Simulating Task-Free Continual Learning Streams From Existing Datasets Appendix

A. More Simulated Task-Free Streams

In Figure 1, Figure 2, Figure 3, we present a comparison of a conventional distinct-task stream and a simulated task-free (STF) for EMNIST, CIFAR-100, and tinyImageNet respectively.

B. Architectures

Please refer to Table 1.

C. Setting the Average Standard Deviation

We set the values of μ_σ by associating them to a number of tasks. Previous work typically uses a range of 5–20 number of tasks. In that case, each class's distribution is essentially a uniform with support the inverse of the number of tasks. If the number of tasks is set to T, the standard deviation of a uniform with support $1/T$ is $\sqrt{1/12} \times 1/T$. Here, we have used average standard deviations μ_σ which correspond to $T = 12, 5, 10, 20$ tasks for EMNIST, CIFAR-10, CIFAR-100, and tinyImageNet, respectively.

D. Beta and the Truncated Normal

Since the truncated normal is the maximum-entropy distribution of a given mean and standard deviation defined on a closed interval, we want to examine how much of its entropy the Beta can capture. We start by sampling a mean $\mu \in [0, 1]$ and a standard deviation $\sigma \in [0, 0.5]$, and we take a normal distribution with these moments and truncate it in $[0, 1]$. We evaluate the mean $\hat{\mu}$ and standard deviation $\hat{\sigma}$ of the truncated normal, and define a Beta with the same moments. Afterwards, we can evaluate the ratio of the entropy of the truncated normal that the Beta can capture. Performing this simulation 10^5 times, we find that the Beta captures 99.84% of the entropy of the truncated normal. Hence, we argue that the Beta can serve as a substitute for the truncated normal, since, as we mentioned earlier, it is more convenient mathematically.

E. On the Truncated Exponential

The *truncated exponential* distribution, with a *rate* parameter $\lambda \neq 0$ and the truncation parameter $\gamma > 0$ for its support, is defined as

$$f_{\mathcal{E}}(x \mid \lambda, \gamma) \triangleq ce^{\lambda x}, \quad x \in [0, \gamma], \tag{1}$$

where $c = \lambda/(e^{\lambda\gamma} - 1)$ is the *normalizing constant*. The mean of a $\mathcal{E}(\lambda, \gamma)$ over its support is computed as

$$\mu = c \int_0^\gamma x e^{\lambda x} dx = \frac{(\lambda\gamma - 1)e^{\lambda\gamma} + 1}{\lambda(e^{\lambda\gamma} - 1)} = \frac{\gamma}{1 - e^{-\lambda\gamma}} - \frac{1}{\lambda}. \tag{2}$$

An appropriate rate can be found by numerically solving the previous equation for specific values of μ and γ.

The *cumulative distribution function* (CDF) is

$$F_{\mathcal{E}}(x \mid \lambda, \gamma) = \frac{e^{\lambda x} - 1}{e^{\lambda\gamma} - 1}, \quad x \in [0, \gamma], \tag{3}$$

hence, we can sample from a TE distribution using *inverse-transform* sampling (ITS) as follows:

$$x = \frac{1}{\lambda} \ln\big[(e^{\lambda\gamma} - 1)u + 1\big], \quad \text{where } u \sim \mathcal{U}[0, 1]. \tag{4}$$

F. On the Beta Distribution

Here, we discuss our motivation for using instances of the Beta distribution to model the individual class distributions. Let us assume that we have sampled a mean μ_j and a standard deviation σ_j for each class j in our dataset, and now we want to assign to that class a distribution defined on the interval $[0, 1]$, with the same mean and standard deviation. The maximum-entropy distribution of a specified mean and standard deviation, and also defined on a bounded interval is called the truncated Gaussian [**?**]. However, we would also like to be able to easily derive the parameters of each distribution given its mean and standard deviation. In the case of the truncated normal, deriving its parameterization involves solving a non-linear system of equations, which does not have an analytical solution and is not guaranteed to be solvable in a numerically stable way. Instead, we argue that the Beta distribution is a more appropriate choice for a number reasons. First, deriving the parameters of a Beta given a desired mean and standard deviation is trivial (see **??** and **??**). Second, as we discussed earlier in the appendix, we empirically found that the Beta captures

