution $P_1^*(\mathbf{x})$ as it allows us to visually asses the degradation of reconstructions. This is a common setup utilized in continual learning [31, 32] and we extend it here to the generative setting.

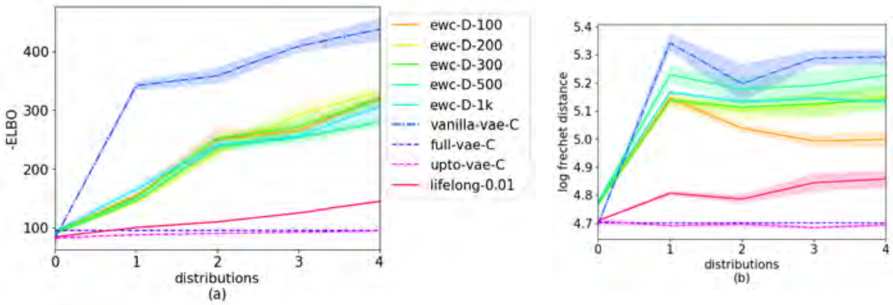


Figure 6: PermutedMNIST Experiment: (a) Negative Test ELBO (b) *log*-Frechet Distance

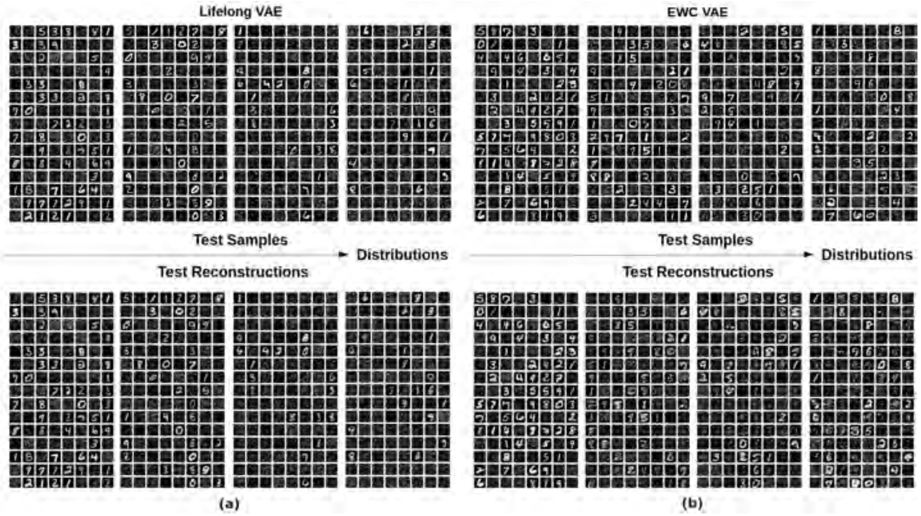


Figure 7: Top row: test-samples; bottom row: reconstructions. We visualize an increasing number of accumulated distributions from left to right. (a) Lifelong VAE model (b) EWC VAE model

EWC works well when the learnt parameters of the old task are relevant for the learning of the new task. In this experiment however, EWC is forced to accommodate the new task, while still preserving the parameters learnt over a drastically different old task. This poses a challenge for the restrictive EWC distance metric (Section 4.2). When Permuted MNIST is used in a classification setting [31] observe a small degradation in accuracy over time, however we observe a much more pronounced effect in the generative setting.

The *lifelong* implementation on the other hand allows the model to flexibly adapt it's distance metric (Section 4.2) in order to learn an appropriate constraint for preserving both the current distribution $P_i^*(\mathbf{x})$ as well as the previous distributions $P_{<i}^*(\mathbf{x})$. This is due to the fact that we constrain the latent posterior distribution (Section 4.1) and keep the conditional $P_{\boldsymbol{\theta}}(\hat{\mathbf{x}}|\boldsymbol{z}_d, \boldsymbol{z}_c)$ similar to that of the previous task through the data augmentation step rather than simply constraining by a simple quadratic parameter difference. In these ex-

periments we see the *lifelong* model outperform all other models (barring the *upto-vae* and *full-vae* which present the best attainable performance) in terms of both reconstructions (Figure 6a) and generations (Figure 6b).

### 5.3. SVHN to MNIST

In this experiment we explore the ability of our model to retain and transfer knowledge across completely different datasets. We use MNIST and SVHN [49] to demonstrate this. We treat all samples from SVHN as being generated by one distribution $P_1^*(\mathbf{x})$ and all the MNIST[11] samples as generated by another distribution $P_2^*(\mathbf{x})$ (irrespective of the specific digit).

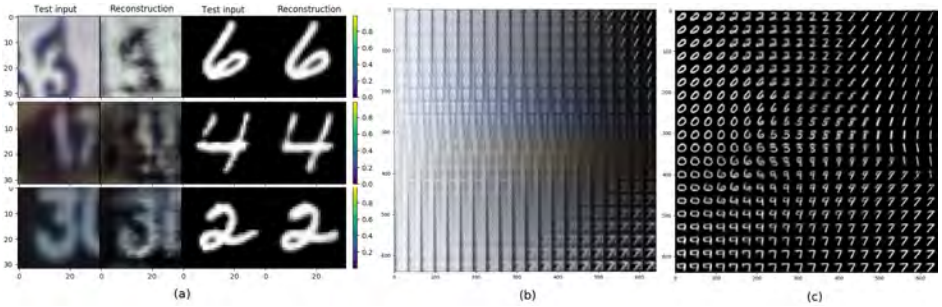


Figure 8: (a) Reconstructions of test samples from SVHN[left] and MNIST[right]; (b) Decoded samples $\hat{\mathbf{x}} \sim P_{\boldsymbol{\theta}}(\mathbf{x}|\boldsymbol{z}_d, \boldsymbol{z}_c)$ based on linear interpolation of $\boldsymbol{z}_c \in \mathcal{R}^2$ with $\boldsymbol{z}_d = [0, 1]$; (c) Same as (b) but with $\boldsymbol{z}_d = [1, 0]$.

We visualise examples of the true inputs $\mathbf{x}$ and the respective reconstructions $\hat{\mathbf{x}}$ in figure 8(a). We see that even though the only true data the final model received for training were from MNIST, it can still reconstruct SVHN data observed previously. This confirms the ability of our architecture to transition between complex distributions while still preserving the knowledge learned from the previously observed distributions. Finally, in figure 8(b) and 8(c) we illus-

[11] MNIST was resized to 32x32 and converted to RBG to make it consistent with the dimensions of SVHN.

trate the data generated from an interpolation of a 2-dimensional continuous latent space. For this we specifically trained the models with the continuous latent variable $\boldsymbol{z}_c \in \mathcal{R}^2$. To generate the data, we fix the discrete categorical $\boldsymbol{z}_d$ to one of the possible values $\{[0,1],[1,0]\}$ and linearly interpolate the continuous $\boldsymbol{z}_c$ over the range $[-3,3]$. We then decode these to obtain the samples $\hat{\mathbf{x}} \sim P_{\boldsymbol{\theta}}(\mathbf{x}|\boldsymbol{z}_d,\boldsymbol{z}_c)$. The model learns a common continuous structure for the two distributions which can be followed by observing the development in the generated samples from top left to bottom right on both figure 8(b) and 8(c).

