The Conditional Bernoulli and its Application to Speech Recognition

Sean Robertson

October 15, 2020

1 Motivations

A major challenge in speech recognition involves converting a variable number of speech frames $\{x_t\}_{t\in[1,T]}$ into a variable number of transcription tokens $\{y_\ell\}_{\ell\in[1,L]}$, where $L \ll T$. In hybrid architectures, y_ℓ are generated as a by-product of transitioning between states s_t in a weighted finite-state transducer. In end-to-end neural ASR, this process is commonly achieved either with Connectionist Temporal Classification (CTC) [14] or sequence-to-sequence (seq2seq) architectures [3]. The former introduces a special blank label; performs a one-to-many mapping $y_\ell \mapsto \tilde{y}_t^{(i)}$ by injecting blank tokens until the transcription matches length T in all possible configurations (i) during training; and removes all blank labels during testing. Seq2seq architectures first encode the speech frames x_t into some encoding h, then some separate recurrent neural network conditions on h to generate the token sequence y_ℓ.

In 2017, Luo et al. developed a novel end-to-end speech recognizer. Given a prefix of acoustic feature frames including the current frame $\{x_{t'}\}_{t'\in[t,T]}$ and a prefix of Bernoulli samples excluding the current frame $\{b_{t'}\}_{t'\in[t+1,T]}$, the recognizer produces a Bernoulli sample for the current frame $B_t \sim P(b_t|x_{\leq t}, b_{<t})$, plus or minus some additional conditioned terms. Whenever $B_t = 1$, the model "emits" a token drawn from a class distribution conditioned on the same information $Y_t \sim P(y_t|x_{\leq t}, b_{<t})$. The paper had two primary motivations. First, though it resembles a decoder in a *seq2seq* architecture [3], it does not need to encode the entire input sequence x_t before it can start making decisions about what was said, making it suitable to online recognition. Second, we can interpret the emission points, or "highs," of the Bernoulli sequence $B_t = 1$ as a form of hard alignment: the token output according to Y_t is unaffected by speech $x_{>t}$[1].

Because of the stochasticity introduced by sampling B_t discretely, the network cannot determine the exact gradient for parameterizations of B_t. Thus,

[1] This is not necessarily a synchronous alignment. $B_t = 1$ may occur well after whatever caused the emission. The last high $\arg\max_{t'<t} B_{t'} = 1$ cannot be assumed to bound the event to times after t' for the same reason. Finite t and vanishing gradients will force some synchronicity, however.

the authors rely on an estimate of the REINFORCE gradient [24]:

$$\frac{\partial R}{\partial \theta} = \mathbb{E}_b \left[\sum_{t=1}^{T} \left(\frac{\partial R_t}{\partial \theta} + \left(\sum_{t' \geq t} R_{t'} \right) \frac{\partial}{\partial \theta} \log P(b_t | b_{<t}, y_{<\ell_t}) \right) \right] \tag{1}$$

where

$$R_t = \begin{cases} \log P(Y_t = y_{\sum_{t'<t} b_{t'}} | x_{\leq t}, b_{<t}, y_{\sum_{t'<t-1} b_{t'}}) & \text{if } B_t = 1 \\ 0 & \text{if } B_t = 0 \end{cases} \tag{2}$$

The reward (eq. (2)) is the log probability of the k-th class label, where k counts the number of high Bernoulli values up to and including time step t. The return for time step t accumulates the instantaneous rewards for all non-past time steps $t' \geq t$.

In practice, using eq. (1) is very slow to train and yields mixed results. The authors found it was necessary to add a baseline function and an entropy function in order to converge. In a later publication [16], a bidirectional model[2] used Variational Inference to speed up convergence, though this failed to improve the overall performance of the model on the TIMIT corpus. The mixed performance and convergence of these models was blamed on the high-variance gradient estimate of eq. (1) [16].

We believe that the performance and convergence issues of these models are not due, at least in whole, to a high-variance estimate. Instead, we propose that a a bias was introduced into eq. (1) via the mechanism which guaranteed a fixed number of Bernoulli highs during training.

In order to ensure the total number of high Bernoulli values matched the total number of labels L during training, i.e. $\sum_t b_t = L$, the authors would force later samples to some specific value. For example, if at point $t = T - L + \ell$ only ℓ samples emitted, $B_t = 1$ regardless of $P(b_t)$. Likewise, if L samples emitted before t, $B_t = 0$.

Though this bias appears harmless at first, it has great ramifications for the estimator early on during training. We will provide an illustrative example.

Upon initialization of our network, we expect any dependencies between Bernoulli variables to be negligible. Thus, for the purposes of this example, we make the strong assumption that they are distributed i.i.d. with probability p. Since the variables are identically distributed, each has an equal probability of being chosen as one of the L highs, making $P(B_t = 1|L) = L/T$. However, under the reasonable assumptions $0 < L < T - L < T$, the emission probability for some given t induced by the forced-suffix mechanism evaluates to

$$\begin{aligned} P^*(B_t = 1 | L < T - L < T) = & p - I[t > L] p \bar{B}(p; L, t - L) + \\ & I[t > T - L](1 - p) \bar{B}(1 - p; T - L; t - T + L) \end{aligned} \tag{3}$$

[2] Forgoing the motivation for online speech recognition.

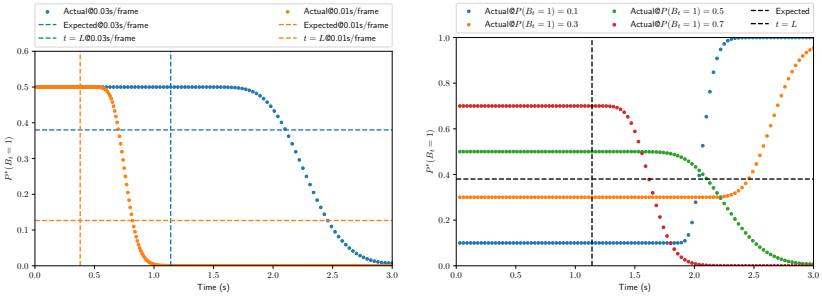

Figure 1: The disparity between the expected and actual probabilities of each Bernoulli random variable being high over time due to the forced suffix hack. Horizontal lines indicate the expected probability for the given number of frames and labels. Vertical lines indicate when the frame number matches the label number. *Left*: varying frame rate at $p = 0.5$. *Right*: varying p at frame rate of $0.03s$.

Where $I[\ldots]$ is the indicator function and $\bar{B}(p; a, b)$ is the Regularized Incomplete Beta function. The derivation of this distribution can be found in appendix A. The first $t \leq L$ Bernoulli variables get to ignore the restriction entirely, giving them a probability p; the samples $t > L$ lose out on some of that probability mass because some paths that would have contributed would have more than L samples; and, samples $t > L - T$ gain some additional probability mass when they are forced to 1 in order to emit L highs by the end of the utterance.

Figure 1 illustrates the distribution of $P^*(B_t = 1|\ldots)$ on an average-length utterance of TIMIT (3 seconds) and an average number of labels per utterance (38). The blue scatter plot on the left-hand side produces a Bernoulli random variable every 30 milliseconds, the same as Luo et al. [17], with even odds of each value. The forced suffix distribution overestimates the probability of the first two thirds of samples being and underestimates the last third, with the last sixth rarely if ever being sampled. The disparity becomes more exaggerated when we produce a random variable every 10 milliseconds (the orange scatter plot) - the usual rate at which speech features are produced - with over half of the Bernoulli variables having virtually zero probability of drawing value 1. The right-hand plot illustrates that the bias becomes more or less severe depending on how close $P(B_t = 1)$ is to the expected probability L/T.

While eq. (3) makes strong independence assumptions about the Bernoulli random variables, the fundamental flaw with the distribution will hold even if we relax these assumptions. Even under the distribution with dependence, a prefix of Bernoulli random variables will ignore the conditioning on the number of labels, $t \leq L \implies P^*(b_t|b_{<t}, L) = P(b_t|b_{<t})$, whose over- or underestimation of $P(b_t|b_{<t}, L)$ will be paid for in the suffix through either under- or overestimation itself. Indeed, Luo et al. [17] observed this predicted behaviour themselves: without an additional entropy penalty, the model would learn to emit entirely

at either the beginning or end of the sequence.

To solve the problem of bias, we propose replacing the T independent Bernoulli random variables B_t sampled during training with a single sample B from the Conditional Bernoulli (CB) distribution during training. The CB conditions on the required number of high trials, which will make the objective well-defined during training. It avoids placing undue emphasis on earlier trials, which should curtail the convergence problems faced by Luo et al. [17]. In addition, the CB can be decomposed into Bernoulli trials that condition on past trial results, similar to eq. (2). We also show how the CB can be relaxed to a continuous variable for use in Straight-Through estimators [4, 15] or RELAX-like estimators [18, 11]. Finally, we outline under which conditions the likelihood of y can be exactly and efficiently calculated under the assumptions of the CB, and how it relates to CTC and RNN Transducers [13].

2 The Conditional Bernoulli

2.1 Definitions

The Conditional Bernoulli distribution [8, 7], sometimes called the Conditional Poisson distribution [1, 5], is defined as

$$P\left(b \middle| \sum_t b_t = k; w\right) = \frac{\prod_t w_t^{b_t}}{\sum_{\left\{b' : \sum_t b'_t = k\right\}} \prod_t w_t^{b'_t}} \tag{4}$$

Where $w_t = p_t/(1-p_t)$ are the odds/weights of a Bernoulli random variable $B_t \sim P(b_t; w_t) = p_t^{b_t}(1-p_t)^{(1-b_t)} = w_t^{b_t}/(1+w_t)$. Equation (4) reads as "what is the probability that Bernoulli random variables $B = \{B_t\}_{t\in[1,T]}$ have values $\{b_t\}_t$, given that exactly k of them are high ($\sum_t b_t = k$)?" Letting $K = \sum_t B_t$, K is a random variable that counts the total number of "highs" in a series of Bernoulli trials. K is distributed according to the Poisson-Binomial (PB) distribution, a generalization of the Binomial distribution for when $p_i \neq p_j$. It is defined as

$$\begin{aligned} P(K = k; w) &= \sum_{\left\{b : \sum_t b_t = k\right\}} P(b; w) \\ &= \left(\prod_{t=1}^{T} 1 + w_t\right)^{-1} \sum_{\left\{b : \sum_t b_t = k\right\}} \prod_{t=1}^{T} w_t^{b_t} \end{aligned} \tag{5}$$

If we use eq. (5) to marginalize out K from eq. (4), we recover the independent Bernoulli probabilities:

$$\begin{aligned}
P(b;w) &= \sum_{k=0}^{T} P(b,k;w) = \sum_{k=0}^{T} P(b|k;w)P(k;w) \\
&= P(b|k';w)P(k';w) \text{ for exactly one } k' = \sum_t b_t \\
&= \left(\prod_t 1 + w_t\right)^{-1} \frac{\prod_{t=1}^{T} w_t^{b_t}}{\sum_{\{b':\sum_t b'_t = k'\}} \prod_{t=1}^{T} w_t^{b'_t}} \left(\sum_{\{b':\sum_t b'_t = k'\}} \prod_{t=1}^{T} w_t^{b'_t}\right) \\
&= \prod_{t=1}^{T} (1+w_t)^{-1} w_t^{b_t}
\end{aligned} \tag{6}$$

Which is to say that, if we do not have knowledge of the number of highs *a priori*, assuming a Poisson-Binomial prior, the probability of sample B is the product of the probabilities of the outcomes of T independent Bernoulli trials.

Direct calculation of equation eq. (4) involves summing over T-choose-k products of k odds, making it infeasible for large T and k. To combat this, Chen and Liu [7] propose a number of alternative algorithms where the sample B is constructed by iteratively deciding on the individual values of B_i. We will not only use these algorithms for efficiency: we will also use them to factor the CB distribution into useful forms for different objectives.

To better describe these algorithms, we define the set of indices $t \in [1,T] = I$ s.t. $B = \{B_t\}_{t\in I}$. The set $A \subseteq I$ maps to some sample B such that all the high Bernoulli variables' indices can be found in A, i.e. $B_t = 1 \iff t \in A$. Then the CB can be restated as

$$P(A|k;w) = \frac{\prod_{a\in A} w_a}{C(k,I;w)} \tag{7}$$

where

$$C(v,S;w) = \sum_{\{A'\subseteq S:|A'|=v\}} \prod_{a\in A'} w_a \tag{8}$$

normalizes over all possible k-tuples of w_i in some set S. Equation (8) can be considered a generalization of the binomial coefficient, which can be recovered by setting all $w_t = 1$. If we identify the product of weights from a set A as a weight indexed by A (i.e. $\prod_{a\in A} w_a \mapsto w'_A$), we can interpret eq. (7) as a categorical distribution.

The Draft Sampling procedure [7] recursively builds A by choosing a new weight to add to an ordered set. We use $j \in [1,T]$ to index elements of I in the order in which they are drafted into A: $I = \{t_j\}_j$, $A_j = (t_1, t_2, \dots t_j)$, and $A_j^c = I \setminus A_j = \{t_{j+1}, t_{j+2}, \dots, t_T\}$. Then the probability that some $t \in A_{j-1}^c$ is

the j-th sample to be drafted into A is defined as

$$P(t \in A_j | A_{j-1}, k; w) = \frac{w_t C(k-j, A_{j-1}^c \setminus \{t\}; w)}{(k-j+1) C(k-j+1, A_{j-1}^c; w)} \tag{9}$$

Terms in both the numerator and denominator of eq. (9) sum over suffix sets of length $k-j+1$ that could be appended to A_{j-1} to get a k-tuple A. The numerator is the sum of products of odds including w_t. The conditional probability is conditioned on the remaining ("future") odds with respect to j, as well as whatever samples t_j were chosen in the past. The total probability of a drafted sample is

$$\begin{aligned} P(A_k | k; w) &= \prod_{j=1}^{k} P(t_j \in A_j | A_{j-1}; w) \\ &= \prod_{j=1}^{k} \frac{w_{t_j} C(k-j, A_j^c; w)}{(k-j+1) C(k-j+1, A_{j-1}^c; w)} \\ &= \left(\prod_{j=1}^{k} w_{t_j} \right) \frac{C(0, A_k^c; w)}{k! C(k, I; w)} \\ &= \frac{1}{k!} P(A | k, w) \end{aligned} \tag{10}$$

Equation (10) produces almost the same probability as the Conditional Bernoulli, except for the factorial term. The factorial term accounts for the fact that samples are drafted into A_k in some fixed order. Summing over the probabilities of the $k!$ possible permutations of A_k yields the Conditional Bernoulli. We will call the distribution defined in eq. (10) the Draft Bernoulli (DB). Though the DB is not the same distribution as the CB, an expected value over the DB will be the same as that over the CB as long as the order of samples in A_k is ignored by the value function.

