Note to reviewers: We managed to thighten Theorems 3.5 and 3.6 after the main text submission deadline. Hence, we also include the updated main text in the supplementary material.

Furthermore, we also updated the left middle panel in Figure 2 with a more precise estimate of the codelength for the methods. We suspect that split-on-sample GPRS's slight divergence from the predicted average is due to a numerical issue in our implementation, which we did not manage to fix before the supplementary material submission deadline.

Greedy Poisson Rejection Sampling

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

One-shot channel simulation is a fundamental data compression problem concerned with encoding a single sample from a target distribution Q using a coding distribution P using as few bits as possible on average. Algorithms that solve this problem find applications in neural data compression and differential privacy and can serve as a more efficient and natural alternative to quantization-based methods. Unfortunately, existing solutions are too slow or have limited applicability, preventing their widespread adoption. In this paper, we conclusively solve one-shot channel simulation for one-dimensional problems where the target-proposal density ratio is unimodal by describing an algorithm with optimal runtime. We achieve this by constructing a rejection sampling procedure equivalent to greedily searching over the points of a Poisson process. Hence, we call our algorithm greedy Poisson rejection sampling (GPRS) and analyze the correctness and time complexity of several of its variants. Finally, we empirically verify our theorems, demonstrating that GPRS significantly outperforms the current state-of-the-art method, A* coding.

1 Introduction

It is a common misconception that quantization is essential to lossy data compression; it is merely a way to discard information deterministically. In this paper, we consider the alternative, that is, to discard information stochastically using *one-shot channel simulation*. To illustrate the main idea, take lossy image compression as an example. Assume we have a generative model given by a joint distribution $P_{\mathbf{x},\mathbf{y}}$ over images $\mathbf{y}$ and latent variables $\mathbf{x}$, e.g. we might have trained a variational autoencoder (VAE; Kingma & Welling, 2014) on a dataset of images. To compress a new image $\mathbf{y}$, we encode a single sample from its posterior $\mathbf{x} \sim P_{\mathbf{x}|\mathbf{y}}$ as its stochastic lossy representation. The decoder can obtain a lossy reconstruction of $\mathbf{y}$ by decoding $\mathbf{x}$ and drawing a sample $\hat{\mathbf{y}} \sim P_{\mathbf{y}|\mathbf{x}}$ (though in practice, for a VAE we normally just take the mean predicted by the generative network).

Abstracting away from our example, in this paper we will be entirely focused on *channel simulation* for a pair of correlated random variables $\mathbf{x}, \mathbf{y} \sim P_{\mathbf{x},\mathbf{y}}$: given a source symbol $\mathbf{y} \sim P_{\mathbf{y}}$ we wish to encode **a single sample** $\mathbf{x} \sim P_{\mathbf{x}|\mathbf{y}}$. A simple way to achieve this is to encode $\mathbf{x}$ with entropy coding using the marginal $P_{\mathbf{x}}$, whose average coding cost is approximately the entropy $\mathbb{H}[\mathbf{x}]$. Surprisingly, however, we can do much better by using a *channel simulation protocol*, whose average coding cost is approximately the mutual information $I[\mathbf{x}; \mathbf{y}]$ (Li & El Gamal, 2018). This is remarkable, since not only $I[\mathbf{x}; \mathbf{y}] \leq \mathbb{H}[\mathbf{x}]$, but in many cases $I[\mathbf{x}; \mathbf{y}]$ might be finite even though $\mathbb{H}[\mathbf{x}]$ is infinite, such as when $\mathbf{x}$ is continuous. Sadly, most existing protocols place heavy restrictions on $P_{\mathbf{x},\mathbf{y}}$ or their runtime scales much worse than $\mathcal{O}(I[\mathbf{x}; \mathbf{y}])$, limiting their practical applicability (Agustsson & Theis, 2020).