### *5.4. Celeb-A Sequential Generation*

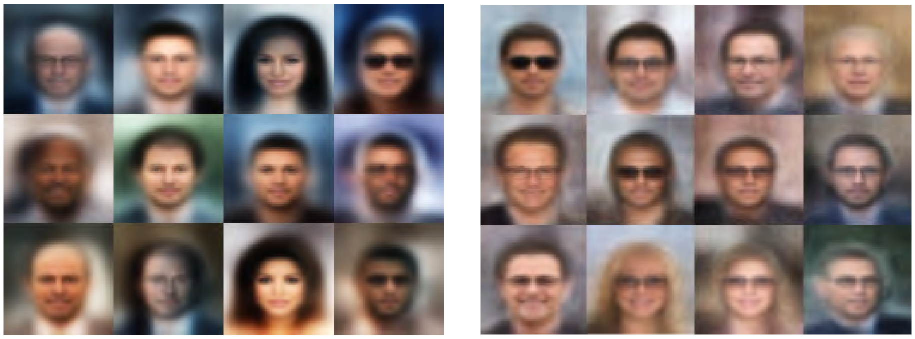
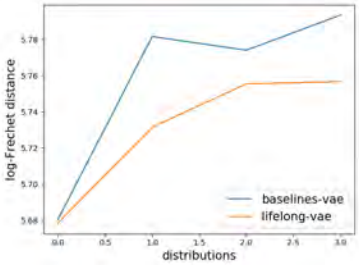


Figure 9: *Left*: Sequential generations for Celeb-A for the *lifelong* model for *bald*, *male*, *young* and *eye-glasses* (left to right). *Right*: (random) generations by the baseline VAE model.

In this experiment we split Celeb-A into four individual distributions using the features: *bald*, *male*, *young* and *eye-glasses*. As with the previous experiments, we treat each subset of data as an individual distribution, $P_i^*(\mathbf{x})$, $i \in \{1..4\}$, and present our model samples from a single distribution at a time. This presents a real world scenario as the samples per distribution varies drastically from only 3,713 samples for the *bald* distribution, to 126,788 samples for *young*. In addition specific samples can span one or more of these distributions.



| 44,218 *male* samples | baseline-VAE | Lifelong |
|---|---|---|
| training-epoch (s) | 43.1 +/- 0.6 | 56.63 +/- 0.28 |
| testing-epoch (s) | 9.79 +/- 0.12 | 16.09 +/- 0.01 |

Table 1: Mean & standard deviation wall-clock for one epoch of *male* distribution of Celeb-A.

Figure 10: Celeb-A *log*-Frechet distance of *lifelong* vs. *baseline* model over the four distributions. Listed on the right is the time per epoch (in seconds) for an epoch of the corresponding models.

We train a *lifelong* model and a *baseline* (typical VAE) model and evaluate their generations in Figure 9. As visually demonstrated in Figure 9-*Left*, the *lifelong* model is able to generate instances from all of the previous distributions, however the *baseline* model catastrophically forgets (Figure 9-*Right*) and only generates samples from the *eye-glasses* distribution. This is also reinforced by the *log*-Frechet distance shown in Figure 10. In general, VAEs produce blurry images, however due to the incorporation of the information restricting regularizer (Section 4.4) and the latent variable decomposition (Section 4.3, Experiment 6.1) we observe higher quality generations than typical VAEs. [12]

We also evaluate the wall-clock time in seconds (Table 1) for the *lifelong* model and the *baseline-vae* for the 44,218 samples of the *male* distribution. We observe that the *lifelong* model does not add a significant overhead, especially since the *baseline-vae* undergoes catastrophic interference (Figure 9 *Right*) and completely fails to generate samples from previous distributions. Note that we present the number of parameters and other *detailed model information* in our code and Appendix 9.3.

[12] The *baseline* model also utilizes the discrete and continuous variable decomposition as in the previous experiments.

# 6. Ablation Studies

In this section we independently validate the benefit of each of the newly introduced components to the learning objective proposed in Section 4.6. In Experiment 6.1 we validate the benefit of the discrete-continuous posterior decoupling introduced in Section 4.3. We achieve this by validating the linear separability of the latent representation in contrast to various different posterior re-parameterizations. Finally, in Experiment 6.2 we validate the benefit of the information restricting regularizer introduced in Section 4.4 by examining a learning scenario (with and without the proposed regularizer) similar to Experiment 5.1.

## *6.1. Linear Separability of Discrete and Continuous Posterior*

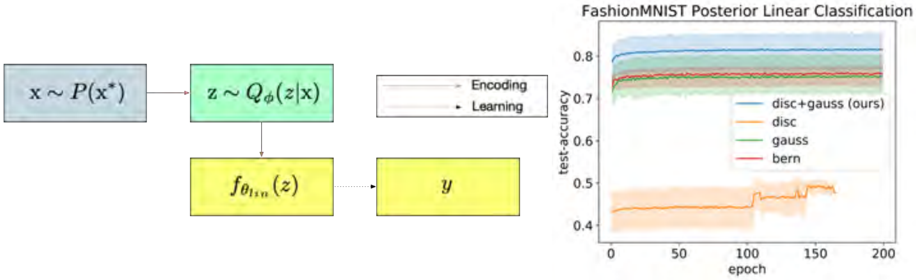


Figure 11: *Left:* Graphical model depicting classification using pretrained VAE, coupled with a linear classifier, $f_{\boldsymbol{\theta}_{lin}} : \boldsymbol{z} \mapsto \boldsymbol{y}$. *Right:* Linear classifier accuracy on the Fashion MNIST test set for a varying range of latent dimensions, $|\boldsymbol{z}| \in [32, 64, 128, 256, 512, 1024]$ and distributions.

In order to validate that the introduction of the (independent) discrete and continuous latent variable posterior, $Q_{\boldsymbol{\phi}}(\boldsymbol{z}_d, \boldsymbol{z}_c|\mathbf{x})$, aids in learning a better disentangled representation, we perform an experiment where we classify the encoded posterior sample using a simple linear classifier $f_{\boldsymbol{\theta}_{lin}} : \boldsymbol{z} \mapsto \boldsymbol{y}$, where $\boldsymbol{y}$ correponds to the categorical class prediction. Higher (linear) classification accuracies demonstrate that the the VAE was able to learn a better disentangled representation. This is a standard method used to measure posterior disentanglement and is used in methods such as Associative Compression Networks [50].