The ID-Checking Sampling procedure [7] is another useful treatment of the CB. This procedure builds A by iterating over Bernoulli trials and making binary decisions whether to include the trial in A. First, choose and fix an order j in which samples I will potentially be added to A. Let $A_{r_j,j} \subseteq A_j = (t_1, t_2, \ldots, t_j)$ be the subset of r_j samples ($|A_{r_j,j}| = r_j$) that have been added to A. At every step j, we choose to either add t_j to $A_{r_{j-1},j-1}$ and recurse on $A_{r_j,j} = A_{r_{j-1},j-1} \cup \{t_j\}$ or exclude t_j and recurse on $A_{r_j,j} = A_{r_{j-1},j-1}$. The probability of including t_j is

$$P(t_j \in A_{r_j,j} | A_{r_{j-1},j-1}, k; w) = \frac{w_{t_j} C(k - r_{j-1} - 1, A_j^c; w)}{C(k - r_{j-1}, A_{j-1} u^c; w)} \tag{11}$$

From the perspective of Bernoulli trials, $P(t_j \in A_{r_j,j} | \ldots) = P(B_{t_j} = 1 | k - r_j; w)$. Equation (11) can be interpreted as the probability that B_{t_j} is high, given that $k - r_j$ remaining trials must be high. Like in eq. (9), the numerator

and denominator of eq. (11) consist of products of weights of possible suffixes. The numerator only includes suffixes where w_{t_j} is a multiplicand.

The joint probability of a prefix of Bernoulli trials $b_{t_{\leq j}} = (b_{t_1}, b_{t_2}, \ldots, b_{t_j})$ using eq. (11) equals

$$\begin{aligned} P(b_{t_{\leq j}}|k - r_j; w) &= \prod_{j'=1}^{j} P(b_{t_{j'}}|k - r_{j'}; w) \\ &= \prod_{j'=1}^{j} \frac{w_{t_{j'}}^{b_{t_{j'}}} C(k - r_{j'}, A^c_{j'}; w)}{C(k - r_{j'-1}, A^c_{j'-1}; w)} \\ &= \left(\prod_{j'=1}^{j} w_{t_{j'}}^{b_{t_{j'}}} \right) \frac{C(k - r_j, A^c_j; w)}{C(k, I; w)} \end{aligned} \tag{12}$$

The dependence on prior trials is implicit in the $r_{j'}$ term. We will call the family of distributions over different prefixes the ID-checking Bernoulli (IDB). When the prefix is the length of the entire sequence $j = T$, $P(b_{t_{\leq T}}|k - r_T; w) = P(b|k; w)$ and the IDB distribution matches the CB distribution.

We will find a novel third decomposition useful. This method combines the ID-Checking and Drafting methods so that the draft at a given step must come from a bounded suffix of weights. Define $A_{r,j_r} \subseteq A_{j_r} = (t_1, t_2, \ldots t_{j_r})$ to be the C samples of A_{j_r} that have been added to A. Define the probability that the next sample $t_j \in A_{t_{j_{r-1}}}$ is the r-th drafted sample to be

$$P(j = j_r|k - r, j_{r-1}; w) = \frac{w_{t_j} C(k - r, A^c_{t_j}; w)}{C(k - r + 1, A^c_{t_{j_{r-1}}}; w)} \tag{13}$$

The draft is bound to the suffix $A^c_{t_{j_{r-1}}} = (t_{j_{r-1}+1}, t_{j_{r-1}+2}, \ldots, t_{j_T})$. Further, the draft requires that if t_j is the r-th draft, the remaining drafts must come from indexed values $t_{>j}$. To balance the restriction, earlier t_j will be more probable than later t_j to be drafted earlier. Using the fact that $C(k - r + 1, A^c_{t_j}; w) = w_{t_j+1} C(k - r, A^c_{t_j+1}; w) + C(k - r + 1, A^c_{t_j+1}; w)$, it is easily shown via induction that $C(k - r + 1, A^c_{t_{j_{r-1}}}; w) = \sum_{j=j_{r-1}+1}^{T} w_{t_j} C(k - r, A^c_{t_j}; w)$, proving that eq. (13) is a valid probability distribution. The probability of a draft prefix is calculated as

$$\begin{aligned} P(A_{r,j_r}|k - r; w) &= \prod_{r'=1}^{r} P(j_{r'}|k - r', j_{r'-1}; w) \\ &= \prod_{r'=1}^{r} \frac{w_{t_{j_{r'}}} C(k - r', A^c_{t_{j_{r'}}}; w)}{C(k - r' + 1, A^c_{t_{j_{r'-1}}}; w)} \\ &= \left(\prod_{r'=1}^{r} w_{t_{j_{r'}}} \right) \frac{C(k - r, A^c_{t_{j_r}}; w)}{C(k, I; w)} \end{aligned} \tag{14}$$

We call this distribution the Bounded Bernoulli (BB). When $r = k$, the BB matches the CB. The BB fixes the multiple orderings problem of the DB. The conditional probabilities of eq. (13) can be efficiently calculated using intermediate values when calculating C using Method 2 from [7]. Observing eqs. (12) and (14), the probability of a prefix under the IDB matches a probability of some BB draft prefix whenever the last sampled Bernoulli from the IDB was high. Assuming $b_{t_j} = 1$ and $\sum_{j'=1}^{j} b_{t_{j'}} = r$, $b_{t_{\leq j}} \mapsto A_{r,j_r}$ by the relation $b_{t'} = 1 \Leftrightarrow t' \in A_{r,j_r}$. The BB allows us to marginalize out prior drafted samples and ask what the probability is that t_j is the r-th drafted sample:

$$\begin{aligned} P(j = j_r | k - r; w) &= \sum_{A_{r,j_{r-1}}} P(A_{r,j_r} | k - r; w) \\ &= \left(\sum_{\{j_{<r} : j_{r'} < j\}} \prod_{r'=1}^{r-1} w_{t_{j_{r'}}} \right) \frac{w_{t_j} C(k - r, A^c_{t_j}; w)}{C(k, I; w)} \\ &= \frac{C(r - 1, A_{t_{j-1}}; w) w_{t_j} C(k - r, A^c_{t_j}; w)}{C(k, I; w)} \end{aligned} \tag{15}$$

The second line features sums over the possible size-$(r-1)$ prefixes that could have been drafted prior to j, which means that each occurs within the subset $A_{t_{j-1}}$. Intuitively, the numerator enumerates all possible prefixes and all possible suffixes around w_{t_j}, subject to the constraint that $r - 1$ elements come before and $k - r$ come after.

t_j being the r-th drafted sample and t_j being the $(r+1)$-th drafted sample are clearly disjoint events. Summing over these disjoint probabilities recovers the probability that t_j belongs to A:

$$\begin{aligned} \sum_{r=1}^{k} P(j = j_r | k - r; w) &= \frac{w_{t_j}}{C(k, I; w)} \sum_{r=1}^{k} C(r - 1, A_{t_{j-1}}; w) C(k - r, A^c_{t_j}; w) \\ &= \frac{w_{t_j} C(k - 1, I \setminus \{t_j\})}{C(k, I; w)} \end{aligned} \tag{16}$$

where the second line follows from noting $A_{t_{j-1}} \cup A^c_{t_j} = I \setminus \{t_j\}$ and applying Proposition 1.c. from Chen et al. [8]:

$$\forall S \subseteq I \quad C(k, I; w) = \sum_{r=0}^{k} C(r, S; w) C(k - r, I \setminus S; w) \tag{17}$$

a generalization of Vandermonde's identity.

Outside of statistics, Swersky et al. [22] linked the CB distribution with the goal of choosing a subset of k items from a set of N alternatives. In this case, the N alternatives are class labels, where one or more class labels may be active at a time. Models could be trained in a Maximum-Likelihood setting using the CB distribution: $B_n = 1$ implies class n is present and the probability of the

data can be estimated via eq. (4). The authors note that it was insufficient to rely on the implicit prior induced by training via eq. (4) and had to explicitly learn and condition on it.

Xie and Ermon [27] approximates the T-choose-k sampling procedure by using a top-k procedure called Weighted Reservoir Sampling. This procedure produces samples in an identical fashion to the Plackett-Luce (PL) distribution [28], which has also been explored in the realm of gradient estimation [10]. While the PL distribution has a similar construction to the DB, its top-k rankings do not have a uniform distribution over permutations and, as such, the PL does not match the expectation of the CB. Nonetheless, estimators involving the DB can be trivially modified to sample from the PL.

2.2 REINFORCE Objective

From section 1, we are interested in sampling T Bernoulli random variables such that the total number of emissions/highs matches the number of tokens L during training. We will start by considering the probability of a token sequence $y = \{y_\ell\}_{\ell \in [1,L]}$ under a model and work our way to a REINFORCE objective. For brevity, we supress conditioning on the acoustic data $\{x_t\}_{t \in [1,T]}$ and model parameters.

$$\begin{aligned} P(y) &= P(y, L) \\ &= \sum_b P(y, b, L) \\ &= \sum_b P(b, L) P(y|b) \\ &= P(L) \sum_{b|L} P(b|L) P(y|b) \\ &= P(L) \mathbb{E}_{b|L} \left[P(y|b) \right] \end{aligned} \tag{18}$$

Where $P(y) = P(y, L)$ follows from the fact that L is a deterministic function of y.

Taking the log, we get

$$\begin{aligned} \log P(y) &= \log P(L) + \log \mathbb{E}_{b|L} \left[P(y|b) \right] \\ &\geq \log P(L) + \mathbb{E}_{b|L} \left[\log P(y|b) \right] \end{aligned}$$

Where we have used Jensen's Inequality to establish a lower bound. Calling the bound R and differentiating with respect to some parameter θ, we get

$$\frac{\partial R}{\partial \theta} = \frac{\partial \log P(L)}{\partial \theta} + \frac{\partial}{\partial \theta} \mathbb{E}_{b|L} \left[\log P(y|b) \right] \tag{19}$$

We have yet to make any assumptions about the distributions of any $P(\cdot)$, except to say that $|y| = L$. To recover the REINFORCE objective of eq. (1), we

assume B is a sequence of independent Bernoulli trials. Further, we approximate $P(b, L) \approx P(b)$. Then we factor $P(y, b)$ as [16]:

$$P(y, b) = \prod_{t=1}^{T} P(y_{\ell_t} | b_{\leq t}, y_{<\ell_t})^{b_t} P(b_t | b_{<t}, y_{<\ell_t}) \tag{20}$$

where $\ell_t = \sum_{t'=1}^{t} b'_t$.

Under these assumptions, the rightmost expectation in eq. (19) decomposes into[3]

$$\begin{aligned}
\frac{\partial}{\partial\theta}\mathbb{E}_b\left[\log P(y|b)\right] &= \frac{\partial}{\partial\theta}\mathbb{E}_b\left[\sum_{t=1}^{T} b_t \log P(y_{\ell_t}|b_{\leq t}, y_{<\ell_t})\right] \\
&= \sum_{t=1}^{T} \frac{\partial}{\partial\theta}\mathbb{E}_b\left[R_t\right] \text{ from eq. (2)} \\
&= \sum_{t=1}^{T} \frac{\partial}{\partial\theta}\mathbb{E}_{b_{\leq t}}\left[R_t\right] \text{ since } R_t \text{ not based on } b_{>t} \\
&= \sum_{t=1}^{T} \mathbb{E}_{b_{\leq t}}\left[\frac{\partial R_t}{\partial\theta} + R_t \frac{\partial}{\partial\theta}\log P(b_{\leq t}|y_{<\ell_t})\right] \\
&= \sum_{t=1}^{T} \mathbb{E}_{b_{\leq t}}\left[\frac{\partial R_t}{\partial\theta} + R_t \sum_{t'\leq t} \frac{\partial}{\partial\theta}\log P(b_{t'}|b_{t'-1}, y_{<\ell_{t'}})\right] \\
&= \mathbb{E}_b\left[\sum_{t=1}^{T}\left(\frac{\partial R_t}{\partial\theta} + \left(\sum_{t'\geq t} R_{t'}\right)\frac{\partial}{\partial\theta}\log P(b_t|b_{<t}, y_{<\ell_t})\right)\right]
\end{aligned}$$

The expectation of the sum of frame-wise objectives is the same as the expectation of the "global" objective, where no subset of B is attributed to a given class label y_ℓ:

$$\frac{\partial}{\partial\theta}\mathbb{E}_b\left[\log P(y|b)\right] = \mathbb{E}_b\left[\sum_{\ell=1}^{L}\left(\frac{\partial \log P(y_\ell|b)}{\partial\theta} + \log P(y_\ell|b)\frac{\partial}{\partial\theta}\log P(b)\right)\right]$$

However, the frame-wise - or "local" - signal is assumed to be less noisy [19].

The decomposition of $P(y, b)$ from eq. (20) is only well-defined when $|y| = \sum_{t=1}^{T} b_t$. This is not a problem during testing but it is during training when $|y|$ is fixed. For that reason, Luo et al. [17] hacks the Bernoulli sequence probabilities using the methods described in section 1. This produces a biased estimator with a variety of problems. If we can condition the joint on the number of highs in y, we can avoid the problem entirely.