In this paper, we propose a family of channel simulation protocols based on a new rejection sampling algorithm, which we can apply to simulate samples from a target distribution Q using a proposal distribution P over an arbitrary probability space. The inspiration for our construction comes from an exciting recent line of work which recasts random variate simulation as a search problem over a set of randomly placed points, specifically a Poisson process (Maddison, 2016). The most well-known

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

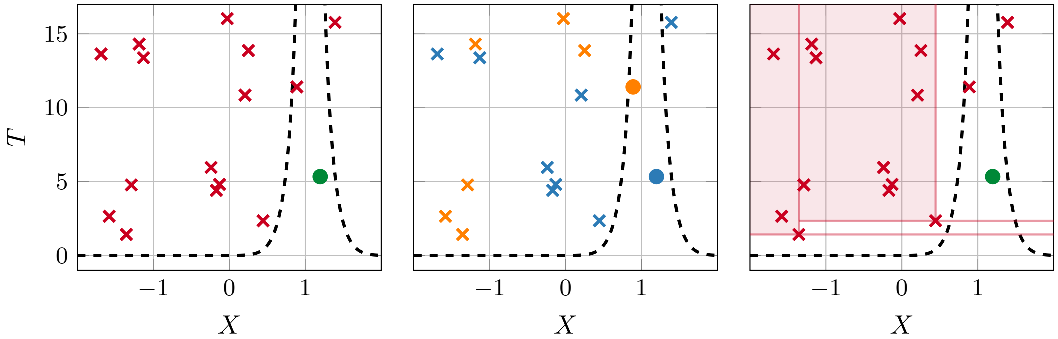

Figure 1: Illustration of three GPRS procedures for a Gaussian target $Q = \mathcal{N}(1, 0.25^2)$ and Gaussian proposal distribution $P = \mathcal{N}(0, 1)$, with the time axis truncated to the first 17 units. All three variants find the first arrival of a $(1, P)$-Poisson process Π under the graph of $\varphi = \sigma \circ r$ indicated by the **thick dashed black line** in each plot. Here, $r = dQ/dP$ is the target-proposal density ratio, and σ is given by Equation (3). **Left:** Algorithm 3 sequentially searching through the points of Π. The green circle (●) shows the first point of Π that falls under φ, and is accepted. All other points are rejected, as indicated by red crosses (✕). In practice, Algorithm 3 does not simulate points of Π that arrive after the accepted arrival. **Middle:** Parallelized GPRS (Algorithm 4) searching through two independent $(1/2, P)$-Poisson processes Π_1 and Π_2 in parallel. Blue points are arrivals in Π_1 and orange points are arrivals in Π_2. Crosses (✕) indicate rejected, and circles (●) indicate accepted points by each thread. In the end, the algorithm accepts the earliest arrival across all processes, which in this case is marked by the blue circle (●). **Right:** GPRS with binary search (Algorithm 5), when φ is unimodal. The shaded red areas are never searched or simulated by the algorithm since, given the first two rejections, we know points in those regions cannot fall under φ.

examples are the Gumbel-Max trick and A* sampling (Maddison et al., 2014). Our algorithm, which we call *greedy Poisson rejection sampling* (GPRS), differs significantly from all previous approaches in terms of what it is searching for, which we can succinctly summarise as: "GPRS searches for the first arrival of a Poisson process Π under the graph of an appropriately defined function φ". The first and simplest variant of GPRS is equivalent to an exhaustive search over all points of Π in time order. Next, we show that the linear search is embarrassingly parallelizable, leading to a parallelized variant of GPRS. Finally, when the underlying probability space has more structure, we develop branch-and-bound variants of GPRS that perform a binary search over the points of Π. See Figure 1 for an illustration of these three variants.