We use the standard training set of FashionMNIST [48] (60,000 samples) to train a standard VAE with a discrete only (*disc*) posterior, an isotropic-gaussian only (*gauss*) posterior, a bernoulli only (*bern*) posterior and finally the proposed independent discrete and continuous (*disc+gauss*) posterior presented in Section 4.3. For each different posterior reparameterization, we train a set of VAEs with varying latent dimensions, $|\boldsymbol{z}| \in [32, 64, 128, 256, 512, 1024]$. In the case of the *disc+gauss* model we fix the discrete dimension, $|\boldsymbol{z}_d| = 10$ and vary the isotropic-gaussian dimension to match the total required dimension. After training each VAE, we proceed to use the same training data to train a linear classifier on the encoded posterior sample, $\boldsymbol{z} \sim Q_{\boldsymbol{\phi}}(\boldsymbol{z}|\mathbf{x})$.

In Figure 11 we present the mean and standard deviation linear test classification accuracies of each set of the different experiments. As expected, the discrete only (*disc*) posterior performs poorly due to the strong restriction of mapping an entire input sample to a single one-hot vector. The isotropic-gaussian (*gauss*) and bernoulli (*bern*) only models provide a strong baseline, but the combination of isotropic-gaussian and discrete posteriors (*disc+gauss*) performs much better, reaching an upper-bound (linear) test-classification accuracy of **87.1%**. This validates that the decoupling of latent represention presented in Section 4.3 aids in learning a more meaningful, disentangled posterior.

### *6.2. Validating the Information Restricting and Consistency Regularizers*

In order to independently evaluate the benefit of our proposed Bayesian update regularizer (Section 4.1) and the negative information gain term proposed in (Section 4.3) we perform a simple ablation study examining the Frechet distance over a sequence of distributions. We also visualize sequential generations from the final student model as in the previous experiment. We utilize the MNIST dataset instead of Fashion MNIST in order to provide experiment diversity. The dataset is divided and iterated over as in Experiment 5.1.

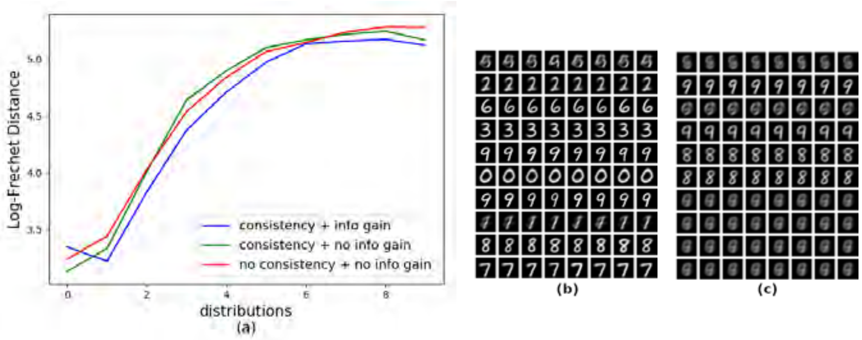


Figure 12: MNIST Ablation: (a) *log*-Frechet Distance; (b) Generating samples by fixing $\boldsymbol{z}_d$ per row and varying $\boldsymbol{z}_c$ per column for the model with both regularizers. (c) Generating samples by fixing $\boldsymbol{z}_d$ per row and varying $\boldsymbol{z}_c$ per column for the model without the information restricting regularizer.

In contrast to Experiment 5.1, we evaluate three scenarios: 1) with consistency and mutual information regularizers, 2) only consistency regularizer and 3) without both regularizers. For this experiment we also fix the seed used by pytorch [51] and numpy [52] such that the effects of initialization and dataset shuffling are non-existent [13].

We observe that both components are necessary in order to generate high quality samples as evidenced by the **log**-Frechet distance (Figure 12a) [14]. The generations produced without the information gain regularizer (Figure 12c) are blurry for all but the last two observed distributions (eight and nine in this case). This can be attributed to two possibilities: : 1) uniformly sampling the discrete component is not guaranteed to generate samples from $P_{\boldsymbol{\theta}}(\mathbf{x}_{<i})$, one of the unique, previously approximated distributions (see mixing issue in Figure 1b) and 2) the decoder $P_{\boldsymbol{\Theta}}(\hat{\mathbf{x}}|\boldsymbol{z}_d, \boldsymbol{z}_c)$ leverages more information from the continuous component, i.e. $P_{\boldsymbol{\Theta}}(\hat{\mathbf{x}}|\boldsymbol{z}_d, \boldsymbol{z}_c) = P_{\boldsymbol{\Theta}}(\hat{\mathbf{x}}|\boldsymbol{z}_c)$, causing catastrophic forgetting and posterior collapse [53].

[13] We only run a single experiment here since multiple trials produce the same solution.

[14] We observe that this effect gets more pronounced when the dimensionality of $\boldsymbol{z}_c$ is increased.

## 7. Limitations

Throughout this work we demonstrate that our method performs as well as state-of-the-art methods (Experiment 5.1), or vastly outperforms them (Experiment 5.2). However, while our model alleviates the catastrophic interference problem, it fails to completely solve it and we see a slow degradation in model performance over time. We attribute this mainly to the problem of poor VAE generations that compound upon each other (also discussed below). In addition, there are a few poignant issues that need to be resolved in order to achieve an optimal (in terms of non-degrading Frechet distance / -ELBO) unsupervised generative lifelong learner:

**Distribution Boundary Evaluation**: The standard assumption in current lifelong / continual learning approaches [33, 32, 29, 30, 31, 27] is to use known, fixed distributions instead of learning the distribution transition boundaries. For the purposes of this work, we focus on the accumulation of distributions (in an unsupervised way), rather than introduce an additional level of indirection through the incorporation of anomaly detection methods that aid in detecting distributional boundaries.

**Blurry VAE Generations**: VAEs are known to generate images that are blurry in contrast to GAN based methods. This has been attributed to the fact that VAEs don't learn the true posterior and make a simplistic assumption regarding the reconstruction distribution $P_{\boldsymbol{\theta}}(\mathbf{x}|\boldsymbol{z})$ [53, 54]. While there exist methods such as ALI [55] and BiGAN [56], that learn a posterior distribution within the GAN framework, recent work has shown that adversarial methods fail to accurately match posterior-prior distribution ratios in large dimensions [57].

**Memory**: In order to scale to a truly lifelong setting, we posit that a learning algorithm needs a global pool of memory that can be decoupled from the learning algorithm itself. This decoupling would also allow for a principled mechanism for parameter transfer between sequentially learnt models as well a centralized location for compressing non-essential historical data. Recent work such as the Kanerva Machine [58] and its extensions [59] provide a principled way to do this in the VAE setting.

## 8. Conclusion

In this work we propose a novel method for learning generative models over a lifelong setting. The principal assumption for the data is that they are generated by multiple distributions and presented to the learner in a sequential manner. A key *limitation* for the learning process is that the method *has no access to any of the old data and that it shall distill all the necessary information into a single final model*. The proposed method is based on a dual student-teacher architecture where the teacher's role is to preserve the past knowledge and aid the student in future learning. We argue for and augment the standard VAE's ELBO objective by terms helping the teacher-student knowledge transfer. We demonstrate the benefits this augmented objective brings to the lifelong learning setting using a series of experiments. The architecture, combined with the proposed regularizers, aid in mitigating the effects of catastrophic interference by supporting the retention of previously learned knowledge.