The easiest fix to being ill-defined is to remove the auto-regressive property over Bernoulli trials and treat them as independent: $P(b) = \prod_{t=1}^{T} P(b_t)$. In this

[3]Thanks to Dieterich Lawson for this derivation.

case, $P(b|L)$ is the CB a global REINFORCE objective can be defined as

$$\frac{\partial R}{\partial \theta} = \frac{\partial \log P(L)}{\partial \theta} + \mathbb{E}_{b|L}\left[\frac{\partial \sum_{t=1}^{T} R_t}{\partial \theta} + \left(\sum_{t=1}^{T} R_t\right)\frac{\partial}{\partial \theta}\log P(b|L)\right] \tag{21}$$

Equation (21) is tractable and, unlike eq. (1), well-defined. Unfortunately, it is no longer autoregressive nor local.

The requirement that the model is not auto-regressive with respect to sequential B_t is a by-product of sampling from $P(b|L)$. If $P(b|L)$ is a Conditional Bernoulli, the odds of each Bernoulli trial w_t must be known before sampling a prefix. In an auto-regressive model, w_t depends on the prefix of samples. We know of no way to determine all w_t without iterating through all the sequences of Bernoulli trials with L highs, which would be intractable. Some distribution other than the CB could be chosen for $P(b|L)$, but this distribution would need to be able to sample both $P(b_t|b_{<t}, L)$ and $P(b_t|b_{<t})$ (i.e. with or without conditioning on the number of class labels) without conditioning on the odds of future events. To the best of our knowledge, existing research meets one, but not both, requirements.

That being said, even though w_t cannot condition on prior samples or class labels, $R_t = b_t \log P(y_{\ell_t}|\dots)$ can. The model can still be auto-regressive, as long as that auto-regression does not impact the odds of a given Bernoulli sample. We can re-inject the auto-regressive property into the model by treating $P(y_{\ell_t}|\cdots)$ as the output of an auto-regressive decoder whose decision to step forward depends on whether $B_t = 1$.

If we still assume no prior dependence between Bernoulli trials B_t, the expectation is over the CB distribution $P(b|L)$. We can use the various decompositions of the CB defined in section 2.1 to derive local gradient estimates.

Our first frame-wise objective is courtesy of the IDB decomposition of the CB from eq. (12). Though a given trial sample B_{t_j} is conditioned on non-past weights $w_{t_{\geq j}}$, it is only conditioned on samples from the past $b_{t_{<j}}$. Setting $t_j = j$, we decompose the joint probability of the class label sequence and the CB sample as

$$P(y, b|L) = P(y|b, L)P(B|L) = \prod_{t=1}^{T} P(y_{\ell_t}|b_{\leq t}, y_{<\ell_t})^{b_t} P(b_t|L - r_t) \tag{22}$$

Equation (22) is very similar to eq. (20), except the conditioning on the number of class labels L forces ℓ_t to be well-defined whenever $B_t = 1$. The derivation of the IDB REINFORCE gradient is almost identical to that for eq. (1), yielding

$$\frac{\partial R}{\partial \theta} = \frac{\partial \log P(L)}{\partial \theta} + \mathbb{E}_{b|L}\left[\sum_{t=1}^{T}\left(\frac{\partial R_t}{\partial \theta} + \left(\sum_{t' \geq t} R_{t'}\right)\frac{\partial}{\partial \theta}\log P(b_t|L - \ell_t)\right)\right] \tag{23}$$

where $R_t = b_t \log P(y_{\ell_t}|b_{\leq t}, y_{<\ell_t})$.

Equation (23) is very similar to eq. (1), but is unbiased. In the example given in fig. 1, the IDB estimate will treat each valid sequence of Bernoulli trials as equally likely. The sum of rewards over future trials is a function of the dependence of R_t on past trials $b_{<t}$.

We can prove that the variance of eq. (23) is no greater than that of eq. (21). Representing the expectation in eq. (23) as $\mathbb{E}_{b|L}[Y]$ and that in eq. (21) as $\mathbb{E}_{b|L}[Z]$,

$$\begin{aligned}
\mathbb{E}_{b|L}[Z] &= \mathbb{E}_{b|L}\left[\frac{\partial \sum_{t=1}^{T} R_t}{\partial\theta} + \left(\sum_{t'=1}^{T} R_{t'}\right)\left(\sum_{t=1}^{T}\frac{\partial}{\partial\theta}\log P(b_t|b_{<t},L)\right)\right] \\
&= \mathbb{E}_{b|L}\left[\sum_{t=1}^{T}\left(\frac{\partial R_t}{\partial\theta} + \left(\sum_{t'=1}^{T} R_{t'}\right)\frac{\partial}{\partial\theta}\log P(b_t|b_{<t},L)\right)\right] \\
&= \mathbb{E}_{b|L}\left[\sum_{t=1}^{T}\left(\frac{\partial R_t}{\partial\theta} + \left(\sum_{t'=t}^{T} R_{t'}\right)\frac{\partial}{\partial\theta}\log P(b_t|b_{<t},L)\right) + \right. \\
&\qquad \left. \sum_{t=1}^{T}\left(\left(\sum_{t'=1}^{t-1} R_{t'}\right)\frac{\partial}{\partial\theta}\log P(b_t|b_{<t},L)\right)\right] \\
&= \mathbb{E}_{b|L}[Y+X] \\
&= \mathbb{E}_{b|L}[Y] + \mathbb{E}_{b|L}[X]
\end{aligned} \tag{24}$$

We can see that $Z = X + Y$. We first prove that the expectation of the two estimators are equivalent by showing $\mathbb{E}_{b|L}[X] = 0$:

$$\begin{aligned}
\mathbb{E}_{b|L}[X] &= \mathbb{E}_{b|L}\left[\sum_{t=1}^{T} R_{<t}\frac{\partial}{\partial\theta}\log P(b_t|b_{<t},L-\ell_t)\right] \\
&= \sum_{t=1}^{T}\mathbb{E}_{b|L}\left[R_{<t}\frac{\partial}{\partial\theta}\log P(b_t|b_{<t},L-\ell_t)\right] \\
&= \sum_{t=1}^{T}\mathbb{E}_{b_{<t}|L}\left[\mathbb{E}_{b_t|b_{<t},L}\left[R_{<t}\frac{\partial}{\partial\theta}\log P(b_t|b_{<t},L-\ell_t)\right]\right] \\
&= \sum_{t=1}^{T}\mathbb{E}_{b_{<t}|L}\left[R_{<t}\mathbb{E}_{b_t|b_{<t},L}\left[\frac{\partial}{\partial\theta}\log P(b_t|b_{<t},L-\ell_t)\right]\right] \\
&= \sum_{t=1}^{T}\mathbb{E}_{b_{<t}|L}\left[R_{<t}\frac{\partial}{\partial\theta}\mathbb{E}_{b_t|b_{<t},L}[1]\right] \\
&= 0
\end{aligned} \tag{25}$$

Now we can treat the random variable Z as a function of X and Y, $Z = X+Y = J(X,Y)$ and calculate the marginal expectation of Y:

$$\widehat{J}(Y) = \mathbb{E}_x[J(X,Y)|Y] = Y + \mathbb{E}_x[X] = Y \tag{26}$$

which follows from eq. (25). By eq. (24), $\mathbb{E}_y[\widehat{J}(Y)] = \mathbb{E}_y[Y]$ is expectation in the IDB estimator of eq. (23). The remainder of the proof is merely an application of the Rao-Blackwell-Kolmogorov theorem:

$$\begin{aligned} Var(J(X,Y)) &= \mathbb{E}_{x,y}[J(X,Y)^2] - \mathbb{E}_{x,y}[J(X,Y)]^2 \\ &= \mathbb{E}_y[\mathbb{E}_x[J(X,Y)^2|Y]] - \mathbb{E}_{x,y}[J(X,Y)]^2 \\ &\geq \mathbb{E}_y[\mathbb{E}_x[J(X,Y)|Y]^2] - \mathbb{E}_{x,y}[J(X,Y)]^2 \\ &= \mathbb{E}_y[\widehat{J}(Y)^2] - \mathbb{E}_{x,y}[J(X,Y)]^2 \\ &= \mathbb{E}_y[\widehat{J}(Y)^2] - \mathbb{E}_y[\widehat{J}(Y)]^2 \text{ from eq. (24)} \\ &= Var(\widehat{J}(Y)) \end{aligned} \tag{27}$$

where the third line follows from the convexity of $(\cdot)^2$ and Jensen's Inequality.

The IDB REINFORCE gradient can be more efficiently calculated using the BB step function. Letting R_{t_ℓ} denote the reward for timestep t_ℓ whenever $B_t = 1$, all remaining R_t have reward zero. Thus

$$\begin{aligned} \frac{\partial R}{\partial \theta} &= \frac{\partial \log P(L)}{\partial \theta} + \frac{\partial}{\partial \theta}\mathbb{E}_{b|L}\left[\log P(y|b)\right] \\ &= \frac{\partial \log P(L)}{\partial \theta} + \sum_{t=1}^{T} \mathbb{E}_{b_{\leq t}|L}\left[\frac{\partial R_t}{\partial \theta} + R_t \frac{\partial}{\partial \theta}\log P(b_{\leq t}|L - t_\ell)\right] \\ &= \frac{\partial \log P(L)}{\partial \theta} + \sum_{\ell=1}^{L} \mathbb{E}_{b_{\leq t_\ell}|L}\left[\frac{\partial R_{t_\ell}}{\partial \theta} + R_{t_\ell} \frac{\partial}{\partial \theta}\log P(b_{\leq t_\ell}|L - \ell)\right] \\ &= \frac{\partial \log P(L)}{\partial \theta} + \sum_{\ell=1}^{L} \mathbb{E}_{t_{\leq \ell}|L}\left[\frac{\partial R_{t_\ell}}{\partial \theta} + R_{t_\ell} \frac{\partial}{\partial \theta}\log P(t_{\leq \ell}|L - \ell)\right] \\ &= \frac{\partial \log P(L)}{\partial \theta} + \mathbb{E}_{b|L}\left[\sum_{\ell=1}^{L}\left(\frac{\partial R_{t_\ell}}{\partial \theta} + \left(\sum_{\ell' \geq \ell} R_{t_{\ell'}}\right)\frac{\partial}{\partial \theta}\log P(t_\ell|t_{\ell-1}, L-\ell)\right)\right] \end{aligned} \tag{28}$$

where $R_{t_\ell} = \log P(y_\ell|t_{\leq \ell}, y_{<\ell})$. The $P(t_\ell|\ldots)$ term is recognized as the BB step function of eq. (13). Equation (28) yields identical sample estimates as eq. (23), but requires calculation of far fewer terms.

Equations (23) and (28) allow R_t to condition on the history of Bernoulli trials $b_{<t}$ sampled. For example, $\log P(y_\ell|t_{\leq \ell}, y_{<\ell})$ can be parameterized by a decoder neural network which concatenates together a hidden state from an encoder network from time t_ℓ and an embedding of the previous class label $y_{\ell-1}$ as input to the RNN. Unfortunately, the dependence on $t_{\leq \ell}$ means that $t_{\ell'}$ is not only responsible for reward $R_{t_{\ell'}}$, but also for all rewards succeeding it $R_{t_{\ell'+1}}, R_{t_{\ell'+2}}, \ldots$. For this reason, $\frac{\partial}{\partial \theta}\log P(t_\ell|t_{\ell-1}, L-\ell)$ will tend to receive higher magnitude updates than $t_{\ell+1}$, which we expect to increase the variance of the estimator.

We can make the estimator more "local" if we make a conditional independence assumption $P(y_\ell|t_{\leq\ell}, y_{<\ell}) = P(y_\ell|t_\ell, y_{<\ell})$. This costs us, for example, the ability to feed an encoder hidden state as part of the input to a decoder RNN since that produces an implicit dependence on all $t_{\leq\ell}$. Our example decoder may still, however, condition its output on a hidden state at time t_ℓ and previous class labels $y_{<\ell}$, similar to [26].

Deriving the new estimator is similar to before. Letting $R_{t_\ell} = \log P(y_\ell|t_\ell, y_{<\ell})$,

$$\begin{aligned}
\frac{\partial R}{\partial \theta} &= \frac{\partial \log P(L)}{\partial \theta} + \sum_{\ell=1}^{L} \frac{\partial}{\partial \theta} \mathbb{E}_{b|L}[R_{t_\ell}] \\
&= \frac{\partial \log P(L)}{\partial \theta} + \sum_{\ell=1}^{L} \frac{\partial}{\partial \theta} \mathbb{E}_{t_\ell|L-\ell}[R_{t_\ell}] \\
&= \frac{\partial \log P(L)}{\partial \theta} + \mathbb{E}_{b|L}\left[\sum_{\ell=1}^{L}\left(\frac{\partial R_{t_\ell}}{\partial \theta} + R_{t_\ell}\frac{\partial}{\partial \theta}\log P(t_\ell|L-\ell)\right)\right]
\end{aligned} \tag{29}$$

where $P(t_\ell|L-\ell)$ is the marginal probability of the t being the ℓ-th BB-drafted sample, i.e. eq. (15). We call this estimator the Marginal Bounded Bernoulli (MBB) REINFORCE estimator. Equation (15) is similar to eq. (28) but only multiplies the log-probability of the t_ℓ-th draft (not all $t_{\leq\ell}$ drafts) with its local reward R_{t_ℓ}. This should make the magnitude of the update more uniform with respect to the timestep.