While GPRS is an interesting sampling algorithm on its own, we also show that each of its variants induces a new one-shot channel simulation protocol. That is, after we receive $\mathbf{y} \sim P_{\mathbf{y}}$, we can set $Q \leftarrow P_{\mathbf{x}|\mathbf{y}}$ and $P \leftarrow P_{\mathbf{x}}$ and use GPRS to encode a sample $\mathbf{x} \sim P_{\mathbf{x}|\mathbf{y}}$ at an average bitrate of a little more than the mutual information $I[\mathbf{x}; \mathbf{y}]$. In particular, on one-dimensional channel simulation problems where the density ratio $dP_{\mathbf{x}|\mathbf{y}}/dP_{\mathbf{x}}$ is unimodal for all $\mathbf{y}$, the binary search variant of GPRS leads to a protocol with an average runtime of $\mathcal{O}(I[\mathbf{x}; \mathbf{y}])$, which is optimal. This is a considerable improvement over A* coding (Flamich et al., 2022), the current state-of-the-art method.

In summary, our contributions are as follows:

- We construct a new rejection sampling algorithm called *greedy Poisson rejection sampling*, which we can construe as a greedy search over the points of a Poisson process (Algorithm 3). We propose a parallelized (Algorithm 4) and a branch-and-bound variant (Algorithms 5 and 6) of GPRS. We analyze the correctness and runtime of these algorithms.
- We show that each variant of GPRS induces a one-shot channel simulation protocol for correlated random variables $\mathbf{x}, \mathbf{y} \sim P_{\mathbf{x},\mathbf{y}}$, achieving the optimal average codelength of $I[\mathbf{x}; \mathbf{y}] + \log_2(I[\mathbf{x}; \mathbf{y}] + 1) + \mathcal{O}(1)$ bits.
- We prove that when $\mathbf{x}$ is a $\mathbb{R}$-valued random variable and the density ratio $dP_{\mathbf{x}|\mathbf{y}}/dP_{\mathbf{x}}$ is always unimodal, the channel simulation protocol based on the binary search variant of GPRS achieves $\mathcal{O}(I[\mathbf{x}; \mathbf{y}])$ runtime, which is optimal.
- We conduct toy experiments on one-dimensional problems and show that GPRS compares favourably against A* coding, the current state-of-the-art channel simulation protocol.

2 Background

The sampling algorithms we construct in this paper are search procedures on randomly placed points in space whose distribution is given by a Poisson process Π. Thus, in this section, we first review the necessary theoretical background for Poisson processes and how we can simulate them on a computer. Then, we formulate standard rejection sampling as a search procedure over the points of a Poisson process to serve as a prototype for our algorithm in the next section. Up to this point, our exposition loosely follows Sections 2, 3 and 5.1 of the excellent work of Maddison (2016), peppered with a few additional results that will be useful for analyzing our algorithm later. Finally, we describe the channel simulation problem, using rejection sampling as a rudimentary solution and describe its shortcomings. This motivates the development of greedy Poisson rejection sampling in Section 3.

2.1 Poisson Processes

A Poisson process Π is a countable collection of random points in some mathematical space Ω. In the main text, we will always assume that Π is defined over $\Omega = \mathbb{R}^+ \times \mathbb{R}^d$ and that all objects involved are measure-theoretically well-behaved for simplicity. For this choice of Ω, the positive reals represent *time*, and the unit interval represents *space*. However, most results generalize to settings when the spatial domain $\mathbb{R}^d$ is replaced with some more general space, and we give a general measure-theoretic construction in Appendix A, where the spatial domain is an arbitrary Polish space.

Basic properties of Π**:** For a set $A \subseteq \Omega$, let $\mathbf{N}(A) \stackrel{def}{=} |\Pi \cap A|$ denote the number of points of Π falling in the set A, where $|\cdot|$ denotes the cardinality of a set. Then, Π is characterized by the following two fundamental properties (Kingman, 1992). First, for two disjoint sets $A, B \subseteq \Omega, A \cap B = \emptyset$, the number of points of Π that fall in either set are independent random variables: $\mathbf{N}(A) \perp \mathbf{N}(B)$. Second, $\mathbf{N}(A)$ is Poisson distributed with *mean measure* $\mu(A) \stackrel{def}{=} \mathbb{E}[\mathbf{N}(A)]$.