## References

[1] S. Thrun, T. M. Mitchell, Lifelong robot learning, in: The biology and technology of intelligent autonomous agents, Springer, 1995, pp. 165–196.

[2] S. Thrun, Lifelong learning: A case study., Tech. rep., CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER SCIENCE (1995).

[3] D. P. Kingma, M. Welling, Auto-encoding variational bayes, ICLR.

[4] J. M. Wolterink, T. Leiner, M. A. Viergever, I. Išgum, Generative adversarial networks for noise reduction in low-dose ct, IEEE transactions on medical imaging 36 (12) (2017) 2536–2545.

[5] D. Ulyanov, A. Vedaldi, V. Lempitsky, Deep image prior, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 9446–9454.

[6] R. A. Yeh, C. Chen, T. Yian Lim, A. G. Schwing, M. Hasegawa-Johnson, M. N. Do, Semantic image inpainting with deep generative models, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5485–5493.

[7] W. Wang, Q. Huang, S. You, C. Yang, U. Neumann, Shape inpainting using 3d generative adversarial network and recurrent convolutional networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2298–2306.

[8] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al., Photo-realistic single image super-resolution using a generative adversarial network, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4681–4690.

[9] K. Sohn, H. Lee, X. Yan, Learning structured output representation using deep conditional generative models, in: Advances in neural information processing systems, 2015, pp. 3483–3491.

[10] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, B. Frey, Adversarial autoencoders, arXiv preprint arXiv:1511.05644.

[11] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, arXiv preprint arXiv:1511.06434.

[12] Y. LeCun, C. Cortes, MNIST handwritten digit database [cited 2016-01-14 14:24:11].
URL `http://yann.lecun.com/exdb/mnist/`

[13] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, stat 1050 (2015) 9.

[14] T. Furlanello, J. Zhao, A. M. Saxe, L. Itti, B. S. Tjan, Active long term memory networks, arXiv preprint arXiv:1606.02355.

[15] M. McCloskey, N. J. Cohen, Catastrophic interference in connectionist networks: The sequential learning problem, Psychology of learning and motivation 24 (1989) 109–165.

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in neural information processing systems, 2014, pp. 2672–2680.

[17] H. Kim, A. Mnih, Disentangling by factorising, in: International Conference on Machine Learning, 2018, pp. 2654–2663.

[18] F. A. Quintana, P. L. Iglesias, Bayesian clustering and product partition models, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 65 (2) (2003) 557–574.

[19] K. O. Perlmutter, S. M. Perlmutter, R. M. Gray, R. A. Olshen, K. L. Oehler, Bayes risk weighted vector quantization with posterior estimation for image compression and classification, IEEE Transactions on Image Processing 5 (2) (1996) 347–360.

[20] L. Fe-Fei, et al., A bayesian approach to unsupervised one-shot learning of object categories, in: Proceedings Ninth IEEE International Conference on Computer Vision, IEEE, 2003, pp. 1134–1141.

[21] E. Dupont, Learning disentangled joint continuous and discrete representations, in: Advances in Neural Information Processing Systems, 2018, pp. 708–718.

[22] C. Louizos, K. Swersky, Y. Li, M. Welling, R. Zemel, The variational fair autoencoder, ICLR.

[23] T. M. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, J. Betteridge, A. Carlson, B. D. Mishra, M. Gardner, B. Kisiel, J. Krishnamurthy, et al., Never-ending learning, in: Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.

[24] T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, B. Yang, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, et al., Never-ending learning, Communications of the ACM 61 (5) (2018) 103–115.

[25] D. L. Silver, Q. Yang, L. Li, Lifelong machine learning systems: Beyond learning algorithms, in: 2013 AAAI spring symposium series, 2013.

[26] Z. Chen, B. Liu, Lifelong machine learning, Synthesis Lectures on Artificial Intelligence and Machine Learning 10 (3) (2016) 1–145.

[27] N. C. Rabinowitz, G. Desjardins, A.-A. Rusu, K. Kavukcuoglu, R. T. Hadsell, R. Pascanu, J. Kirkpatrick, H. J. Soyer, Progressive neural networks, uS Patent App. 15/396,319 (Nov. 23 2017).

[28] A. V. Terekhov, G. Montone, J. K. O'Regan, Knowledge transfer in deep block-modular neural networks, in: Proceedings of the 4th International Conference on Biomimetic and Biohybrid Systems-Volume 9222, Springer-Verlag New York, Inc., 2015, pp. 268–279.

[29] H. Shin, J. K. Lee, J. Kim, J. Kim, Continual learning with deep generative replay, in: Advances in Neural Information Processing Systems, 2017, pp. 2994–3003.

[30] N. Kamra, U. Gupta, Y. Liu, Deep generative dual memory network for continual learning, arXiv preprint arXiv:1710.10368.

[31] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al., Overcoming catastrophic forgetting in neural networks, Proceedings of the National Academy of Sciences (2017) 201611835.

[32] F. Zenke, B. Poole, S. Ganguli, Continual learning through synaptic intelligence, in: International Conference on Machine Learning, 2017, pp. 3987–3995.

[33] C. V. Nguyen, Y. Li, T. D. Bui, R. E. Turner, Variational continual learning, ICLR.

[34] R. M. Neal, Bayesian learning for neural networks, Ph.D. thesis, University of Toronto (1995).

[35] C. Blundell, J. Cornebise, K. Kavukcuoglu, D. Wierstra, Weight uncertainty in neural network, in: International Conference on Machine Learning, 2015, pp. 1613–1622.

[36] Z. Li, D. Hoiem, Learning without forgetting, in: European Conference on Computer Vision, Springer, 2016, pp. 614–629.

[37] D. P. Kingma, ”variational inference & deep learning: A new synthesis”, Ph.D. thesis (2017).

[38] S. Gershman, N. Goodman, Amortized inference in probabilistic reasoning, in: Proceedings of the Cognitive Science Society, Vol. 36, 2014.

[39] T. Broderick, N. Boyd, A. Wibisono, A. C. Wilson, M. I. Jordan, Streaming variational bayes, in: C. J. C. Burges, L. Bottou, Z. Ghahramani, K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States., 2013, pp. 1727–1735.
URL http://papers.nips.cc/paper/4980-streaming-variational-bayes

[40] I. Katakis, G. Tsoumakas, I. Vlahavas, Incremental clustering for the classification of concept-drifting data streams.

[41] R. Gomes, M. Welling, P. Perona, Incremental learning of nonparametric bayesian mixture models, in: 2008 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2008, pp. 1–8.

[42] L. Bottou, Online learning and stochastic approximations, On-line learning in neural networks 17 (9) (1998) 142.

[43] V. Jain, E. Learned-Miller, Online domain adaptation of a pre-trained cascade of classifiers, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2011, pp. 577–584.