When we make the conditional independence assumption above, we can prove that the MBB estimator has variance no greater than that of the IDB estimator. The proof is similar to that showing the IDB estimator has no greater variance than that of the CB estimator. First, we split the expectation of the IDB estimator into two random variables $Z = X + Y$:

$$\begin{aligned}
\mathbb{E}_{b|L}[Z] &= \mathbb{E}_{b|L}\left[\sum_{\ell=1}^{L}\left(\frac{\partial R_{t_\ell}}{\partial \theta} + R_{t_\ell}\frac{\partial}{\partial \theta}\log P(t_{\leq\ell}|L-\ell)\right)\right] \\
&= \mathbb{E}_{b|L}\left[\sum_{\ell=1}^{L}\left(\frac{\partial R_{t_\ell}}{\partial \theta} + R_{t_\ell}\frac{\partial}{\partial \theta}\log P(t_\ell|L)\right) + \right. \\
&\qquad \left. \sum_{\ell=1}^{L}\left(R_{t_\ell}\frac{\partial}{\partial \theta}\log P(t_{<\ell}|t_\ell, L-\ell)\right)\right] \\
&= \mathbb{E}_{b|L}[Y + X] \\
&= \mathbb{E}_{b|L}[Y] + \mathbb{E}_{b|L}[X]
\end{aligned} \tag{30}$$

Where we note that the expectation over Y is that of eq. (29). We are left to

prove, once again, that $\mathbb{E}_{b|L}[X] = 0$.

$$\begin{aligned}
\mathbb{E}_{b|L}[X] &= \mathbb{E}_{b|L}\left[\sum_{\ell=1}^{L}\left(R_{t_\ell}\frac{\partial}{\partial\theta}\log P(t_{<\ell}|t_\ell, L-\ell)\right)\right] \\
&= \sum_{\ell=1}^{L}\mathbb{E}_{t_{\leq\ell}|L}\left[R_{t_\ell}\frac{\partial}{\partial\theta}\log P(t_{<\ell}|t_\ell, L-\ell)\right] \\
&= \sum_{\ell=1}^{L}\mathbb{E}_{t_\ell|L}\left[\mathbb{E}_{t_{<\ell}|t_\ell,L-\ell}\left[R_{t_\ell}\frac{\partial}{\partial\theta}\log P(t_{<\ell}|t_\ell, L-\ell)\right]\right] \\
&= \sum_{\ell=1}^{L}\mathbb{E}_{t_\ell|L}\left[R_{t_\ell}\frac{\partial}{\partial\theta}\mathbb{E}_{t_{<\ell}|t_\ell,L-\ell}[1]\right] \\
&= 0
\end{aligned} \tag{31}$$

The remainder of the proof identically follows from eqs. (26) and (27). We emphasize that this only holds when R_ℓ is memoryless ($R_\ell \perp\!\!\!\perp t_{<\ell}|t_\ell$),. If R_ℓ is not memoryless eq. (29) is not an unbiased estimator of the total reward.

Once we have made the conditional independence assumption required for the MBB, however, we can efficiently calculate the exact expectation using dynamic programming. We will discuss how in section 2.4. The MBB may still be preferred over the exact form if the cost to compute R_ℓ is prohibitively expensive.

2.3 Continuous relaxations

A continuous relaxation is a continuous random variable that approximates (relaxes) some discrete random variable. Of particular note is the Concrete/Gumbel-SoftMax distribution [18, 15], which approximates a categorical random variable $B \in [1, N]$ with odds $\{w_n\}_{n\in[1,N]}$, Gumbel noise $G_n = -\log(-\log U_n), U_n \sim Uniform(0, 1)$, and a scalar temperature $\lambda \in \mathbb{R}^+$. The Concrete random variable $Z \in \{x \in [0, 1]^N; \sum_n x_n = 1\}$ is defined as

$$Z_n = \frac{\exp((\log w_n + G_n)/\lambda)}{\sum_{n'=1}^{N}\exp((\log w_{n'} + G_{n'})/\lambda)} \tag{32}$$

A categorical sample $B \sim P(n; N)$ can be recovered from a Concrete sample in two equivalent manners. First, by the Gumbel-Max trick [28]:

$$P(\forall n'.Z_n \geq Z_{n'}) = \frac{w_n}{\sum_{n'=1}^{N} w_{n'}} = P(B = n) \tag{33}$$

Which implies that $B = H(Z) = \arg_n \max(Z_n)$ is a Categorical sample. Alternatively, Z approaches a one-hot representation of B as $\lambda \to 0$:

$$P(\lim_{\lambda\to 0} Z_n = 1) = \frac{w_n}{\sum_{n'=1}^{N} w_{n'}} = P(B = n) \tag{34}$$

When $N = 2$, $P(B = n)$ is Bernoulli, the Concrete variable is defined as

$$Z = \frac{1}{1 + \exp(-(\log w + D)/\lambda)}, D = \log U - \log(1 - U) \tag{35}$$

and the deterministic mapping $B = H(Z) = I[Z > 0.5]$.

Using the mapping $\prod_{a \in A} w_a = w'$, the CB can be considered a categorical distribution and suitable for a Concrete relaxation. Unfortunately, using this mapping directly would convert an N-length vector of weights w_n to a vector of N-choose-k weights, which is intractable for large N. The numerator in eq. (32) cannot be teased into a combination of random variables $W_1(w_1), W_2(w_2), \ldots$, because the Gumbel noise G_n, which would now represent the combination of noise of the W_a terms, would no longer be independent of $G_{n'}, n' \neq n$. Thus, the CB is not directly suited to continuous relaxation.

We can, however, relax the CB indirectly by relaxing the intermediate variables defined in section 2.1. The IDB can be relaxed as a sequence of Bernoulli relaxations of eq. (35) according to the recursive step eq. (11). The BB can be relaxed into a sequence of categorical relaxations of eq. (32) according to the draft eq. (13). Finally, the marginal probability of t_ℓ under the BB (eq. (15)) is just one categorical relaxation per label ℓ.

When the objective can be reframed in terms of the relaxation Z, a network can start by optimizing a high temperature λ, then slowly lower it over the course of training so that Z approaches the discrete distribution. At test time, the deterministic mapping $H(Z)$ can be used. For our objective, a relaxed emission does not make sense. We need to come up with L distinct distributions for each of the class labels y_ℓ.

We focus on two uses of continuous relaxations with a discrete objective. The first is to use a RELAX-based gradient estimator [11]. RELAX-based gradient estimators augment the REINFORCE estimator with some additional terms that are intendent to reduce its variance. Letting B be a discrete random variable of a continuous relaxation Z, the gradient of the expected value of some f (where f can be a reward, e.g.) is defined as

$$\frac{\partial \mathbb{E}_b[f(b)]}{\partial \theta} = \mathbb{E}_b \left[\left(f(b) - \mathbb{E}_{z|b}[\gamma(z)] \right) \frac{\partial \log P(b)}{\partial \theta} - \frac{\partial \mathbb{E}_{z|b}[\gamma(z)]}{\partial \theta} \right] + \frac{\partial \mathbb{E}_z[\gamma(z)]}{\partial \theta} \tag{36}$$

Where $\gamma(z)$ is a control variate, e.g. a neural network trained on the values of the relaxation to minimize the difference between the objective $f(b)$ and itself. $P(z|b)$ is the truncated distribution over Z such that the value of Z obeys the relationship $H(Z) = b$. If $\gamma(z)$ is the concrete distribution parameterized by a learnable λ, eq. (36) is the REBAR gradient [23].

RELAX-style estimators can be paired with eqs. (23) and (29). Equation (23) is preferred over eq. (28) as the latter would involve infinite values in the relaxed categorical draft for $t \leq t_{\ell-1}$. Each Bernoulli in the IDB format has a real relaxation except when $T - t = L - \ell_t$, at which point $\log P(b_{\leq t} | \ldots) = 0$ and hence does not need a baseline. Equation (29) is always real.

The second is the so-called Straight-Through (ST) estimator [4, 15]. An ST estimator uses the discrete sample $H(X)$ during the forward pass, and estimates the partial derivative of $H(X)$ in the backward pass with that of X, i.e. $\frac{\partial H(X)}{\partial \theta} \approx \frac{\partial X}{\partial \theta}$. This estimator is biased, but can work well in practice. If we output a one-hot representation $H(X^{(\ell)}) = b^{(\ell)} \in \{0,1\}^T, b_t^{(\ell)} = 1_{t=n^{(\ell)}}$ for the ℓ-th drafted (DB or BB) sample, adding them together $b = \sum_{\ell=1}^{L} b^{(\ell)}$ produces our CB sample. If we substitute $\frac{\partial b_t^{(\ell)}}{\partial \theta} \approx \frac{\partial X_t^{(\ell)}}{\partial \theta}$ then $\frac{\partial b_t}{\partial \theta} = \sum_\ell \frac{\partial b_t^{(\ell)}}{\partial \theta}$ is well-defined. Alternatively, we can construct b by concatenating together the relaxed Bernoulli trials of the IDB, $b = [b^{(1)}, b^{(2)}, \ldots, b^{(T)}], b^{(t)} = H(X^{(t)})$. Again, the partial derivatives are well-defined: $\frac{\partial b_t}{\partial \theta} = \frac{\partial b^{(t)}}{\partial \theta}$. From there, we maximize the likelihood of the data using the conditional distribution derived from eq. (20):

$$P(y|b, L) = \prod_{t=1}^{T} P(y_{\ell_t}|h_t, b_{\leq t})^{b_t} \tag{37}$$

where h_t is a hidden state of the network at timestep t. Conditioning on $b_{\leq t}$ is implicit in the definition of y_{ℓ_t}, though this conditioning is ignored by the ST estimator.

2.4 Exact expectations

At the end of section 2.2, we mentioned that we can marginalize out the Bernoulli latent variables efficiently, assuming $P(y_\ell|t_{\leq \ell}, y_{<\ell}) = P(y_\ell|t_\ell, y_{<\ell})$. Further, it must be feasible to calculate that probability for all permutations of t and ℓ. In the case of the model proposed by [17], the distribution $\log P(y_t|t_\ell)$ is calculated by a simple linear transformation of the RNN hidden state h_t followed by a SoftMax. These calculations can be parallelized across t and are fully differentiable. The decoder structure of [26] is also a candidate as the distribution $P(y_\ell|t_\ell, y_{<\ell})$ is a simple two-layer feed-forward neural network on the combination of an encoder and a decoder hidden state.

Starting from eq. (18) and making the conditional independence assumption between t_ℓ and $t_{\ell-1}$, we manipulate $P(y)$ into a form suitable for dynamic

programming.

$$
\begin{aligned}
P(y) &= P(L) \sum_b P(b|L) P(y|b) \\
&= P(L) \sum_b P(b|L) \prod_{\ell=1}^{L} P(y_\ell | b, y_{<\ell}) \\
&= P(L) \sum_{\{t_1, t_2, \ldots, t_\ell\}} P(t_1, t_2, \ldots, t_\ell | L) \prod_{\ell=1}^{L} P(y_\ell | t_\ell, y_{<\ell}) \\
&= P(L) \sum_{\ell=1}^{L} \sum_{t_\ell = t_{\ell-1}+1}^{T-L+\ell} P(t_\ell | t_{\ell-1}, L-\ell) P(y_\ell | t_\ell, y_{<\ell}) \\
&= P(L) \sum_{\ell=1}^{L} \sum_{t_\ell=1}^{T} P(t_\ell | t_{\ell-1}, L-\ell) P(y_\ell | t_\ell, y_{<\ell})
\end{aligned}
\tag{38}
$$

where the last line follows as $P(t_\ell | t_{\ell-1}, L-\ell) = 0$ when $t_\ell \leq t_{\ell-1}$.

Treating $P(t_\ell | t_{\ell-1}, L)$ as the transition probability between states $t \in [1, T]$ and $P(y_\ell | t_\ell, y_{<\ell})$ as the emission probability, eq. (38) can be considered a Hidden Markov Model. Thus, $P(y)$ can be efficiently calculated using the forward algorithm. The backward step is implicit in the calculation of $P(t_\ell | \ldots)$, which effectively marginalizes over all suffixes.

There exists a relationship between eq. (38) and the CTC objective [14] when the history of class labels $y_{<\ell}$ is conditionally independent of the current class label $P(y_\ell | t_\ell, y_{<\ell}) = P(y_\ell | t_\ell)$. Recalling that $P(L)P(b|L) = P(b)$, the independent Bernoulli probabilities, then define a new distribution over an augmented class label set $\{y'_t\} = \{y_t\} \cup \{\_\}$ as

$$
P(y'_t) = \begin{cases} P(B_t = 0) & y'_t = \_ \\ P(B_t = 1) P(y_t | t) & \text{otherwise} \end{cases}
\tag{39}
$$

where the label "_" acts as a stand-in for choosing not to emit at a given time step. Letting $\beta(y')$ remove all the "_" labels from the augmented label set,

$$
\begin{aligned}
P(y) &= \sum_b \sum_{t=1}^{T} P(b_t) P(y_{\ell_t} | t)^{b_t} \\
&= \sum_{\{c' : |c| = T \wedge \beta(y') = c\}} \sum_{t=1}^{T} P(y'_t | t)
\end{aligned}
\tag{40}
$$

This expression of the data likelihood is almost identical to that of CTC [14], with two restrictions. First, it assumes the distribution over labels factors as described in eq. (39). In general, eq. (39) will lead to different gradient updates than directly parameterizing the augmented vocabulary $P(y'_t)$ since the blank label has its own parameterization. Second, $\beta(y')$ in eq. (40) does not reduce

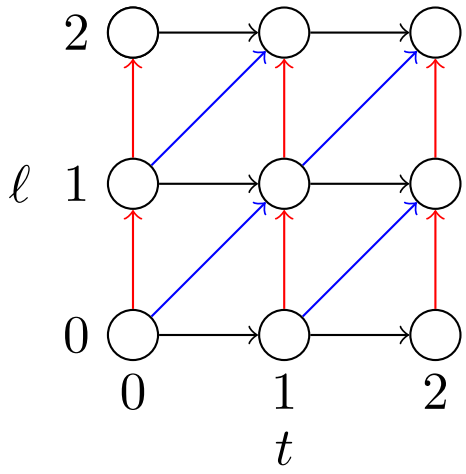

Figure 2: A comparison between RNN-T and eq. (38) forward passes over time (t) and labels (ℓ). The RNN-T steps forward in t or ℓ independently (black and red lines), while the CB steps forward in either t alone or t and ℓ simultaneously (black and blue lines).

repeated labels in c'[4]. Assuming it allows for the non-standard adjustment to β, the data likelihood marginalized over latent Bernoulli sequences can be trivially implemented using an existing CTC loss function. The additional dependency on $y_{<\ell}$ can be considered a generalization of the CTC loss function.