Time-ordering the points of Π**:** Since we assume that $\Omega = \mathbb{R}^+ \times \mathbb{R}^d$ has a product space structure, we may write the points of Π as a pair of time-space coordinates: $\Pi = \{(T_n, X_n)\}_{n=1}^{\infty}$. Furthermore, we can order the points in Π with respect to their time coordinates and *index* them accordingly, i.e. for $i < j$ we have $T_i < T_j$. Hence, we refer to (T_n, X_n) as the nth *arrival* of the process.

As a slight abuse of notation, we define $\mathbf{N}(t) \stackrel{def}{=} \mathbf{N}([0, t) \times \mathbb{R}^d)$ and $\mu(t) \stackrel{def}{=} \mathbb{E}[\mathbf{N}(t)]$, i.e. these quantities measure the number and average number of points of Π that arrive before time t, respectively. In this paper, we assume $\mu(t)$ has derivative $\mu'(t)$, and assume for each $t \geq 0$ there is a conditional probability distribution $P_{X|T=t}$ with density $p(x \mid t)$, such that we can write the mean measure as $\mu(A) = \int_A p(x \mid t)\mu'(t)\,dx\,dt$.

Simulating Π**:** A simple method to simulate Π on a computer is to realize it in time order, i.e. at step n, simulate T_n and then use it to simulate $X_n \sim P_{X|T=T_n}$. We can find out the distribution of Π's first arrival by noting that by its definition, no point of Π can come before it, hence $\mathbb{P}[T_1 \geq t] = \mathbb{P}[\mathbf{N}(t) = 0] = \exp(-\mu(t))$, where the second equality follows from the fact that $\mathbf{N}(t)$ is Poisson distributed. A particularly important case is when Π is *time-homogeneous*, i.e. $\mu(t) = \lambda t$ for some $\lambda > 0$, in which case $T_1 \sim \mathrm{Exp}(\lambda)$ is an exponential random variable with *rate* λ. In fact, all of Π's inter-arrival times $\Delta_n = T_n - T_{n-1}$ for $n \geq 1$ share this simple distribution, where we set $T_0 = 0$. To see this, note that

$$\mathbb{P}[\Delta_n \geq t \mid T_{n-1}] = \mathbb{P}\left[\mathbf{N}\left([T_{n-1}, T_{n-1} + t) \times \mathbb{R}^d\right) = 0 \mid T_{n-1}\right] = \exp(-\lambda t),$$

i.e. all $\Delta_n \mid T_{n-1} \sim \mathrm{Exp}(\lambda)$. Therefore, we can use the above procedure to simulate time-homogeneous Poisson processes, described in Algorithm 1. We will refer to a time-homogeneous Poisson process with time rate λ and spatial distribution $P_{X|T}$ as a $(\lambda, P_{X|T})$-Poisson process.

Algorithm 1: Generating a $(\lambda, P_{X|T})$-Poisson process.

```
Input :Time rate $\lambda$,
       Spatial distribution $P_{X|T}$
$T_0 \leftarrow 0$
for $n = 1, 2, \ldots$ do
    $\Delta_n \sim \mathrm{Exp}(\lambda)$
    $T_n \leftarrow T_{n-1} + \Delta_n$
    $X_n \sim P_{X|T=T_n}$
    yield $(T_n, X_n)$
end
```

Rejection sampling using Π**:** Rejection sampling is a technique to simulate samples from a *target distribution* Q using a *proposal distribution* P, assuming we can find an upper bound $M > 0$ for their density ratio $r = dQ/dP$ (technically, the Radon-Nikodym derivative). We can formulate this

Algorithm 2: Standard rejection sampling.

```
Input  : Proposal distribution $P$,
         Density ratio $r = dQ/dP$,
         Upper bound $M$ for $r$.
Output : Sample $X \sim Q$ and its index $N$.
// Generator for a $(1, P)$-Poisson
   process using Algorithm 1.
$\Pi \leftarrow$ SimulatePP$(1, P)$
for $n = 1, 2, \ldots$ do
    $(T_n, X_n) \leftarrow$ next$(\Pi)$
    $U_n \sim \mathrm{Unif}(0, 1)$
    if $U_n < r(X_n)/M$ then
        return $X_n, n$
    end
end
```