[44] X. Chen, X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, P. Abbeel, Infogan: Interpretable representation learning by information maximizing generative adversarial nets, in: D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, R. Garnett (Eds.), Advances in Neural Information Processing Systems 29, Curran Associates, Inc., 2016, pp. 2172–2180.

[45] F. Huszar, Infogan: using the variational bound on mutual information (twice) (Aug 2016).
URL https://www.inference.vc/infogan-variational-bound-on-mutual-information-twice/

[46] A. Achille, T. Eccles, L. Matthey, C. Burgess, N. Watters, A. Lerchner, I. Higgins, Life-long disentangled representation learning with cross-domain latent homologies, in: Advances in Neural Information Processing Systems, 2018, pp. 9895–9905.

[47] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, S. Hochreiter, Gans trained by a two time-scale update rule converge to a local nash equilibrium, in: Advances in Neural Information Processing Systems, 2017, pp. 6629–6640.

[48] H. Xiao, K. Rasul, R. Vollgraf, Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, arXiv preprint arXiv:1708.07747.

[49] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Y. Ng, Reading digits in natural images with unsupervised feature learning, in: NIPS workshop on deep learning and unsupervised feature learning, 2011, p. 5.

[50] A. Graves, J. Menick, A. v. d. Oord, Associative compression networks, arXiv preprint arXiv:1804.02476.

[51] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in pytorch, in: NIPS-W, 2017.

[52] D. Ascher, P. F. Dubois, K. Hinsen, J. Hugunin, T. Oliphant, Numerical Python, Lawrence Livermore National Laboratory, Livermore, CA, ucrl-ma-128569 Edition (1999).

[53] A. Alemi, B. Poole, I. Fischer, J. Dillon, R. A. Saurous, K. Murphy, Fixing a broken elbo, in: International Conference on Machine Learning, 2018, pp. 159–168.

[54] T. Rainforth, A. Kosiorek, T. A. Le, C. Maddison, M. Igl, F. Wood, Y. W. Teh, Tighter variational bounds are not necessarily better, in: International Conference on Machine Learning, 2018, pp. 4277–4285.

[55] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, A. Courville, Adversarially learned inference, arXiv preprint arXiv:1606.00704.

[56] J. Donahue, P. Krähenbühl, T. Darrell, Adversarial feature learning, arXiv preprint arXiv:1605.09782.

[57] M. Rosca, B. Lakshminarayanan, S. Mohamed, Distribution matching in variational inference, arXiv preprint arXiv:1802.06847.

[58] Y. Wu, G. Wayne, A. Graves, T. Lillicrap, The kanerva machine: A generative distributed memory, ICLR.

[59] Y. Wu, G. Wayne, K. Gregor, T. Lillicrap, Learning attractor dynamics for generative memory, in: Advances in Neural Information Processing Systems, 2018, pp. 9401–9410.

[60] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, arXiv preprint arXiv:1502.03167.

[61] D. P. Kingma, J. L. Ba, Adam: A method for stochastic optimization.

[62] M. D. Hoffman, D. M. Blei, C. Wang, J. Paisley, Stochastic variational inference, The Journal of Machine Learning Research 14 (1) (2013) 1303–1347.

[63] C. J. Maddison, A. Mnih, Y. W. Teh, The concrete distribution: A continuous relaxation of discrete random variables, arXiv preprint arXiv:1611.00712.

[64] E. Jang, S. Gu, B. Poole, Categorical reparameterization with gumbel-softmax, International Conference on Learning Representations.

[65] E. D. Sontag, Vc dimension of neural networks, NATO ASI Series F Computer and Systems Sciences 168 (1998) 69–96.

[66] M. Karpinski, A. Macintyre, Polynomial bounds for vc dimension of sigmoidal and general pfaffian neural networks, Journal of Computer and System Sciences 54 (1) (1997) 169–176.

[67] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks., in: Aistats, Vol. 9, 2010, pp. 249–256.

[68] K. P. Murphy, Machine learning: a probabilistic perspective, MIT press, 2012.

## 9. Appendix

### *9.1. Understanding the Consistency Regularizer*

The analytical derivations of the consistency regularizer show that the regularizer can be interpreted as an a transformation of the standard VAE regularizer. In the case of an isotropic gaussian posterior, the proposed regularizer scales the mean and variance of the student posterior by the variance of the teacher 1 and adds an extra 'volume' term. This interpretation of the consistency regularizer shows that the proposed regularizer preserves the same learning objective as that of the standard VAE. Below we present the analytical form of the consistency regularizer with categorical and isotropic gaussian posteriors:

**Proof 1.** *We assume the learnt posterior of the teacher is parameterized by a centered, isotropic gaussian with* $\boldsymbol{\Phi} = [\boldsymbol{\mu^E = 0}, \boldsymbol{\Sigma^E} = diag(\boldsymbol{\sigma^{E^2}})]$ *and the posterior of our student by a non-centered isotropic gaussian with* $\boldsymbol{\phi} = [\boldsymbol{\mu^S}, \boldsymbol{\Sigma^S} = diag(\boldsymbol{\sigma^{S2}})]$*, then*

$$\begin{aligned} KL(Q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})||Q_{\boldsymbol{\Phi}}(\boldsymbol{z}|\boldsymbol{x})) &= 0.5\left[tr(\boldsymbol{\Sigma}^{E^{-1}}\boldsymbol{\Sigma^S}) + (\boldsymbol{\mu^E} - \boldsymbol{\mu^S})^T\boldsymbol{\Sigma}^{E^{-1}}(\boldsymbol{\mu^E} - \boldsymbol{\mu^S}) - F + log\left(\frac{|\boldsymbol{\Sigma^E}|}{|\boldsymbol{\Sigma^S}|}\right)\right] \\ &= 0.5\sum_{j=1}^{F}\left[\frac{1}{\sigma^{E2}(j)}(\sigma^{S2}(j) + \mu^{S2}(j)) - 1 + log\ \sigma^{E2}(j) - log\ \sigma^{S2}(j)\right] \\ &= KL(Q_{\boldsymbol{\phi}^*}(\boldsymbol{z}|\boldsymbol{x})||\mathcal{N}(0, \boldsymbol{I})) - log\ |\boldsymbol{\Sigma}^E| \end{aligned} \tag{11}$$

*Via a reparameterization of the student's parameters:*

$$\begin{aligned} \boldsymbol{\phi}^* &= [\boldsymbol{\mu}^{S*}, \boldsymbol{\sigma}^{S*2}] \\ \boldsymbol{\mu}^{S*} = \frac{\mu^S(j)}{\sigma^{E2}(j)}&; \boldsymbol{\sigma}^{S*2} = \frac{\sigma^{S2}(j)}{\sigma^{E2}(j)} \end{aligned} \tag{12}$$

It is also interesting to note that our posterior regularizer becomes the prior if:

$$lim_{\boldsymbol{\sigma}^{E2}\mapsto 1}KL(Q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})||Q_{\boldsymbol{\Phi}}(\boldsymbol{z}|\boldsymbol{x})) = KL(Q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})||\mathcal{N}(0, \boldsymbol{I}))$$