Equation (38) is similar to the RNN Transducer [13] with one key difference: the RNN-T emits labels and absorbs frames in time separately. Figure 2 illustrates the difference between the two methods' forward calculations. To calculate the probability of having emitted the prefix $y_{\leq\ell}$ by time t, the RNN-T sums contributions from either having seen the entire prefix by time $t-1$ and transitioning to time t via a blank label, or having seen $\ell-1$ labels by time t and then emitting y_ℓ (without moving forward in time). Equation (38) maintains the first contribution (transitioning from time $t-1$ to t via an implied blank label), but replaces the latter with a transition that both absorbs a step in time $(t-1 \to t)$ and emits a label $(\ell-1 \to \ell)$.

The choice to absorb time steps and emit labels independently in the RNN-T architecture was motivated partly by the desire to allow sequence transduction between inputs and outputs of arbitrary length. Equation (38) requires $L \leq T$, making it (in its current form) unsuited to tasks where the output is potentially longer than the input, such as machine translation. However, this extra flexibility comes at the cost of uncoupling the events from time. Conceptually,

[4]To the best of our knowledge, there has been no attempt to explore whether the reduction operation leads to any performance benefits over just using the blank label. Graves [12] mention that reducing repeated labels existed prior to the blank label in the formation of the CTC objective.

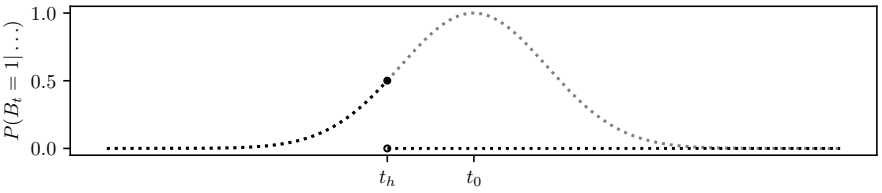

Figure 3: An example of the different distributions of Bernoulli r.v.s B_t around some event that occurred at approximately time t_0, depending on whether $P(B_t|\ldots)$ is conditioned on prior $b_{<t}$ (black) or not (grey). t_h is assumed to be a point where both distributions emit $B_{t_h} = 1$.

the RNN-T models label events as instantaneous, making the path where all labels are output at time $t = 0$ *a priori* equally likely to the path where labels are distributed over time. This has a number of practical ramifications. First, when the problem can be conceptualized as sequential events in time, the RNN-T wastes probability mass on these impossible, instantaneous paths. Second, the monotonicity over t makes heuristic search strategies for approximating $\arg_y \max P(y)$ more efficient and possibly more accurate for eq. (38) than for RNN-Ts[5]. Finally, eq. (38) can be easily parallelized over labels ℓ resulting in $O(T)$ serial operations, a naïve implementation of the RNN-T forward pass requires $O(TL)$ serial operations and a vector-skewed implementation requires $O(T + L)$ serial operations [2].

There are two notable extensions that can be made to eq. (38) that derive from it being a first-order Markov model. First, we can perform an efficient search for $b* = \arg_b \max P(b|y)$ via the Viterbi algorithm. Such a search would be performed after training in order to determine the most likely times $t* = \{t : b^*_t = 1\}$ when events y occurred in a given window of time. Second, we can easily adapt eq. (38) for higher-order models so that $P(y_\ell|\ldots)$ can depend on some arbitrary fixed-length history of emission points $t_\ell, \ldots, t_{\ell-W+1}$. While the number of states that must be stored for a given order grows exponentially with that order, even second-order models may be sufficient to avoid the pitfalls of conditionally independent B_t. We discuss this more in section 3.

3 Approximations

The CB and PB abandon the auto-regressive property of the distribution over Bernoulli trials. This implies each trial is independent though not necessar-

[5]Such searches, most popularly beam search, involve iteratively building prefix paths over time, pruning low-probability paths. The monotonicity w.r.t. t of eq. (38) ensures only $V + 1$ possible extensions of each stored prefix for the next time step, where V is the size of the vocabulary. Because the RNN-T can output as many labels as it wants at a given time, the search space per prefix is technically arbitrarily big, though practically bounded to some constant positive multiple of $V + 1$.

ily identically distributed. In this section, we focus on the case of arbitrary dependencies on past Bernoulli values.

Before continuing, it is worth considering what the auto-regressive property purchases. Auto-regressive models are often employed in contexts where the dependent variables are not latent but observable, such as in language modelling or machine translation. As such, prior claims of the superior performance of data-conditioned models may not translate to latent-conditioned models. Also, auto-regressive networks inhibit parallelization during inference: computations at time t cannot proceed before those at time $t-1$ are completely finished. We can, however, imagine situations where the additional dependencies would be beneficial. For example, fig. 3 illustrates a situation where we are approximating $P(y) \approx P(b, y)$ with some best-path heuristic at test time. Here we assume some event has occurred around time t_0 but neither a model that parameterizes independent B_t (grey curve) nor a model with dependent B_t (black curves) can localize it to a specific time. However, after the first high emission at around t_h, only the auto-regressive model can adjust the probability of future B_t to avoid repeating emissions for the same event. That said, this specific example can be handled without resorting to arbitrary-length dependencies. We may, for example, establish a second-order Markov relationship between y_ℓ and t_ℓ, i.e. $P(y_\ell|t_{\leq\ell}, y_{<\ell}) = P(y_\ell|t_\ell, t_{\ell-1}, y_{<\ell})$, which can easily penalize emissions that are too nearby. Qualifiers notwithstanding, we assume for the rest of this section that an auto-regressive conditional with arbitrarily long history is indeed worthwhile.

We assume $P(b_t|b_{<t})$ to be well-defined and easily calculable, such as the output of a recurrent neural network. The general probability over the number of emissions, $P(L)$, resembles the CB with additional conditional dependence requirements:

$$P(L|b_{<t}) = \sum_{b:\sum_t b_{t'}=L} \left(\prod_{t'=t}^{T} P(b_{t'}|b_{<t'}) \right) \tag{41}$$

where we have allowed conditioning on an arbitrary prefix of Bernoulli trials $b_{<t}$ that do not contribute to the count but can change the conditional probabilities per trial. Making the independence assumption over trials $P(b_t|b_{<t}) = P(b_t) = \frac{w_t}{1+w_t}$ recovers eq. (4).

We can also express the conditional probability mass function of the next Bernoulli trial given the remaining number of highs in terms of the unconditioned probabilities and the distribution over the number of highs:

$$\begin{aligned} P(b_t|b_{<t}, L) &= \frac{P(b_t, L|b_{<t})}{P(L|b_{<t})} \\ &= \frac{P(b_t|b_{<t}) P(L - \sum_{t'=1}^{t} b_{t'}|b_{\leq t})}{P(L - \sum_{t'=1}^{t-1} b_{t'}|b_{<t})} \end{aligned} \tag{42}$$

If we consider $P(\ell|\ldots)$ a normalized, conditional version of the count function $C(\ell, \ldots)$, we can see how an independence assumption would make eq. (42) equivalent to the step distribution of the IDB eq. (11).

If $P(b_t|b_{<t})$ is unique for the given prefix $b_{<t}$, direct calculations of eqs. (41) and (42) are infeasible given the sheer number of paths yielding L emissions (T choose L). To work around this, we have to simplify or approximate either $P(b_t|b_{<t})$ or $P(\ell|\ldots)$. The independence assumption applied is an example of simplifying $P(b_t|b_{<t})$.

This section concerns itself with finding suitable approximations for eqs. (41) and (42). Specifically, we are interested in finding a parameterized family of functions $f(b_{<t}, p_{\leq t}, T, L)$ to approximate $P(B_t = 1|b_{<t}, L)$ and one for the prior $P(L)$ via function family $g(p, T, L)$. The family f only has access to the prefix of samples $b_{<t}$ and some version of those probabilities $p_{\leq t}$ s.t. $p_t = P(B_t = 1|b_{<t})$. f does not have access to future p_t in order to allow for auto-regressive generation of the sample. We assume $p_t \in (0, 1)$ as most neural networks will parameterize p_t using log-odds[6].

Since we plan on using f to sample a value for B_t, it is important that f only be able to sample values that could be taken by B_t under the distribution $P(B_t = 1|b_{<t}, L)$. Critically, we require that f follow the restriction that only L high samples may be generated total per sequence. Formally, letting $\ell_{<t} = \sum_{t'=1}^{t-1} b_{t'}$, these restrictions can be stated as

1. $f(b_{<t}, p_{\leq t}, T, L) = 1$ if $T - t + 1 = L - \ell_{<t}$.
2. $f(b_{<t}, p_{\leq t}, T, L) = 0$ if $L - \ell_{<t} = 0$.

that is, the probability of sampling $B_t = 1$ becomes 1 when the number of remaining random variables equals the remaining number of highs and becomes 0 when all L highs have already occurred.

In what follows, we discuss an additional restriction that would produce an "ideal" estimator, then discuss some approximators that we can use instead.

3.1 The ideal approximation

Restrictions 1 and 2 are necessary in order to ensure f produces only valid samples under the distribution $P(B_t = 1|b_{<t}, L)$. This ensures that $P(y|b)$ is well-defined. This section concerns itself with what additional restrictions on f and g would be helpful in modelling $P(b|L)$ and $P(L)$.

Restrictions 1 and 2 ensure that the sequence of values b drawn from f are in the domain of $P(b|L)$, but do not ensure that all valid samples from the domain of $P(b|L)$ can be drawn. For convenience, define

$$f^{b_t}(b_{<t}, p_{\leq t}, T, L) = \begin{cases} f(b_{<t}, p_{\leq t}, T, L) & b_t = 1 \\ 1 - f(b_{<t}, p_{\leq t}, T, L) & b_t = 0 \end{cases} \tag{43}$$

The following restriction ensures the domains of f and g are nonzero for all nonzero b of $P(b|L)$ and $P(L)$ resp.

[6] In addition, the only family of approximators that ensure only valid b could be sampled is a degenerate one that outputs $B_t = 1$ whenever $p_t > 1$ and $\sum b_{<t} < L$.

3. $P(L) > 0 \wedge P(b|L) > 0 \implies g(p, T, L) > 0 \wedge \prod_{t=1}^{T} f^{b_t}(b_{<t}, p_{\leq t}, T, L) > 0$

Without *a priori* reason for excluding certain samples from the domain of f, not enforcing this restriction would produce situations in inference where the behaviour of $P(B_t)$ is undefined.

While it is necessary to sample from $P(b|L)$ in order to sample b with exclusively L highs, we ultimately only care about matching $P(b)$ at test time. Instead of requiring $P(L) = g(p, T, L)$ and $P(B_t = 1|b_{<t}, L) = f(b_{<t}, p_{\leq t}, T, L)$, g and p could be some other prior and conditional such that their product is $P(b)$. Formally

4. $g(p, T, L) = g' > 0 \wedge \prod_{t=1}^{T} f^{b_t}(b_{<t}, p_{\leq t}, T, L) = f' > 0 \implies f'g' = P(b)$

With restrictions 1-4 we could, for example, derive an unbiased estimator for $P(y)$ with samples over f and corrected by g in a form similar to eq. (23).

The distribution over b induced by the forced-suffix mechanism from section 1 is an example of an approximator which satisfies restrictions 1-3. If we make f only assign one b a nonzero probability and set $g' = P(b)$, we can satisfy restrictions 1, 2, and 4. However, restrictions 1-4 cannot be met simultaneously. We prove this by counterexample. Choose $T = 2$ and $L = 1$. Note that $B_1 = 1 \iff B_2 = 0$. This means $f(b_1, p_{\leq 2}, 2, 1) = I[b_1 = 0]$ by restrictions 1-2. Let $f(\emptyset, p_1, 2, 1) = \alpha$. $B_1 = 1$ with probability α. Plugging in the samples $b \in \{0, 1\}^2$ into restriction 3 gives us a system of two equations with two unknowns (α and $h(\cdot)$):

$$\begin{aligned} g(p, 2, 1)\alpha &= p_1(1 - p_2) \\ g(p, 2, 1)(1 - \alpha) &= (1 - p_1)p_2 \end{aligned}$$

The system has unique solutions

$$\begin{aligned} \alpha &= \frac{p_1 - p_1 p_2}{p_1 - 2p_1 p_2 + p_2} \\ g(p, 2, 1) &= p_1 - 2p_1 p_2 + p_3 \end{aligned}$$

which are both valid probabilities. However, the solution to α is a function of p_2, which is not an input to $f(\emptyset, p_1, 2, 1)$. α must be constant for $p = (p_1, p_2)$ and $p' = (p_1, p_2')$, $p_2 \neq p_2'$, which means it will not satisfy restriction 4 for all choices of p.

This implies there are no f and g that can be decomposed nicely into $P(b|L)$ and $P(L)$ respectively. In the above counterexample, p_2 and p_2' represent the marginal probabilities of the second Bernoulli trial across two different distributions p and p'; they are not conditioned on b_1. Thus we cannot make the decomposition even with the independence assumption, i.e. $P(b_t|b_{<t}) = P(b_t)$.

Given that 1-4 cannot be satisfied simultaneously, and 1-2 are necessary, we must decide whether to drop restriction 3 or 4. We believe that it is more important that the behaviour of our optimized distribution $P(b)$ at test time be well-defined (though biased) than it is to be unbiased for some choices of b but

undefined for others. While we will present a workaround for restriction 4, the same trade-off will resurface shortly.

While no g can act as a proper prior $P(L)$, we can still recover the unconditioned probability of a sample b by relaxing the constraints on g. In particular, if we allow g to depend on the sample b, finding g to satisfy the analogue to restriction 4 is trivial:

$$g(b, p, T) = P(b) / \prod_{t=1}^{T} f^{b_t}(b_{<t}, p_{\leq t}, T, L) \tag{44}$$

from which, if we move the "prior" into the expectation, we get

$$P(y) = \mathbb{E}_{b\sim f}\left[g(b, p, T)P(y|b)\right] \tag{45}$$

which can be considered importance sampling [6].