Algorithm 3: Greedy Poisson rejection sampling.

```
Input  : Proposal distribution $P$,
         Density ratio $r = dQ/dP$,
         Stretch function $\sigma$.
Output : Sample $X \sim Q$ and its index $N$.
// Generator for a $(1, P)$-Poisson
   process using Algorithm 1.
$\Pi \leftarrow$ SimulatePP$(1, P)$
for $n = 1, 2, \ldots$ do
    $(T_n, X_n) \leftarrow$ next$(\Pi)$
    if $T_n < \sigma(r(X_n))$ then
        return $X_n, n$
    end
end
```

procedure using a Poisson process: we simulate the arrivals (T_n, X_n) of a $(1, P)$-Poisson process Π, but we only keep them with probability $r(X_n)/M$, otherwise, we delete them. This algorithm is described in Algorithm 2; its correctness is guaranteed by the *thinning theorem* (Maddison, 2016).

Rejection sampling is suboptimal: Using Poisson processes to formulate rejection sampling highlights a subtle but crucial inefficiency: it does not make use of Π's temporal structure and only uses the spatial coordinates. GPRS fixes this by using a rejection criterion that does depend on the time variable. As we show, this significantly speeds up sampling for certain classes of distributions.

2.2 Channel Simulation

The main motivation for our work is to develop a *one-shot channel simulation protocol* using the sampling algorithm we derive in Section 3. Channel simulation is of significant theoretical and practical interest. Recent works used it to compress neural network weights, achieving state-of-the-art performance (Havasi et al., 2018); to perform image compression using variational autoencoders (Flamich et al., 2020) and diffusion models with perfect realism (Theis et al., 2022); and to perform differentially private federated learning by compressing noisy gradients (Shah et al., 2022).

One-shot channel simulation is a communication problem between two parties, Alice and Bob, sharing a joint distribution $P_{\mathbf{x},\mathbf{y}}$ over two correlated random variables $\mathbf{x}$ and $\mathbf{y}$, where we assume that Alice and Bob can simulate samples from the marginal $P_{\mathbf{x}}$. In a single round of communication, Alice receives a sample $\mathbf{y} \sim P_{\mathbf{y}}$ from the marginal distribution over $\mathbf{y}$. Then, she needs to send the minimum number of bits to Bob such that he can **simulate a single sample** from the conditional distribution $\mathbf{x} \sim P_{\mathbf{x}|\mathbf{y}}$. Note that Bob **does not want to learn** $P_{\mathbf{x}|\mathbf{y}}$; he just wants to simulate a single sample from it. Surprisingly, when Alice and Bob have access to *shared randomness*, e.g. by sharing the seed of their random number generator before communication, they can solve channel simulation very efficiently. Mathematically, in this paper we will always model this shared randomness by some time-homogeneous Poisson process (or processes) Π, since given a shared random seed, Alice and Bob can always simulate the same process e.g. using Algorithm 1. Then, the average coding cost of $\mathbf{x}$ given Π is its conditional entropy $\mathbb{H}[\mathbf{x} \mid \Pi]$ and, surprisingly, it is always upper bounded by $\mathbb{H}[\mathbf{x} \mid \Pi] \leq I[\mathbf{x}; \mathbf{y}] + \log_2 I[\mathbf{x}; \mathbf{y}] + \mathcal{O}(1)$, where $I[\mathbf{x}; \mathbf{y}]$ is the mutual information between $\mathbf{x}$ and $\mathbf{y}$ (Li & El Gamal, 2018). This is an especially curious result, given that in many cases $\mathbb{H}[\mathbf{x}]$ is infinite while $I[\mathbf{x}; \mathbf{y}]$ is finite, e.g. when $\mathbf{x}$ is a continuous variable. In essence, this result means that given the additional structure $P_{\mathbf{x},\mathbf{y}}$, channel simulation protocols can "offload" an infinite amount of information into the shared randomness, and only communicate the finitely many "necessary" bits.