**Proof 2.** *We parameterize the learnt posterior of the teacher by* $\Phi_i = \frac{\exp(p_i^E)}{\sum_{i=1}^{J}\exp(p_i^E)}$ *and the posterior of the student by* $\phi_i = \frac{\exp(p_i^S)}{\sum_{i=1}^{J}\exp(p_i^S)}$*. We also redefine the* *normalizing constants as* $c^E = \sum_{i=1}^{J} \exp(p_i^E)$ *and* $c^S = \sum_{i=1}^{J} \exp(p_i^S)$ *for the teacher and student models respectively. The reverse KL divergence in equation 15 can now be re-written as:*

$$\begin{aligned} KL(Q_{\boldsymbol{\phi}}(\boldsymbol{z}_d|\boldsymbol{x})||Q_{\boldsymbol{\Phi}}(\boldsymbol{z}_d|\boldsymbol{x})) &= \sum_{i=1}^{J} \frac{\exp(p_i^S)}{c^S} log \left( \frac{\exp(p_i^S)}{c^S} \frac{c^E}{\exp(p_i^E)} \right) \\ &= H(\boldsymbol{p}^S, \boldsymbol{p}^S - \boldsymbol{p}^E) = -H(\boldsymbol{p}^s) + H(\boldsymbol{p}^S, \boldsymbol{p}^E) \end{aligned} \quad (13)$$

*where* $H(\_)$ *is the entropy operator and* $H(\_,\_)$ *is the cross-entropy operator.*

### *9.2. Reconstruction Regularizer*

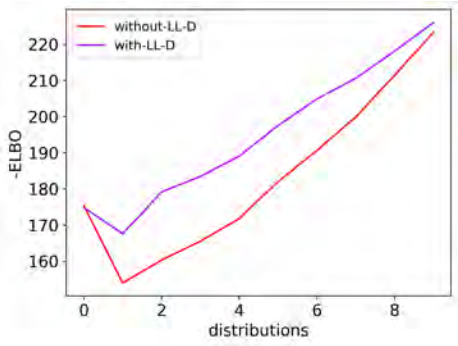


Figure 13: Fashion Negative Test ELBO

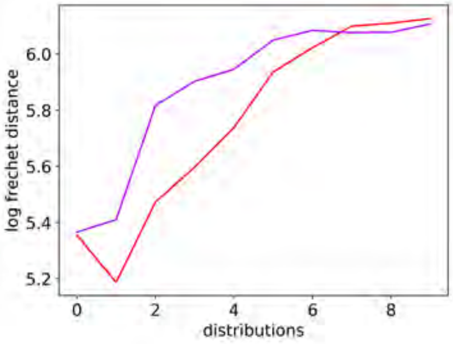


Figure 14: Fashion Log-Frechet Distance

It is also possible to constrain the reconstruction term of the VAE in a similar manner to the consistency posterior-regularizer, i.e: $KL[P_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{<i}|\boldsymbol{z})||P_{\boldsymbol{\Theta}}(\hat{\mathbf{x}}_{<i}|\boldsymbol{z})]$, however this results in diminished model performance. We hypothesize that this is due to the fact that this regularizer contradicts the objective of the reconstruction term $P_{\boldsymbol{\theta}}(\mathbf{x}|\boldsymbol{z})$ in the ELBO which already aims to minimize some metric between the input samples $\mathbf{x}$ and the reconstructed samples $\hat{\mathbf{x}}$; eg: if $P_{\boldsymbol{\theta}}(\mathbf{x}|\boldsymbol{z}) \sim \mathcal{N}(\boldsymbol{\mu}, \text{diag}[\boldsymbol{\sigma}])$, then the loss is proportional to $||\hat{\mathbf{x}} - \mathbf{x}||_2^2$, the standard L2 loss. Without the addition of this reconstruction cross-model regularizer, the model is also provided with more flexibility in how it reconstructs the output samples.

In order to quantify the this we run Experiment 5.1 utilizing two dense models (-D): one with the consistency regularizer (*without-LL-D*) and one with the consistency and likelihood regularizer (*with-LL-D*). We observe the model performance drop (with respect to the Frechet distance as well the test ELBO) in the case of the *with-LL-D* as demonstrated in Figures 13 and 14.

### *9.3. Model Architecture*

We utilized two different architectures for our experiments. When we utilize a dense network (-D- in experiments) we used two layers of 512 to map to the latent representation and two layers of 512 to map back to the reconstruction for the decoder. We used batch norm [60] and ELU (and sometimes SeLU) activations for all the layers barring the layer projecting into the latent representation and the output layer. Note that while we used the same architecture for EWC we observed a drastic negative effect when using batch norm and thus dropped it's usage. The convolution architectures used the architecture described below for the encoder and the decoder (where the decoder used conv-transpose layers for upsampling). The notation is [OutputChannels, (filterX, filterY), stride]:

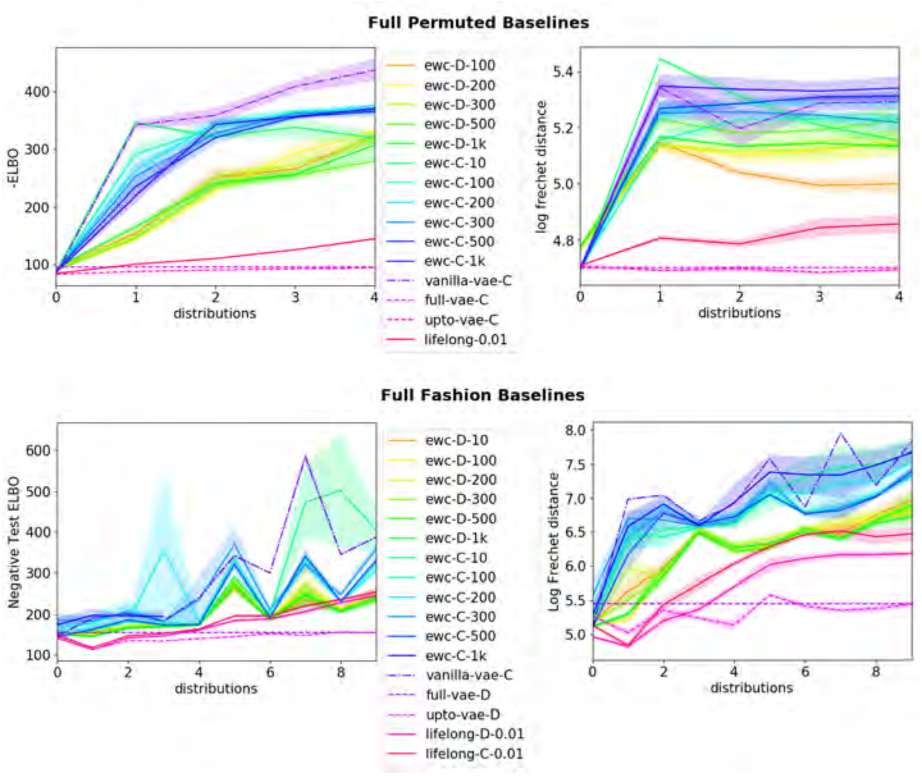
$$\begin{aligned}
\text{Encoder: } & [32,(5,5),1] \mapsto \text{GN+ELU} \mapsto [64,(4,4),2] \mapsto \text{GN+ELU} \mapsto [128,(4,4),1] \mapsto \\
& \text{GN+ELU} \mapsto [256,(4,4),2] \mapsto \text{GN+ELU} \mapsto [512,(1,1),1] \mapsto \\
& \text{GN+ELU} \mapsto [512,(1,1),1] \\
\text{Decoder: } & [256,(4,4),1] \mapsto \text{GN+ELU} \mapsto [128,(4,4),2] \mapsto \text{GN+ELU} \mapsto [64,(4,4),1] \\
& \mapsto \text{GN+ELU} \mapsto [32,(4,4),2] \mapsto \text{GN+ELU} \mapsto [32,(5,5),1] \\
& \mapsto \text{GN+ELU} \mapsto [\text{chans},(1,1),1]
\end{aligned} \tag{14}$$