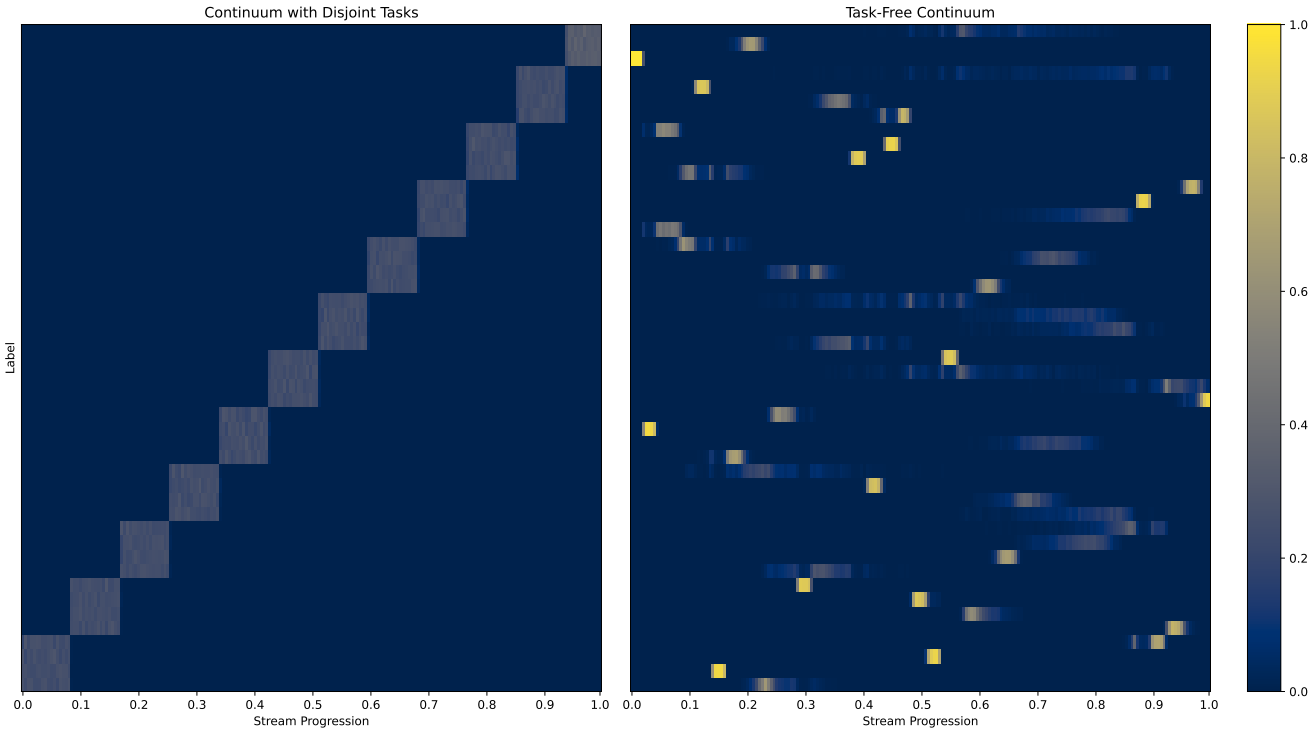

Figure 1. A conventional EMNIST stream with disjoint tasks (left) and a simulated task-free (STF) stream of the same dataset (right). Best viewed zoomed-in and in color.