An example channel simulation protocol with rejection sampling: Given Π and $\mathbf{y} \sim P_{\mathbf{y}}$, Alice sets $Q \leftarrow P_{\mathbf{x}|\mathbf{y}}$ as the target and $P \leftarrow P_{\mathbf{x}}$ as the proposal distribution with density ratio $r = dQ/dP$, and run the rejection sampler in Algorithm 2 to find the first point of Π that was not deleted. She counts the number of samples N she had to simulate before acceptance and sends this number to Bob. He decodes a sample $\mathbf{x} \sim P_{\mathbf{x}|\mathbf{y}}$ by selecting the spatial coordinate of the Nth arrival of Π.

Unfortunately, this simple protocol is suboptimal. To see this, let $D_\infty[Q \| P] \overset{def}{=} \sup_{\mathbf{x} \in \Omega}\{\log_2 r(\mathbf{x})\}$ denote Rényi ∞-divergence from Q to P, and recall two standard facts: (1) the best possible upper bound Alice can use for rejection sampling is $M_{opt} = \exp_2\left(D_\infty[Q\|P]\right)$, where $\exp_2(x) = 2^x$, and (2), the number of samples N drawn until acceptance is a geometric random variable with mean M_{opt} (Maddison, 2016). We now state the two issues with rejection sampling that GPRS solves.

Problem 1: Codelength: However, by using the formula for the entropy of a geometric random variable and assuming Alice uses the best possible bound M_{opt} in the protocol, we find that

$$\mathbb{H}[\mathbf{x} \mid \Pi] = \mathbb{E}_{\mathbf{y}\sim P_{\mathbf{y}}}[\mathbb{H}[N \mid \mathbf{y}]] \geq \mathbb{E}_{\mathbf{y}\sim P_{\mathbf{y}}}[D_\infty[P_{\mathbf{x}|\mathbf{y}}\|P_{\mathbf{x}}]] \overset{(a)}{\geq} I[\mathbf{x};\mathbf{y}],$$

see Appendix I for the derivation. Unfortunately, inequality (a) can be *arbitrarily loose*, hence the average codelength scales with the expected ∞-divergence instead of $I[\mathbf{x};\mathbf{y}]$, as would be optimal.

Problem 2: Slow runtime: We are interested in classifying the time complexity of our protocol. As we saw, for a target Q and proposal P, Algorithm 2 draws $M_{opt} = \exp_2\left(D_\infty[Q\|P]\right)$ samples on average. Unfortunately, under the computational hardness assumption RP $\neq$ NP, Agustsson & Theis (2020) showed that without any further assumptions, there is no sampler that scales polynomially in $D_{\mathrm{KL}}[Q\|P]$. However, with further assumptions, we can do much better, as we show in Section 3.1.

3 Greedy Poisson Rejection Sampling

We now describe GPRS; its pseudo-code is shown in Algorithm 3. This section assumes that Q and P are the target and proposal distributions, respectively, and $r = dQ/dP$ is their density ratio. Let Π be a $(1, P)$-Poisson process. Our proposed rejection criterion is now embarrassingly simple: for an appropriately defined invertible function $\sigma : \mathbb{R}^+ \to \mathbb{R}^+$, accept the first arrival of Π that falls under the graph of the composite function $\varphi = \sigma \circ r$, as illustrated in the left plot in Figure 1. We refer to σ as the *stretch function* for r, as its purpose is to stretch the density ratio along the time-axis of Π.