While eq. (45) is unbiased, it is high variance and its gradients suffer from numerical underflow. Instead, we take the log and use Jensen's Inequality:

$$\begin{aligned}
\log P(y) &= \log \mathbb{E}_{b\sim f}\left[g(b, p, T)P(y|b)\right] \\
&\geq \mathbb{E}_{b\sim Q}\left[\log \frac{P(y, b)}{Q(b)}\right] \\
&= \mathbb{E}_{b\sim Q}\left[\log P(y) + \log \frac{P(b|y)}{Q(b)}\right] \\
&= \log P(y) - \mathcal{D}(Q\|P)
\end{aligned} \tag{46}$$

where we have applied eq. (44) and defined $Q(b) = \prod_{t=1}^{T} f^{b_t}(b_{<t}, p_{\leq t}, T, L)$ as the proposal distribution. $\mathcal{D}(Q\|P)$ is the KL-divergence from $P(b|y)$ to $Q(b)$. This is the same lower bound used in Variational Inference (VI). Since both $P(b)$ and $Q(b)$ factor over time, we can apply a similar REINFORCE estimator to eq. (23) and that in Lawson et al. [16], but with the additional restrictions 1-3 on $Q(b)$. The VI lower bound of eq. (46) hopefully simultaneously maximizes $P(y|b)$ and forces $Q(b) \to P(b|y)$.

If we consider the right-hand side of eq. (46) an estimator for $\log P(y)$, we can see that the estimator is biased unless both sides are equal. This can only occur when the KL-divergence between $Q(b)$ and $P(b|y)$ is zero. This optimal state cannot be reached in general. This fact follows almost immediately from the above proof. Choose $P(y|b) = I[\sum_t b_t = L]P(y)$ which results in $P(b|y) = I[\sum_t b_t = L]P(b)$. Allowing $\sum_t b_t = L$ and taking $g \equiv 1$ in the above proof, we have already shown $Q(b) \not\equiv P(b)$ if we are simultaneously satisfying restrictions 1-3[7]. Thus, once again, we must violate either restriction 3 or 4.

Unlike the REINFORCE lower bounds, the variational lower bound eq. (46) does not have an explicit prior. The prior is implicit in the log-probability ratio. By setting $Q(b) = P(b|L)$, The ratio $P(b)/Q(b) = P(L)$, and we recover

[7]This proof clearly holds if we also allow Q to depend on y, since our choice of $P(y|b)$ precludes any useful information from y except its length.

the REINFORCE bound. This formulation also makes clear the bias in our REINFORCE-style estimates: we are approximating $P(b|y)$ with $P(b|L)$. In section 4, we show how the generalized CB of eq. (42) is well-suited to modelling $P(b|y)$ as well.

We can mitigate the bias in eq. (46) by taking the mean ratio over many samples inside the log [6]:

$$\log P(y) \geq \mathbb{E}_{b^{(1)},b^{(2)},\ldots,b^{(K)}\sim Q}\left[\log \frac{1}{K}\sum_k \frac{P(y,b^{(k)})}{Q(b^{(k)})}\right] \geq \mathbb{E}_{b\sim Q}\left[\log \frac{P(y,b)}{Q(b)}\right] \tag{47}$$

A sum within the log term precludes factorizing $P(b)$ and $Q(b)$ over time. Thus, the decrease in bias of eq. (47) is at the cost of the higher variance of a global reward function. The same technique can be used for the REINFORCE estimates of section 2.2.

3.2 Proposal distributions

Assuming we are optimizing an approximation to eq. (42) via the variational lower bound eq. (46). We now go over some approximations that satisfy restrictions 1-3 from earlier in this section.

The forced-suffix distribution from [17, 16], done properly, is one such candidate. It is the easiest and fastest approximation to calculate:

$$Q_{FS}(B_t = 1|b_{<t}, L) = \begin{cases} 1 & T-t+1 = L-\ell_{<t} \\ 0 & \ell_{<t} = L \\ \frac{w_t}{1+w_t} & \text{otherwise} \end{cases} \tag{48}$$

While eq. (48) will tend to oversample highs at the beginning of the sequence (see section 1), the log-ratio between P and Q will hopefully penalize this tendency (the ratio only leaves the log probability of the suffix $P(b_{>t_L}|b_{\leq t_L})$). We suspect that eq. (48) will be slow to converge because it makes no attempt at approximating the future.

The remaining approximators take advantage of the following form of eq. (42):

$$\begin{aligned} P(b_t|b_{<t}, L) &= \frac{P(b_t|b_{<t})P(L-\ell_t|b_{\leq t})}{P(L-\ell_{<t}|b_{<t})} \\ &= \frac{P(b_t|b_{<t})P(L-\ell_t|b_{\leq t})}{\sum_{v\in\{0,1\}} P(B_t=v|b_{<t})P(L-\ell_{<t}-v|[b_{<t},v])} \\ &= \frac{w_t^{b_t}P(L-\ell_t|b_{\leq t})}{P(L-\ell_{<t}|[b_{<t},0]) + w_t P(L-\ell_{<t}-1|[b_{<t},1])} \end{aligned} \tag{49}$$

where $P(B_t = 1|b_{<t}) = w_t/(1+w_t)$.

This form shows us how to derive an approximation of eq. (42) using an approximation of eq. (41) instead. It also factors out the contribution of the log-odds w_t so that the remaining approximations can be parameterized independently.

Restated, $P(L|b_{<t})$ is the probability that $T-t+1$ possibly dependent Bernoulli random variables B_t sum to L. A natural choice for an approximation to this distribution is to replace it with one where those variables are independent. This is simply the PB distribution. Assume that we have the log-odds of T independent Bernoulli random variables $w'_1, w'_2, \dots w'_T$. Then we approximate $P(L-\ell_{<t}-v|[b_{<t}, v])$ with the probability of $L-\ell_{<t}-v$ highs from the parameters $w'_{t+1}, w'_{t+2}, \dots, w'_T$. Plugging this approximation into eq. (49) gives:

$$\begin{aligned} Q_{IDB}(B_t = 1|b_{<t}, L) &= \frac{w_t Q(L-\ell_{<t}-1; w'_{>t})}{Q(L-\ell_{<t}; w'_{>t}) + w_t Q(L-\ell_{<t}-1; w'_{>t})} \\ &= \frac{w_t C(L-\ell_{<t}-1, [t+1,T]; w')/Z}{C(L-\ell_{<t}, [t+1,T]; w')/Z + \\ \qquad & \qquad w_t C(L-\ell_{<t}-1; [t+1,T]; w')/Z} \\ &= \frac{w_t C(L-\ell_{<t}-1, [t+1,T]; w' \cup \{w_t\} \setminus \{w'_t\})}{C(L-\ell_{<t}, [t,T]; w' \cup \{w_t\} \setminus \{w'_t\})} \end{aligned} \tag{50}$$

where $Z = \prod_{t'=t+1}^{T}(1+w'_{t'}) = \sum_{\ell=0}^{T} C(\ell, [1,T]; w')$. In fact, eq. (50) is merely the IDB step function of eq. (11) with the independent log-odds for time step t, w'_t, replaced with that of the dependent log-odds w_t. Note that the dependence of Q on w_t, which can only be derived by sampling from B_{t-1}, requires the use of an estimator of the form of eq. (23) rather than eq. (28) or eq. (29).

Intuitively, the closer the individual independent probabilities in the approximation are to the true dependent probabilities, the closer eq. (50) will be to eq. (41). Indeed, this result has been previously proven with exact error bounds on the approximation [21].

To generate the independent probabilities, one could train an auxiliary, non-autoregressive neural network on the same data using a CB-based objective from section 2.2 or section 2.4, but the success of such a network in predicting $P(L|b_{<t})$ would signal that the main network is not effectively using the auto-regressive property. Instead, we propose training this auxiliary non-autoregressive model alongside the main network using the gradients from eq. (46). This approach also allows us to condition our distribution on the target output labels y by injecting them into the network as input, for example through an attention layer [3]. The conditioning on y may allow $Q_{IDB}(b_t|b_{<t}, y) \to P(b_t|b_{<t}, y)$. Finally, we can have the model produce L distinct sets of odds ${w_t^{(\ell)}}_t$ which can be substituted into the count functions $C(\cdot)$ of eq. (50) depending on the number of remaining labels. This modification can help deal with situations such as that illustrated in fig. 3.

Training an auxiliary network simultaneously to approximate Q is both resource-intensive and strongly biased at the beginning of training. A coarse-grained alternative to using the PB is to make even stronger assumptions on the distribution $P(L|b_{<t})$ by forcing Bernoulli random variables to be identically distributed. That is, use a binomial distribution. The maximum likelihood es-

timate for binomial parameter p is L/T. The resulting approximation is

$$\begin{aligned} Q_{Bin}(B_t = 1|b_{<t}, L) &= \frac{w_t \binom{T-t}{L-\ell_{<t}-1} p^{L-\ell_{<t}-1}(1-p)^{T-t-L+\ell_{<t}+1}}{\sum_{v\in\{0,1\}} w_t^v \binom{T-t}{L-\ell_{<t}-v} p^{L-\ell_{<t}-v}(1-p)^{T-t-L+\ell_{<t}+v}} \\ &= \frac{w_t(T-L)(L-\ell_{<t})}{w_t(T-L)(L-\ell_{<t}) + L(T-t-L+\ell_{<t}+1)} \end{aligned} \tag{51}$$

Substituting $\ell_{<t} = L$ and $L - \ell_{<t} = T - t + 1$ will show that eq. (51) satisfies restrictions 1 and 2. While we expect the binomial approximation to speed up initial training, Q_{Bin} will will suffer as training progresses and the individual $P(b_t|b_{<t}, y)$ move away from L/T.

4 Conditioning on class labels

In section 3.1, we mentioned that estimating $P(b|L)$ instead of $P(b|y)$ introduces a bias into our estimators, even in the conditionally independent case. In this section, we discuss the extent of the mismatch, and what we can do about it.

Conditioning on y results in at least as much information gain, likely more, on $P(b)$ as L since the value of the random variable L can be determined determined by the size of y, $L = |y|$, whereas we usually cannot infer y from L. It follows that $P(b|L)$ is only a coarse-grained estimate of $P(b|y)$ and, given the option, we would rather an unbiased estimate of $P(b|y)$.

One might wonder, then, if our discussion of approximators in section 3 is based off a false premise: that is, the focus on $P(b|L)$ makes assumptions that are inapplicable or just plain wrong about $P(b|y)$. We can answer immediately in the negative with respect to restrictions 1 and 2 as they ensure that either approximation, to $P(b|y)$ or to $P(b|L)$, is well-defined. The third restriction, namely that if a sample is assigned nonzero $P(L)$ and $P(b|L)$, then the approximator should also assign it nonzero $P(b|L)$, does not directly follow. There might exist some y, $|y| = L$, such that $P(L) > 0$ and $P(b|L) > 0$, but $P(y) = 0$ ($P(b|y)$ is undefined). Under eq. (20), however, this would require some $P(y_\ell|b_{\leq t_\ell}, y_{<\ell}) = 0$ for all valid prefixes $b_{\leq t_\ell}$, which we do not assume *a priori* and usually cannot occur when parameterizing that distribution via neural network. The only remaining restriction is the fourth. We have previously addressed its (un)satisfiability when extended to $P(b|y)$ in section 3.1.

What is left is to justify why we are choosing to focus on proposal distributions modelling eqs. (41) and (42) instead of $P(y)$ and $P(b|y)$. The answer comes from the fact that $P(b|y)$ and $P(y)$ are constructed identically (in a combinatorial sense) to $P(b|L)$ and $P(L)$. That is, to construct e.g. $P(y)$, we count weights in the exact same way as we would $P(L)$ - only the weights themselves have changed. Formally, we prove that for any *fixed* y distributed as the sum over the joint y, b under eq. (20), there exists some $P^y(b|L)$ distributed under eq. (42) such that $P(b|y) \equiv P^y(b|L)$.

The proof of this derives from the recursive definitions of $P(y)$, which appears very similar to that of $P(L)$. Assuming we have a prefix of Bernoulli values $b_{<t}$, the probability of some suffix $y_{>\ell_{t-1}}$ is derived from the contributions of two possible conditions for time t: either $B_t = 1$ and $Y_t = y_{\ell_t}$ and the future steps $t' > t$ must account for the remainder of the suffix $y_{>\ell_{t-1}+1}$; or $B_t = 0$ and the future steps must account for all of the suffix $y_{>\ell_{t-1}}$. For brevity of expression, define $v' = y_{\ell_{t-1}+1}$ to be the first element of the suffix, $r = y_{>\ell_{t-1}+1}$ the rest of the suffix, $p = y_{\leq \ell_{t-1}}$ the prefix, and function $f : \{0, 1\} \to \mathbb{R}$ as

$$f_{b_{<t}}(x) = P(v'|p, b_{<t}, B_t = x)^x P(B_t = x|p, b_{<t}) \tag{52}$$

where we recognize $f_{b_{<t}}(b_t)$ as a multiplicand in eq. (20). f is not necessarily a distribution since it is missing the probability mass $\sum_{v \neq v} P(v|p, b_{<t}, B_t = 1)P(B_t = 1|p, b_{<t})$. Using f, v', r, and p, we can easily express the recursive definition of the probability of y:

$$P(v', r|b_{<t}, p) = f_{b_{<t}}(1)P(r|b_{<t}, B_t = 1, p, v') + f_{b_{<t}}(0)P(v', r|b_{<t}, B_t = 0, p) \tag{53}$$

One can verify that eq. (53) matches $\sum_b P(b, y)$ when $[v', r] = y$ and $b_{<t} = \emptyset$. By comparing eqs. (41) and (53), we can see that our earlier statement that $P(y)$ and $P(L)$ are are constructed in the same way is true: the recursive definitions construct a binary trees where at each node we decide whether to emit. We exploit this to prove that there exists some $P^y(|r| + 1|b_{<t})$ proportional to $P(v', r|b_{<t}, p)$. Denoting all the proportional distributions with P^y (the superscript y is used because proportionality only applies to the given y), we prove that $\forall y, P(y) : \exists P^y(|y|) : \forall t \in [1, T], b_{<t} : \exists \alpha_{b_{<t}} : \forall p, v', r : P(v', r|b_{<t}, p) = \alpha_{b_{<t}} P^y(L - |p||b_{<t})$ via induction on decreasing t.