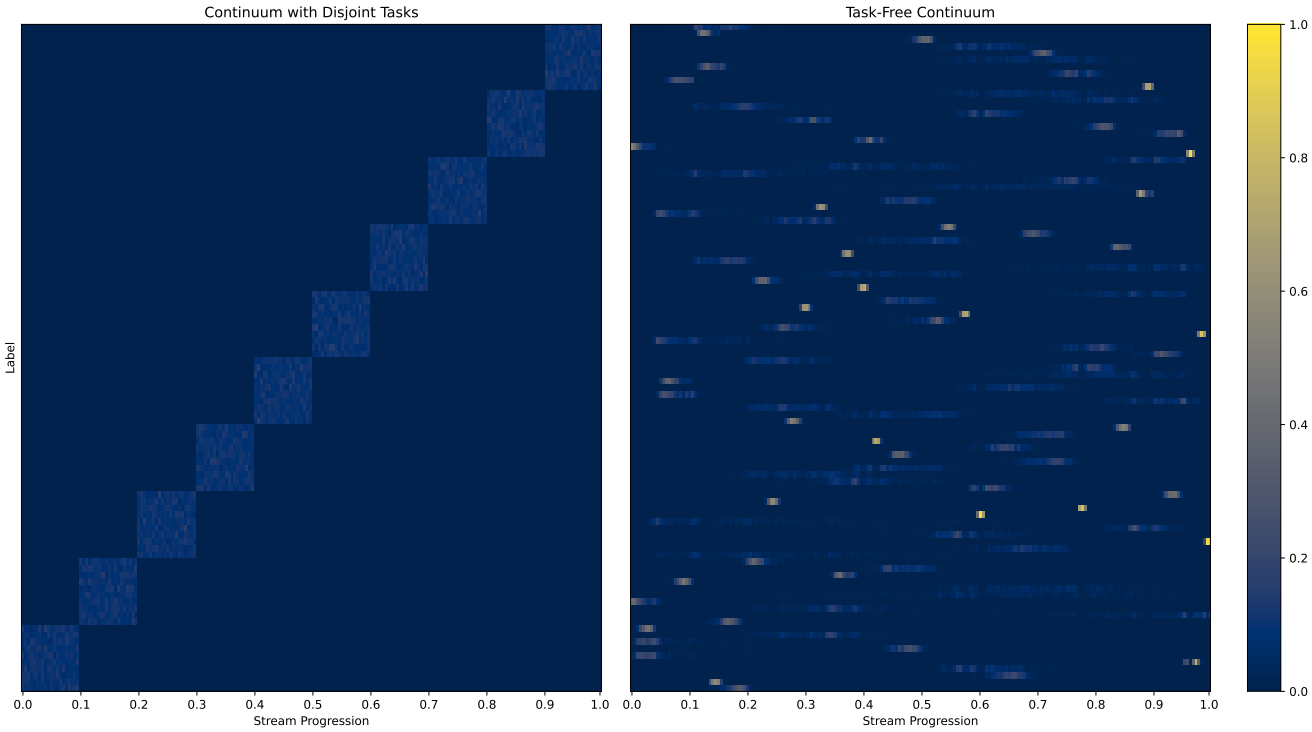

Figure 2. A conventional CIFAR-100 stream with disjoint tasks (left) and a simulated task-free (STF) stream of the same dataset (right). Best viewed zoomed-in and in color.

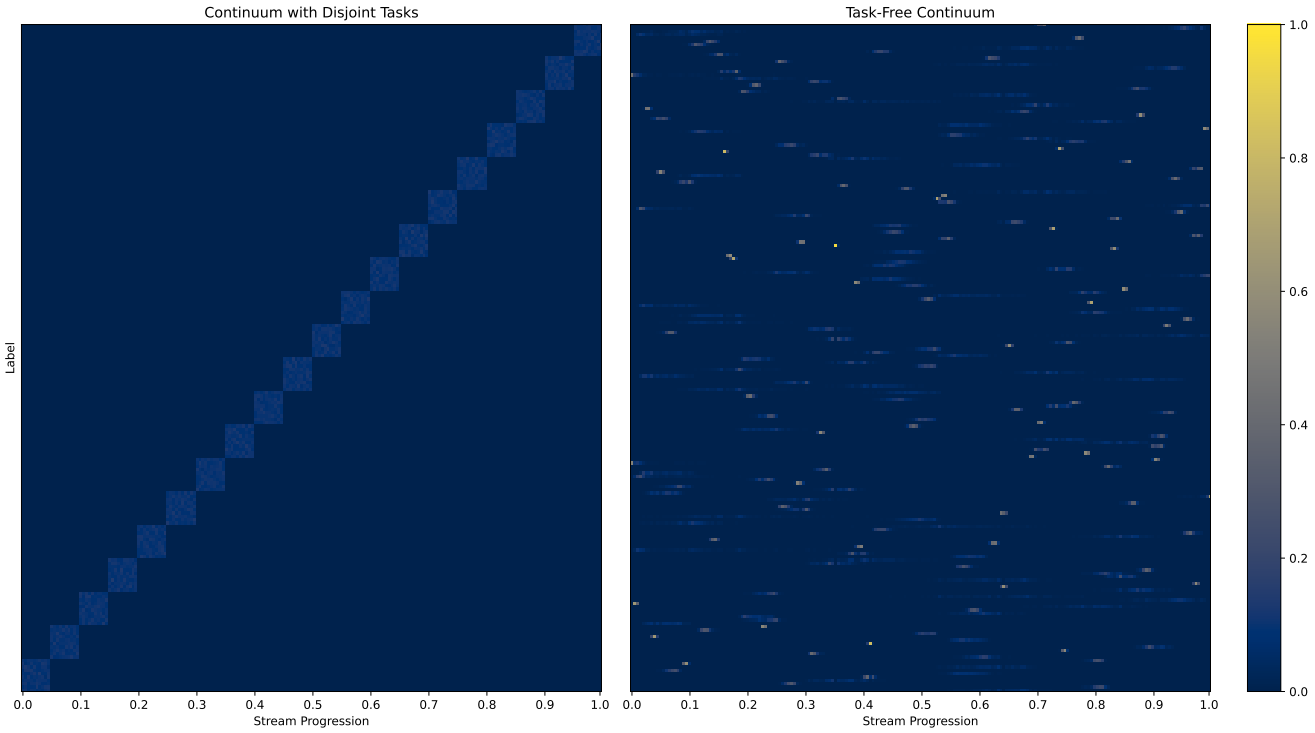

Figure 3. A conventional tinyImageNet stream with disjoint tasks (left) and a simulated task-free (STF) stream of the same dataset (right). Best viewed zoomed-in and in color.