Deriving the stretch function (sketch): Let $\varphi = \sigma \circ r$, where for now σ is an arbitrary invertible function on $\mathbb{R}^+$, let $U = \{(t, x) \in \Omega \mid t \leq \varphi(x)\}$ be the set of points under the graph of φ and let $\tilde{\Pi} = \Pi \cap U$. By the restriction theorem (Kingman, 1992), $\tilde{\Pi}$ is also a Poisson process with mean measure $\tilde{\mu}(A) = \mu(A \cap U)$. Let $(\tilde{T}, \tilde{X})$ be the first arrival of $\tilde{\Pi}$, i.e. the first arrival of Π under φ and let Q_σ be the distribution of $\tilde{X}$. Then, the density ratio dQ_σ/dP obeys the identity (see Appendix A):

$$\frac{dQ_\sigma}{dP}(x) = \int_0^{\varphi(x)} \mathbb{P}[\tilde{T} \geq t]\, dt. \tag{1}$$

Now we pick σ such that $Q_\sigma = Q$, for which we need to ensure that $dQ_\sigma/dP = r$. Substituting $\tau = \varphi(x)$ into Equation (1), and differentiating, we get $\left(\sigma^{-1}\right)'(\tau) = \mathbb{P}[\tilde{T} \geq t]$. Since $(\tilde{T}, \tilde{X})$ falls under the graph of φ by definition, we find that $\mathbb{P}[\tilde{T} \geq t] = \mathbb{P}[\tilde{T} \geq t, r(\tilde{X}) \geq \sigma^{-1}(t)]$. By expanding the definition of the right-hand side, in Appendix A we obtain a time-invariant ODE for σ^{-1}:

$$\left(\sigma^{-1}\right)'(\tau) = w_Q\left(\sigma^{-1}(t)\right) - \sigma^{-1}(t) \cdot w_P\left(\sigma^{-1}(t)\right), \quad \text{with} \quad \sigma^{-1}(0) = 0 \tag{2}$$

where we define $w_P(h) \overset{def}{=} \mathbb{P}_{Z\sim P}[r(Z) \geq h]$ and define w_Q analogously. In Appendix G, we provide the analytic form of w_P and w_Q for discrete, uniform, triangular, Gaussian, and Laplace distributions. Finally, we use the inverse function theorem and integrate both sides to obtain

$$\sigma(h) = \int_0^h \frac{1}{w_Q(\eta) - \eta \cdot w_P(\eta)}\, d\eta. \tag{3}$$

Remember that picking σ according to Equation (3) ensures that GPRS is **correct by construction**. To complete the picture, in Appendix A we prove that $(\tilde{T}, \tilde{X})$ always exists and Algorithm 3 terminates with probability 1. Unfortunately, computing the integral in Equation (3) analytically is usually not possible. Moreover, we can show that solving it numerically is unstable as σ is unbounded. Instead, we numerically solve for σ^{-1} using Equation (2) in practice, which fortunately turns out to be stable.

We now turn our attention to analyzing the runtime of Algorithm 3 and find the following surprising result: the expected runtime of GPRS matches that of standard rejection sampling.

Theorem 3.1 (Expected Runtime). *Let Q and P be the target and proposal distributions for Algorithm 3, respectively, and $r = dQ/dP$ their density ratio. Let N denote the number of samples simulated by the algorithm before it terminates. Then,*

$$\mathbb{E}[N] = \exp_2\left(D_\infty[Q\|P]\right) \quad \textit{and} \quad \mathbb{V}[N] \geq \exp_2\left(D_\infty[Q\|P]\right). \tag{4}$$

Algorithm 4: Parallel GPRS with J available threads.

```
Input  : Proposal distribution $P$,
         Density ratio $r = dQ/dP$,
         Stretch function $\sigma$,
         Number of parallel threads $J$.
Output : Sample $X \sim Q$ and its code $(j^*, N_{j^*})$.
$T^*, X^*, j^*, N_{j^*} \leftarrow \infty, \texttt{nil}, \texttt{nil}, \texttt{nil}$
in parallel for $j = 1, \dots, J$ do
    $\Pi_j \leftarrow \texttt{SimulatePP}(1/J, P)$
    for $n_j = 1, 2, \dots$ do
        $\left(T^{(j)}_{n_j}, X^{(j)}_{n_j}\right) \leftarrow \texttt{next}(\Pi_j)$
        if $T^* < T^{(j)}_{n_j}$ then
            terminate thread $j$.
        end
        if $T^{(j)}_{n_j} < \sigma\left(r\left(X^{(j)}_{n_j}\right)\right)$ then
            $T^*, X^*, j^*, N_{j^*} \leftarrow T^{(j)}_{n_j}, X^{(j)}_{n_j}, j, n_j$
            terminate thread $j$.
        end
    end
end
return $X^*, (j^*, N_{j^*})$
```