In the base case, $t = T$, we consider three distinct cases: the suffix is of size 0, 1, or greater than 1. When the suffix is empty, we cannot take the first element v' from the suffix, making $P(r|\ldots)$ on the r.h.s. of eq. (53) 0. $P(v', r|\ldots)$ on the r.h.s. equals 1 because there are no more paths to recurse on. Thus $P(\emptyset|b_{<T}, y) = f_{b_{<T}}(0)$. When the suffix is of size 1, the reverse is true, making $P(y_L|b_{<T}, y_{<L}) = f_{b_{<T}}(1)$. Finally, any suffix of size greater than one cannot occur because we can only emit at most one more time, making $\forall v', r \neq \emptyset : P(v', r|b_{<T}, p) = 0$. Determining our proportional $P^y(L|b_{<T})$ is straightforward: letting $\alpha_{b_{<T}} = f_{b_{<T}}(0) + f_{b_{<T}}(1)$, define $P^y(B_T = x|b_{<T}) = Z^{-1}_{b_{<T}} f_{b_{<T}}(0)$. The induced distribution $P^y(L|b_{<T})$ is proportional to $P(v', r'|b_{<T}, p)$ with constant $\alpha_{b_{<T}}$.

The inductive case is straightforward:

$$\begin{aligned}
P(v', r|b_{<t}, p) =& f_{b_{<t}}(1)P(r|b_{<t}, B_t = 1, p, v') + f_{b_{<t}}(0)P(v', r|b_{<t}, B_t = 0, p) \\
=& f_{b_{<t}}(1)\alpha_{[b_{<t}, B_t=1]}P^y(|r||b_{<t}, B_t = 1) + \\
& f_{b_{<t}}(0)\alpha_{[b_{<t}, B_t=0]}P^y(|r| + 1|b_{<t}, B_t = 0) \text{ by I.H.} \\
=& \alpha_{b_{<t}}P^y(B_t = 1|b_{<t})P^y(|r||b_{<t}, B_t = 1) + \\
& \alpha_{b_{<t}}P^y(B_t = 0|b_{<t})P^y(|r| + 1|b_{<t}, B_t = 0) \\
=& \alpha_{b_{<t}}P^y(|r| + 1|b_{<t}) \text{ by eq. (41)}
\end{aligned} \tag{54}$$

where

$$P^y(B_t = b_t|b_{<t}) = \alpha^{-1}_{b_{<t}} f_{b_{<t}}(x)\alpha_{b_{\leq t}}$$
$$\alpha_{b_{<t}} = \sum_{b_t=0}^{1} f_{b_{<t}}(b_t)\alpha_{b_{\leq t}} \tag{55}$$

While proportionality is not sufficient to prove equivalence between $P(y)$ and $P^y(L)$, it is sufficient to ensure $P(b|y)$ is equivalent to $P^y(b|L)$:

$$\begin{aligned} P(B_t = 1|b_{<t}, p, v', r) =& \frac{P(B_t = 1, v', r|b_{<t}, p)}{P(v', r|b_{<t}, p)} \\ =& \frac{f_{b_{<t}}(1)P(r|b_{<t}, B_t = 1, p, v')}{P(v', r|b_{<t}, p)} \\ =& \frac{\alpha_{b_{<t}} P^y(B_t = 1|b_{<t})P^y(|r||b_{<t}, B_t = 1)}{\alpha_{b_{<t}} P^y(|r| + 1|b_{<t})} \\ =& P^y(B_t = 1|b_{<t}, |y|) \text{ by eq. (42)} \end{aligned} \tag{56}$$

as required.

Proving $P^y(b|L) \equiv P(b|y)$ allows us to reframe the question of approximating $P(b|y)$ to one where we are approximating $P^y(b|L)$, which is combinatorically identical to $P(b|L)$. Thus, we may mitigate or even remove the bias (under numerous independence constraints) by transforming the odds of the distribution $Q(b)$ to match those of $P^y(b|L)$. The transformation of $P(B_t = 1|b_{<t})$ into $P^y(B_t = 1|b_{<t})$ is the same as that in section 4.

Unfortunately, $\alpha_{b_{<t}}$, in general, is dependent on the distributions over all suffixes. Even if $P(b_t|b_{<t}) = P(b_t)$, as in section 2.2, the $P(y_{\ell_t}|b_{<t}, y_{<\ell_t})$ term in $f_{b_{<t}}(1)$ will cause $\alpha_{b_{<t}}$ to vary with the suffix. We can limit the history of $P(y_{\ell_t}|\ldots)$ to make calculating $\alpha_{b_{<t}}$ more tractable. For example, if we assume $P(y_{\ell_t}|\ldots) = P(y_{\ell_t}|b_t) = y^{\ell}_t$, we can easily modify the algorithm for computing eq. (8) to count products of $w_t y^{\ell}_t$ instead of just w_t, resulting the analogous CB equations from section 2.1 calculating $P(b|y)$ instead of $P(b|L)$. However, such conditions are just as stringent and involve as many computations as exact marginalization[8]. Thus, for distributions with complicated posteriors over class labels, we are relegated to approximating .

In summary, the distribution $P(b|y)$ always matches a member of the parameterized family of the generalized CB from eq. (42), $P^y(b|L)$, and thus inherits the concerns and restrictions of estimating $P(b|L)$. The additional computations involved in directly computing $P^y(b|L)$ suffer from a combinatorial explosion due to dependencies between class labels, which lends itself to approximation. We leave such approximations to future work.

[8]We could use such a scheme for $Q_{IDB}(b)$ from eq. (50), but it would be easier to simply condition directly on y, $Q_{IDB}(b, y)$, in an effort to parameterize $P^y(b|L)$ instead of $P(b|L)$.

5 Experiments

5.1 Toy problem

Check convergence and variance of different estimators.

Choose a fixed-size sequence T and vocabulary size. Define the population distribution via T binary random variables $\widehat{B}_t \sim P_t(\widehat{b}_t)$ and T categorical random variables $Y_t \sim P_t(y_t|y_j)$. Draw a sequence of categorical random variables Y_t by first sampling $\widehat{B}$ and then adding $Y_t \sim P(y_t|y_{\ell_t-1})$ to the sequence whenever $\widehat{B}_t = 1$. The resulting sequences y of size L and b of size T was sampled with probability

$$P(y, \widehat{b}) = \prod_{t=1}^{T} P_t(y_{\ell_t}|y_{\ell_t-1})^{\widehat{b}_t} P_t(\widehat{b}_t)$$

The goal is to make $Q_t(b_t) \to P_t(b_t)$ and $Q_t(y_t|c_j) \to P_t(y_t|c_j)$ for all $t \in [1, T]$ using y and one of the estimators in section 2.2 or the exact expectation from section 2.4. Letting Q_t and P_t belong to the same parameterized family of statistical models (i.e. Bernoulli or categorical), we can determine the distance between the distributions via mean-squared-error over parameters.

Hyperparameters:

1. Estimators
2. T
3. N (batch size, i.e. number of sequences y)
4. M (Monte-Carlo sample, i.e. number of samples B per y)
5. σ (standard deviation of population parameters)

Should fix the number of trials to something very high. Measure for each sample

1. Sample reward
2. Estimator variance
3. MSE between all P_t and Q_t

5.2 Gigaword Abstractive Summarization, TIMIT, WSJ

Following Raffel et al. [20], we can also get to one or all of these tasks. The models and training are fairly interchangeable between corpora (GGWS needs an additional embedding layer, TIMIT + WSJ might use some language modelling).

Decoder structures:

1. Pointwise (feed-forward from encoder hidden state). Similar to Luo et al. [17], Lawson et al. [16].

2. Autoregressive decoder with encoder hidden state input. Similar to Raffel et al. [20].

3. Monotonic attention with fixed- or variable-sized windows (former similar to Chiu and Raffel [9])

4. Autoregressive decoder with attention, but context vector is only used to produce emission distribution, similar to Wu et al. [26], Wu and Cotterell [25].

`https://github.com/j-min/MoChA-pytorch`

References

[1] Unequal probability exponential designs. In *Sampling Algorithms*, pages 63–98. Springer New York, New York, NY, 2006. ISBN 978-0-387-34240-5. doi: 10.1007/0-387-34240-0_5.

[2] T. Bagby, Kanishka Rao, and Khe Chai Sim. Efficient implementation of recurrent neural network transducer in Tensorflow. In *IEEE Spoken Language Technology Workshop*, SLT '18, pages 506–512, 18. doi: 10.1109/SLT.2018.8639690.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations*, ICLR '15, San Diego, USA, 2015. URL `http://arxiv.org/abs/1409.0473`.

[4] Yoshua Bengio, Nicholas Léonard, and Aaron C. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *CoRR*, abs/1308.3432, 2013. URL `http://arxiv.org/abs/1308.3432`.

[5] Lennart Bondesson, Imbi Traat, and Anders Lundqvist. Pareto Sampling versus Sampford and Conditional Poisson Sampling. *Scandinavian Journal of Statistics*, 33(4):699–720, December 2006. ISSN 0303-6898. doi: 10.1111/j.1467-9469.2006.00497.x.

[6] Yuri Burda, Roger B. Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. In *4th International Conference on Learning Representations*, ICLR '16, San Juan, Puerto Rico, 2016. URL `http://arxiv.org/abs/1509.00519`.

[7] Sean X. Chen and Jun S. Liu. Statistical applications of the Poisson-Binomial and Conditional Bernoulli distributions. *Statistica Sinica*, 7(4):875–892, 1997. ISSN 10170405, 19968507. URL `www.jstor.org/stable/24306160`.

[8] Xiang-Hui Chen, Arthur P. Dempster, and Jun S. Liu. Weighted finite population sampling to maximize entropy. *Biometrika*, 81(3):457–469, 1994. ISSN 00063444. doi: 10.2307/2337119.

[9] Chung-Cheng Chiu and Colin Raffel. Monotonic chunkwise attention. In *6th International Conference on Learning Representations*, ICLR '18, Vancouver, Canada, 2018. URL `https://openreview.net/forum?id=Hko85plCW`.

[10] Artyom Gadetsky, Kirill Struminsky, Christopher Robinson, Novi Quadrianto, and Dmitry Vetrov. Low-variance black-box gradient estimates for the Plackett-Luce Distribution. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, AAAI '20, 2020.

[11] Will Grathwohl, Dami Choi, Yuhuai Wu, Geoffrey Roeder, and David Duvenaud. Backpropagation through the void: Optimizing control variates for black-box gradient estimation. In *6th International Conference on Learning Representations*, ICLR '18, Vancouver, Canada, 2018. URL `https://openreview.net/forum?id=SyzKd1bCW`.

[12] Alex Graves. Connectionist Temporal Classification. In *Supervised Sequence Labelling with Recurrent Neural Networks*, pages 61–93. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-24797-2. doi: 10.1007/978-3-642-24797-2_7.

[13] Alex Graves. Sequence transduction with recurrent neural networks. *CoRR*, abs/1211.3711, 2012. URL `http://arxiv.org/abs/1211.3711`.

[14] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist Temporal Classification: Labelling unsegmented sequence data with recurrent neural networks. In *International Conference on Machine Learning*, ICML '06, pages 369–376, New York, NY, USA, 2006. ACM. ISBN 1-59593-383-2. doi: 10.1145/1143844.1143891.

[15] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-Softmax. In *5th International Conference on Learning Representations*, ICLR '17, Toloun, France, 2017. URL `https://openreview.net/forum?id=rkE3y85ee`.

[16] Dieterich Lawson, Chung-Cheng Chiu, George Tucker, Colin Raffel, Kevin Swersky, and Navdeep Jaitly. Learning hard alignments with variational inference. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, ICASSP '18, pages 5799–5803, April 2018. ISBN 2379-190X. doi: 10.1109/ICASSP.2018.8461977.

[17] Yuo Luo, Chung-Cheng Chiu, Navdeep Jaitly, and Ilya Sutskever. Learning online alignments with continuous rewards policy gradient. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, ICASSP '17, pages 2801–2805, March 2017. doi: 10.1109/ICASSP.2017.7952667.

[18] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The Concrete Distribution: A continuous relaxation of discrete random variables. In *5th International Conference on Learning Representations*, ICLR '17, Toloun, France, 2017. URL https://openreview.net/forum?id=S1jE5L5gl.

[19] Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. In *Proceedings of the 31st International Conference on Machine Learning*, ICML '14, pages 1791–1799. PMLR, January 2014. URL http://proceedings.mlr.press/v32/mnih14.html.

[20] Colin Raffel, Minh-Thang Luong, Peter J. Liu, Ron J. Weiss, and Douglas Eck. Online and linear-time attention by enforcing monotonic alignments. In *International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pages 2837–2846, Sydney, Australia, August 2017. PMLR. URL http://proceedings.mlr.press/v70/raffel17a.html.

[21] Robert J. Serfling. Some elementary results on Poisson approximation in a sequence of Bernoulli trials. *SIAM Review*, 20(3):567–579, July 1978. ISSN 0036-1445. doi: 10.1137/1020070.

[22] Kevin Swersky, Brendan J Frey, Daniel Tarlow, Richard S. Zemel, and Ryan P Adams. Probabilistic n-choose-k models for classification and ranking. In *Advances in Neural Information Processing Systems*, number 25 in NIPS, pages 3050–3058. Curran Associates, Inc., 2012. URL http://papers.nips.cc/paper/4702-probabilistic-n-choose-k-models-for-classification-and-ranking.pdf.