| Method | Initial $\boldsymbol{z}_d$ dimension | Final $\boldsymbol{z}_d$ dimension | $\boldsymbol{z}_c$ dimension | # initial parameters | # final parameters |
|---|---|---|---|---|---|
| EWC-D | 10 | 10 | 14 | 4,353,184 | 4,353,184 |
| vanilla-D | 10 | 10 | 14 | **1,089,830** | **1,089,830** |
| full-D | 10 | 10 | 14 | **1,089,830** | **1,089,830** |
| full-D | 10 | 10 | 14 | 2,179,661 | 2,179,661 |
| lifelong-D | 1 | 10 | 14 | 2,165,311 | 2,179,661 |
| EWC-C | 10 | 10 | 14 | 30,767,428 | 30,767,428 |
| vanilla-C | 10 | 10 | 14 | **7,691,280** | **7,691,280** |
| full-C | 10 | 10 | 14 | **7,691,280** | **7,691,280** |
| full-C | 10 | 10 | 14 | 15,382,560 | 15,382,560 |
| lifelong-C | 1 | 10 | 14 | 15,235,072 | 15,382,560 |

The table above lists the number of parameters for each model and architecture for Experiment 5.1. The *lifelong* models initially start with a $\boldsymbol{z}_d$ of dimension 1 and at each step we grow the representation by one dimension to accommodate the new distribution (more info in Section 9.8). In contrast, the baselines are provided with the full representation throughout the learning process. EWC has double the number of parameters because the computed diagonal fisher information matrix is the same dimensionality as the number of parameters. EWC also neeeds the preservation of the teacher model $[\boldsymbol{\Phi}, \boldsymbol{\Theta}]$ to use in it's quadratic regularizer. Both the *vanilla* and *full* models have the fewest number of parameters as they do not use a student-teacher framework and only use one model, however the vanilla model has no protection against catastrophic interference and the *full* model is just used as an upper bound for performance.

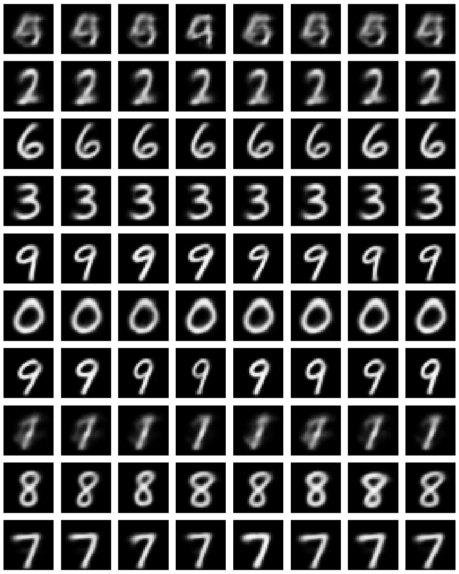
We utilized Adam [61] to optimize all of our problems with a learning rate of 1e-4 or 1e-3. When we utilized weight transfer we re-initialized the accumulated momentum vector of Adam as well as the aggregated mean and covariance of the Batch Norm layers. The full architecture can be examined in our github repository [15] and is provided under an MIT license.

[15] `https://github.com/jramapuram/LifelongVAE_pytorch`



We compared a whole range of EWC baselines and utilized the best performing models few in our experiments. Listed in Figure 9.4 are the full range of EWC baselines run on the Permuted 5.2 and Fashion 5.1 experiments. Recall that C / D describes whether a model is convolutional or dense and the the number following is the hyperparameter for the EWC or Lifelong VAE.

### *9.5. MNIST: Sequential Generation*



We evaluated Experiment 5.1 on MNIST in addition to FashionMNIST and achieved similar results. The best -test ELBO achieved by the *ewc* model was **149 nats** while the *lifelong* model reached 165 nats. However, the *lifelong* model achieved a Frechet distance of **128.42**, while the *ewc* model achieved 193.09. We visualize some of the sequential generations of our *lifelong* model in the figure on the right. Our decision to use FashionMNIST was to promote some diversity within our experiment framework.

### *9.6. ELBO Derivation*

Variational inference [62] side-steps the intractability of the posterior distribution by approximating it with a tractable distribution $Q_{\boldsymbol{\Phi}}(\boldsymbol{z}|\mathbf{x})$; we then optimize the parameters $\boldsymbol{\Phi}$ in order to bring this distribution close to $P_{\boldsymbol{\Phi}}(\boldsymbol{z}|\mathbf{x})$. The form of this approximate distribution is fixed and is generally conjugate to the prior $P(\boldsymbol{z})$. Variational inference converts the problem of posterior inference into an optimization problem over $\boldsymbol{\Phi}$. This allows us to utilize stochastic gradient descent to solve our problem. To be more concrete, variational inference tries to minimize the reverse Kullback-Leibler (KL) divergence between the variational posterior distribution $Q_{\boldsymbol{\Phi}}(\boldsymbol{z}|\mathbf{x})$ and the true posterior $P_{\boldsymbol{\theta}}(\boldsymbol{z}|\mathbf{x})$:

$$KL[Q_{\boldsymbol{\Phi}}(\boldsymbol{z}|\mathbf{x})||P_{\boldsymbol{\theta}}(\boldsymbol{z}|\mathbf{x})] = \log\, P_{\boldsymbol{\theta}}(\mathbf{x}) - \underbrace{\mathbb{E}_{Q_{\boldsymbol{\Phi}}(\boldsymbol{z}|\mathbf{x})}\left[\log \frac{P_{\boldsymbol{\theta}}(x,z)}{Q_{\boldsymbol{\Phi}}(\boldsymbol{z}|\mathbf{x})}\right]}_{\mathcal{L}_{\boldsymbol{\theta}}} \tag{15}$$