Table 1. (left) A simple convolutional block; (middle) The Convolutional Neural Network (CNN) architecture used in the EMNIST experiments. (right) The reduced ResNet-18 architecture used for CIFAR-10, CIFAR-100, and tinyImageNet, is built using the BasicBlock(n_f, n_b, n_s) from [**?**], where n_f is the number of convolutional filters, n_b is the number of sub-blocks per block, and n_s is the stride of the layer.

ConvBlock	CNN	Reduced ResNet-18
Conv2D($n_{\text{in}}, n_{\text{out}}$)	ConvBlock($1, 32$)	BasicBlock(20, 2, 1)
ReLU	ConvBlock($32, 64$)	BasicBlock(40, 2, 2)
BatchNorm2D(n_{out})	Linear($64, c$)	BasicBlock(80, 2, 2)
Conv2D($n_{\text{out}}, n_{\text{out}}$)		BasicBlock(160, 2, 2)
ReLU		AveragePooling
BatchNorm2D(n_{out})		Linear($160, c$)
MaxPooling2D($2, 2$)		

99.84% of the entropy of the truncated normal, on average. Third, the Beta is mathematically convenient for our use-case since its support is the interval $[0, 1]$. Fourth, the Beta distribution takes various shapes (e.g., constant, bell-like, skewed to the left or to the right, etc.) for different parameterizations, a feature which allows for the creation of more diverse streams (please refer to Figure 4 for an illustration).

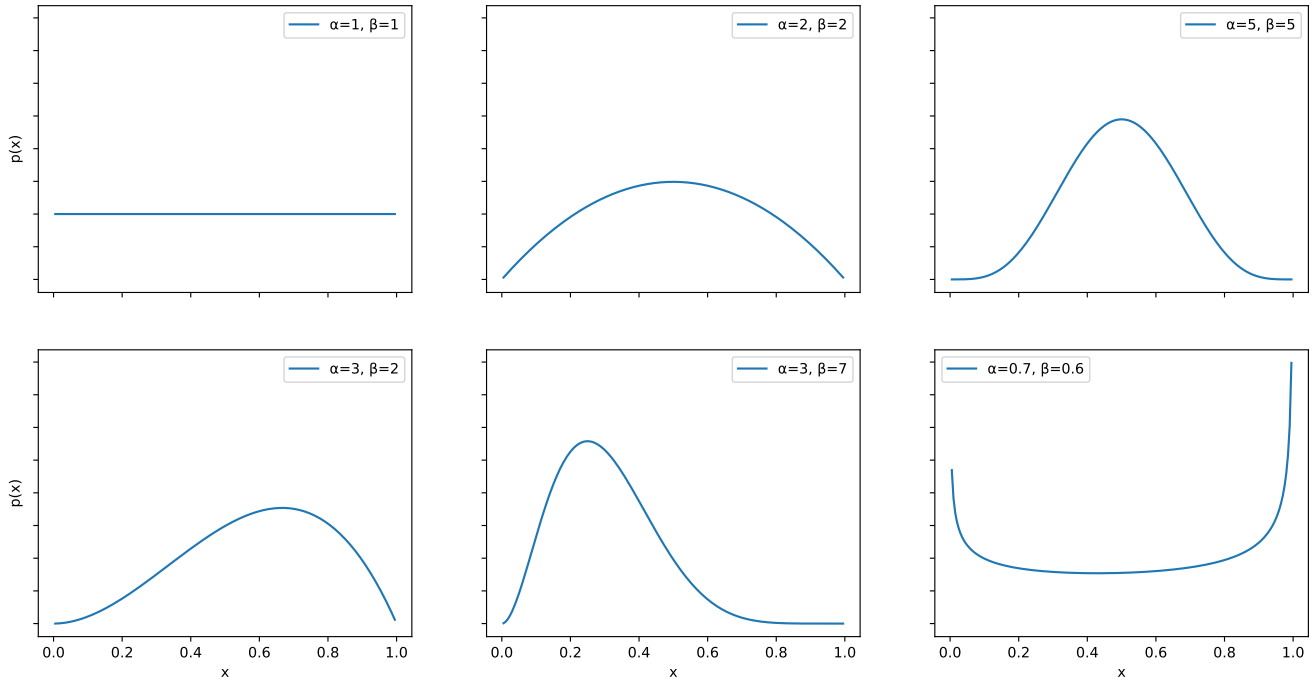

Figure 4. The Beta distribution can take a number of different shapes depending on its parameterization.