[23] George Tucker, Andriy Mnih, Chris J Maddison, John Lawson, and Jascha Sohl-Dickstein. REBAR: Low-variance, unbiased gradient estimates for discrete latent variable models. In *Advances in Neural Information Processing Systems 30*, pages 2627–2636. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/6856-rebar-low-variance-unbiased-gradient-estimates-for-discrete-latent-variable-models.pdf.

[24] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, May 1992. ISSN 1573-0565. doi: 10.1007/BF00992696.

[25] Shijie Wu and Ryan Cotterell. Exact hard monotonic attention for character-level transduction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, ACL '19, pages 1530–1537, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1148.

[26] Shijie Wu, Pamela Shapiro, and Ryan Cotterell. Hard non-monotonic attention for character-level transduction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, EMNLP '18, pages 4425–4438, Brussels, Belgium, 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1473.

[27] Sang Michael Xie and Stefano Ermon. Reparameterizable subset sampling via continuous relaxations. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, IJCAI '19, pages 3919–3925. International Joint Conferences on Artificial Intelligence Organization, 2019. doi: 10.24963/ijcai.2019/544.

[28] John I. Yellott. The relationship between Luce's Choice Axiom, Thurstone's Theory of Comparative Judgment, and the double exponential distribution. *Journal of Mathematical Psychology*, 15(2):109–144, April 1977. ISSN 0022-2496. doi: 10.1016/0022-2496(77)90026-8.

A Determining the bias of forced emissions for i.i.d. Bernoulli

The goal of this section is to derive eq. (3).

We begin by defining the two "force-points" in the Bernoulli process where it stops sampling like a simple Bernoulli process and forces the suffix to be made of exclusively zeros or ones. Define t_0 as the zero-force point, i.e. the minimal value in $[0, T-1]$ such that all $B_t = 0$ for $t_0 < t < T$. Likewise, define $t_1 \in [0, T-1]$ to be the one-force point where $B_t = 1$ for $t_1 < t < T$. Note that $B_T = 0$ implies t_0 exists and $B_T = 1$ implies t_1 exists. Thus t_0 and t_1 are mutually exclusive events where one or the other must occur for a given sample. The exception is when $T = 0$, at which point both vacuously exist at $t_0 = t_1 = 0$, but we ignore this exception sine we've specified $T > 0$.

We can rephrase the "minimal" requirement of the force points to make it easier to define their distributions. If t_0 being the minimal non-negative integer such that $\forall t \in (t_0, T] B_t = 0$, either $t_0 = 0$ or $B_{t_0} = 1$ (if $B_{t_0} = 0$, then $t_0' = t_0 - 1$ satisfies$\forall t \in (t_0', T]\ B_t = 0$, a contradiction). Likewise, either $t_1 = 1$ or $B_{t_1} = 0$.

By introducing the variables $\ell_t = \sum_{t',1,t} b_t$ as the number of high emissions up to t, we can refine our definition of t_0 and t_1 to refer to the points when the suffix must all be zeros or ones explicitly. More formally, for some fixed-value sample $b \sim P^*(b)$:

$$\exists t_0 \in [0, T-1] \implies \ell_{t_0} = L \wedge (t_0 = 0 \vee b_{t_0} = 1) \tag{57}$$

$$\exists t_1 \in [0, T-1] \implies \ell_{t_1} = L - T + t_1 \wedge (t_1 = 0 \vee b_{t_1} = 0) \tag{58}$$

the conditions on ℓ_t reflect the number of high $b_{t'}$ remaining ($t' > t$). For t_0, no high Bernoulli values can follow, which means we must have previously sampled

all L. For t_1, the number of remaining Bernoulli values must equal the number of remaining labels ($L - \ell_{t_0} = T - t_0$).

Letting $P^*(t_0)$ be the probability over samples b that t_0 exists and is equal to the value t_0:

$$P^*(t_0) = \begin{cases} \binom{t_0-1}{L-1} p^L (1-p)^{t_0-L} & t_0 > 0 \\ I[L=0] & t_0 = 0 \end{cases} \tag{59}$$

Likewise

$$P^*(t_1) = \begin{cases} \binom{t_1-1}{T-L-1} p^{t_1-T+L} (1-p)^{T-L} & t_1 > 0 \\ I[T=L] & t_1 = 0 \end{cases} \tag{60}$$

If t_0 exists, all highs must occur before or at t_0. The first equation can be interpreted as the probability of $L-1$ high Bernoulli values before t_0, then multiplying that with the probability that $B_{t_0} = 1$ (p). If t_1 exists, all *lows* must occur before or at t_1. The second can be interpreted as the probability of choosing $T - L - 1$ low Bernoulli values before t_1, then multiplying that with the probability that $B_{t_1} = 0$.

Because t_0 and t_1 are mutually exclusive when $T > 0$ and one must always occur at a given time, $\sum_{t=0}^{T-1} P^*(t_0 = t) + P^*(t_1 = t) = 1$[9], giving us a distribution over the forced suffixes.

Consider the conditional probability of $B_t = 1$ relative to the position of the forced points:

- $P^*(B_t = 1 | t < t_0) = (L-1)/(t_0 - 1)$ since $L-1$ of the $t_0 - 1$ random variables are high *before* t_0 and each such r.v. has an equal chance of being chosen.
- $P^*(B_t = 1 | t = t_0) = 1$ since t_0 is high. Note that this is only possible when $t \neq T$ since $t_0 < T$.
- $P^*(B_t = 1 | t > t_0) = 0$ since all B_t following t_0 are low.
- $P^*(B_t = 1 | t < t_1) = (t_1 - T + L)/(t_1 - 1)$ since $T - L - 1$ of the $t_1 - 1$ random variables are low (and thus $t_1 - T - L$ high) *before* t_1.
- $P^*(B_t = 1 | t = t_1) = 0$ since t_1 is low.
- $P^*(B_t = 1 | t > t_1) = 1$ since all B_t following t_1 are high.

[9] This can be verified by observing that eqs. (59) and (60) are negative binomial distributions: $P^*(t_0)$ with success probability $p' = p$, $r' = L$ successes and $k' = t_0 + L$ failures; and $P^*(t_1)$ with success probability $p' = 1 - p$, $r' = T - L$ successes, and $k' = t_1 + T - L$ failures. The rest proceeds similarly to our calculations for $P^*(B_t = 1 | 0 < t < L)$.

Therefore the probability that $B_t = 1$ under the forced suffix distribution is:

$$
\begin{aligned}
P^*(B_t = 1) = &\sum_{t_0=t+1}^{T-1} P^*(t_0)P^*(B_t = 1|t < t_0) + P^*(t_0 = t)P^*(B_t = 1|t = t_0) + \\
&\sum_{t_1=t+1}^{T-1} P^*(t_1)P^*(B_t = 1|t < t_1) + \sum_{t_1=0}^{t-1} P^*(t_1)P^*(B_t = 1|t > t_1) \\
= &\sum_{t_0=t+1}^{T-1} \frac{L-1}{t_0-1}\binom{t_0-1}{L-1}p^L(1-p)^{t_0-L} + \\
&I[t \neq T]\binom{t-1}{L-1}p^L(1-p)^{t-L} + \\
&\sum_{t_1=t+1}^{T-1} \frac{t_1-T+L}{t_1-1}\binom{t_1-1}{T-L-1}p^{t_1-T+L}(1-p)^{T-L} + \\
&\sum_{t_1=1}^{t-1} \binom{t_1-1}{T-L-1}p^{t_1-T+L}(1-p)^{T-L} + I[T = L] \\
= &I[L > 1] \sum_{t_0=t+1}^{T-1} \binom{t_0-2}{L-2}p^L(1-p)^{t_0-L} + \\
&I[t \neq T]\binom{t-1}{L-1}p^L(1-p)^{t-L} + \\
&\sum_{t_1=t+1}^{T-1} \binom{t_1-2}{T-L-1}p^{t_1-T+L}(1-p)^{T-L} + \\
&\sum_{t_1=T-L}^{t-1} \binom{t_1-1}{T-L-1}p^{t_1-T+L}(1-p)^{T-L} + I[T = L]
\end{aligned}
$$

The contributing summands differ according to t. Assume the usual case

that $L < T - L < T$.

$$\begin{aligned}
P^*(B_t = 1|0 < t < L) = &\sum_{t_0=L}^{T-1} \binom{t_0 - 2}{L-2} p^L (1-p)^{t_0 - L} + \\
&\sum_{t_1=T-L+1}^{T-1} \binom{t_1 - 2}{T-L-1} p^{t_1 - T + L} (1-p)^{T-L} \\
=&p \sum_{t_0=0}^{T-L-1} \binom{t_0 + L - 2}{t_0} p^{L-1} (1-p)^{t_0} + \\
&p \sum_{t_1=0}^{L-2} \binom{t_1 + T - L - 1}{t_1} p^{t_1} (1-p)^{T-L} \\
=&p \left(\bar{B}(p; L-1, T-L) + \bar{B}(1-p; T-L; L-1) \right) \\
=&p \left(1 - \bar{B}(1-p; T-L; L-1) + \bar{B}(1-p; T-L; L-1) \right) \\
=&p
\end{aligned}$$

$$P^*(B_t = 1|L \leq t \leq T - L, L = 1) = p(1-p)^{t-1}$$

$$\begin{aligned}
P^*(B_t = 1|1 < L \leq t \leq T - L) = &\sum_{t_0=t+1}^{T-1} \binom{t_0 - 2}{L-2} p^L (1-p)^{t_0 - L} + \\
&\binom{t-1}{L-1} p^L (1-p)^{t-L} + \\
&\sum_{t_1=T-L+1}^{T-1} \binom{t_1 - 2}{T-L-1} p^{t_1 - T + L} (1-p)^{T-L} \\
=&p + \binom{t-1}{L-1} p^L (1-p)^{t-L} - \\
&p\bar{B}(p; L-1, t-L+1)
\end{aligned}$$

where $\bar{B}(p; r, k)$ is the regularized incomplete beta function, which is also the CDF of the negative binomial distribution with success probability p, r successes and k failures.

Noting $P^*(B_L = 1|L < T - L) = p$ both when $L = 1$ and $L \neq 1$, we can extend the first interval to include L, that is

$$P^*(B_t = 1|0 < t \leq L) = p$$

which allows us to simplify

$$p^*(B_t = 1|L < t \leq T - L) = p \left(1 - \bar{B}(p; L, t-L) \right)$$

Where we have dropped the restriction that $L > 1$ since the right-hand side evaluates to the same value as the right-hand side for $P^*(B_t = 1|L \leq t \leq T - L, L = 1)$.

$$
\begin{aligned}
P^*(B_t = 1 | T - L < t < T) = &\sum_{t_0=t+1}^{T-1} \binom{t_0 - 2}{L - 2} p^L (1-p)^{t_0 - L} + \\
&\binom{t-1}{L-1} p^L (1-p)^{t-L} + \\
&\sum_{t_1=t+1}^{T-1} \binom{t_1 - 2}{T - L - 1} p^{t_1 - T + L} (1-p)^{T-L} + \\
&\sum_{t_1=T-L}^{t-1} \binom{t_1 - 1}{T - L - 1} p^{t_1 - T + L} (1-p)^{T-L} \\
=&p \left(1 - \bar{B}(p; L, T-L)\right) - \\
&\sum_{t_1=0}^{t-T+L-1} \binom{t_1 + T - L - 1}{t_1} p^{t_1+1} (1-p)^{T-L} + \\
&\sum_{t_1=0}^{t-T+L-1} \binom{t_1 + T - L - 1}{t_1} p^{t_1} (1-p)^{T-L} \\
=&p \left(1 - \bar{B}(p; L, t-L)\right) + \\
&(1-p) \bar{B}(1-p; T-L; t-T+L)
\end{aligned}
$$

$$
\begin{aligned}
P^*(B_T = 1) = &\sum_{t_1=T-L}^{T-1} \binom{t_1 - 1}{T - L - 1} p^{t_1 - T + L} (1-p)^{T-L} \\
=&\sum_{t_1=0}^{L-1} \binom{t_1 + T - L - 1}{t_1} p^{t_1} (1-p)^{T-L} \\
=&\bar{B}(1-p; T-L, L) \\
=&p \left(1 - \bar{B}(p; L, T-L)\right) + \\
&(1-p) \bar{B}(1-p; T-L; T-T+L)
\end{aligned}
$$

The right-hand side of $P^*(B_T = 1)$ equals the right-hand side of $P^*(B_t = 1 | T - L < t < T)$, meaning we can extend the latter interval to include T.

Collecting the various conditions on t together into a piecewise function for the probability of $B_t = 1$ under the forced-suffix distribution gives us eq. (3).

We can interpret eq. (3) as follows. The probability of the first L samples will never be tampered with when $L < T - L$ because there will always be at least one remaining high sample possible and at least L samples following. Thus the probability for $B_t = 1$ for $t \leq L$ will be merely the Bernoulli probability. Once $t > L$, some prefixes $b_{\leq t}$ (and all paths extending them) become invalid because they have too many high Bernoulli values. The $I[t > L] \ldots$ term sums the probabilities of all paths that would have contributed to the probability of $B_t = 1$, but have $L + 1$ or more high values (including B_t). When t approaches

the end of the sequence, $t > T - L$, the probability of $B_t = 1$ claims some additional probability mass of invalidated paths which would otherwise set $B_t = 0$ but are forced to 1 in order to make the total number of emissions L. The $I[t > T - L] \ldots$ term sums the probability of those paths, which have $T - L + 1$ *low* values (including the $B_t = 0$ term).

B TODO

Remove $P(L)$ from REINFORCE estimators

Motivations is not working as an intro

Hierarchical learning. Either through the estimators from section 2.2 or the exact form derived in section 2.4, the CB can be used to transduce a sequence of length T into one of length L. The advantage that the CB has over CTC is its ability to incorporate and simultaneously learn the dynamics of the L sequence while learning those of the T sequence. This does not prevent us from stacking a third sequence $M \ll L$ on top of the L sequence, or even deeper. For example, we could choose T to be acoustic frames, L to be sub-word units, and M to be full words. We could even recreate something akin to the FST composition used in hybrid speech recognition.