Rearranging the terms in equation 15 and utilizing the fact that the KL divergence is a measure, we can derive the evidence lower bound $\mathcal{L}_{\boldsymbol{\theta}}$ (ELBO) which is the objective function we directly optimize:

$$log\ P_{\boldsymbol{\theta}}(\mathbf{x}) \geq \mathbb{E}_{Q_{\boldsymbol{\Phi}}(\boldsymbol{z}|\mathbf{x})}[log\ P_{\boldsymbol{\theta}}(\mathbf{x}|\boldsymbol{z})] - KL(Q_{\boldsymbol{\Phi}}(\boldsymbol{z}|\mathbf{x})\ ||\ P(\boldsymbol{z})) = \mathcal{L}_{\boldsymbol{\theta}} \tag{16}$$

In order to backpropagate it is necessary to remove the dependence on the stochastic variable $\boldsymbol{z}$. To achieve this, we push the sampling operation outside of the computational graph for the normal distribution via the reparameterization trick [3] and the gumbel-softmax reparameterization [63, 64] for the discrete distribution. In essence the reparameterization trick allows us to introduce a distribution $P(\boldsymbol{\epsilon})$ that is not a function of the data or computational graph in order to move the gradient operator into the expectation:

$$\nabla\ \mathbb{E}_{Q_{\boldsymbol{\Phi}}(\boldsymbol{z}|\mathbf{x})}\left[\log\ \frac{P_{\boldsymbol{\theta}}(x,z)}{Q_{\boldsymbol{\Phi}}(\boldsymbol{z}|\mathbf{x})}\right] \mapsto \mathbb{E}_{P(\boldsymbol{\epsilon})}\left[\nabla\ \log\ \frac{P_{\boldsymbol{\theta}}(x,z)}{Q_{\boldsymbol{\Phi}}(\boldsymbol{z}|\mathbf{x})}\right] \tag{17}$$

### *9.7. Gumbel Reparameterization*

Since we model our latent variable as a combination of a discrete and a continuous distribution we also use the Gumbel-Softmax reparameterization [63, 64]. The Gumbel-Softmax reparameterization over logits [linear output of the last layer in the encoder] $\boldsymbol{p} \in \mathcal{R}^M$ and an annealed temperature parameter $\tau \in \mathcal{R}$ is defined as:

$$\boldsymbol{z} = softmax(\frac{log(\boldsymbol{p}) + \boldsymbol{g}}{\tau}); \boldsymbol{g} = -log(-log(\boldsymbol{u} \sim Unif(0,1))) \tag{18}$$

$\boldsymbol{u} \in \mathcal{R}^M, \boldsymbol{g} \in \mathcal{R}^M$. As the temperature parameter $\tau \mapsto 0$, $\boldsymbol{z}$ converges to a categorical.

### *9.8. Expandable Model Capacity and Representations*

Multilayer neural networks with sigmoidal activations have a VC dimension bounded between $O(\rho^2)$[65] and $O(\rho^4)$[66] where $\rho$ are the number of parameters. A model that is able to consistently add new information should also be able to expand its VC dimension by adding new parameters over time. Our formulation imposes no restrictions on the model architecture: i.e. new layers can be added freely to the new student model.

In addition we also allow the dimensionality of $\boldsymbol{z}_d \in \mathcal{R}^J$, our discrete latent representation to grow in order to accommodate new distributions. This is possible because the KL divergence between two categorical distributions of different sizes can be evaluated by simply zero padding the teacher's smaller discrete distribution. Since we also transfer weights between the teacher and the student model, we need to handle the case of expanding latent representations appropriately. In the event that we add a new distribution we copy all the weights besides the ones immediately surrounding the projection into and out of the latent distribution. These surrounding weights are reinitialized to their standard Glorot initializations [67].

### *9.9. Forward vs. Reverse KL*

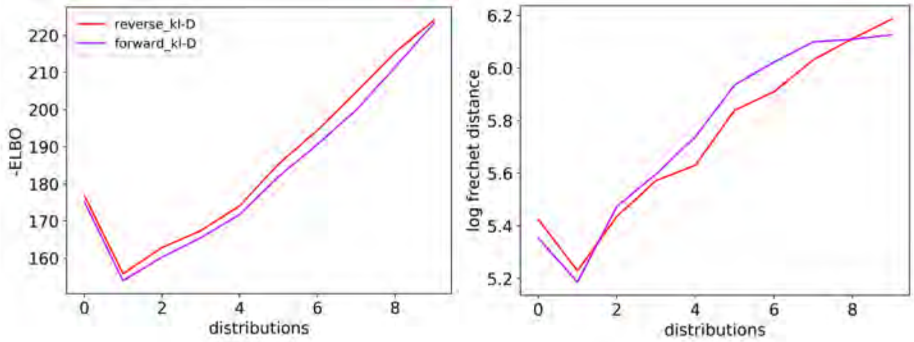


Figure 15: *left*: negative test elbo. *right*: *log-frechet* distance

in our setting we have the ability to utilize the zero forcing (reverse or mode-seeking) kl or the zero avoiding (forward) kl divergence. in general, if the true underlying posterior is multi-modal, it is preferable to operate with the reverse KL divergence ([68] 21.2.2). In addition, utilizing the mode-seeking KL divergence generates more realistic results when operating over image data.

In order to validate this, we repeat the experiment in 5.1. We train two models: one with the forward KL posterior regularizer and one with the reverse. We evaluate the -ELBO mean and variance over ten trials. Empirically, we observed no difference between the different measures. This is demonstrated in figure 15.

### *9.10. Frechet Performance Metric*

The idea proposed in [47] is to utilize a trained classifier model to compare the feature statistics (generally under a Gaussianity assumption) between synthetic samples of the generative model and samples drawn from the test set. If the Frechet distance between these two distributions is small, then the generative model is said to be generating realistic and diverse images. The Frechet distance between two gaussians with means $\boldsymbol{m}_{test}, \boldsymbol{m}_{gen}$ with corresponding covariances $\boldsymbol{C}_{test}, \boldsymbol{C}_{gen}$ is:

$$||\boldsymbol{m}_{test} - \boldsymbol{m}_{gen}||_2^2 + Tr(\boldsymbol{C}_{test} + \boldsymbol{C}_{gen} - 2[\boldsymbol{C}_{test}\boldsymbol{C}_{gen}]^{0.5}) \tag{19}$$

Photo of the author(s)

Click here to download high resolution image

***Conflict of Interest Form**

Conflict of Interest and Authorship Conformation Form

Please check the following as appropriate:

- ○ All authors have participated in (a) conception and design, or analysis and interpretation of the data; (b) drafting the article or revising it critically for important intellectual content; and (c) approval of the final version.
- ○ This manuscript has not been submitted to, nor is under review at, another journal or other publishing venue.
- ○ The authors have no affiliation with any organization with a direct or indirect financial interest in the subject matter discussed in the manuscript
- ○ The following authors have affiliations with organizations with direct or indirect financial interest in the subject matter discussed in the manuscript:

| Author's name | Affiliation |
|---|---|
| Jason Ramapuram | University of Geneva, HES-SO |
| Magda Gregorova | HES-SO |
| Alexandros Kalousis | University of Geneva, HES-SO |