Algorithm 5: Branch-and-bound GPRS on $\mathbb{R}$ with unimodal r

```
Input  : Proposal distribution $P$,
         Density ratio $r = dQ/dP$,
         Stretch function $\sigma$,
         Location $x^*$ of the mode of $r$.
Output : Sample $X \sim Q$ and its heap index $H$.
$T_0, H, B \leftarrow (0, 1, \mathbb{R})$
for $d = 1, 2, \dots$ do
    $X_d \sim P|_B$
    $\Delta_d \sim \mathrm{Exp}\,(P(B))$
    $T_d \leftarrow T_{d-1} + \Delta_d$
    if $T_d < \sigma(r(X_d))$ then
        return $X_d, H$
    end
    if $X_d \geq x^*$ then
        $B \leftarrow B \cap (-\infty, X_d)$
        $H \leftarrow 2H$
    else
        $B \leftarrow B \cap (X_d, \infty)$
        $H \leftarrow 2H + 1$
    end
end
```

GPRS induces a channel simulation protocol similar to the standard rejection sampling-based one in Section 2.2: The encoder simulates the arrivals of Π using shared randomness and encodes the index of their accepted sample. The next theorem shows that this protocol is optimally efficient.

Theorem 3.2 (Expected Codelength)**.** *Let $P_{\mathbf{x},\mathbf{y}}$ be a joint distribution over correlated random variables $\mathbf{x}$ and $\mathbf{y}$, and let Π be the $(1, P)$-Poisson process used by Algorithm 3. Then, the algorithm induces a channel simulation protocol, such that*

$$\mathbb{H}[\mathbf{x} \mid \Pi] \leq I[\mathbf{x}; \mathbf{y}] + \log_2\left(I[\mathbf{x}; \mathbf{y}] + 1\right) + 6. \tag{5}$$

See Appendix B.2 and Appendix B.3 for proofs. This result is analogous to the Poisson functional representation (Li & El Gamal, 2018), and thus it can be used to provide an alternative proof of the strong functional representation lemma (Theorem 1; Li & El Gamal, 2018).

GPRS as greedy search: We contrast GPRS with the well-known A* sampling algorithm (Maddison et al., 2014), which also searches through the points of Π but uses a different criterion. The defining feature of GPRS is that its acceptance criterion at each step is *local*, since if at step n the arrival (T_n, X_n) in Π falls under φ, we will immediately accept it. Thus it is *greedy* search procedure. On the other hand, the A* acceptance criterion is *global*, as the acceptance of a particular arrival (T_n, X_n) depends on all other points of Π in the general case. Surprisingly, this difference between the search criteria does not make a difference in the average runtimes and codelengths in the general case. However, as shown in the next section, GPRS can be much faster in special cases.

3.1 Speeding up the greedy search

This section discusses two ways to improve the runtime of GPRS. First, we show how we can utilize available parallel computing power to speed up Algorithm 3. Second, we propose an advanced search strategy when the spatial domain Ω has more structure and show that we can obtain a **super-exponential improvement** in the runtime from $\exp_2\left(D_\infty[Q\|P]\right)$ to $\mathcal{O}(D_{\mathrm{KL}}[Q\|P])$ in certain cases.

Parallel GPRS: The basis for parallelizing GPRS is the *superposition theorem* (Kingman, 1992): Let $\Pi_1, \dots \Pi_J$ all be $(1/J, P)$-Poisson processes; then, $\Pi = \bigcup_{j=1}^J \Pi_j$ is a $(1, P)$-Poisson process. This result makes parallelizing GPRS very simple, as shown in Algorithm 4, assuming we have J parallel threads: First, we independently look for the first arrivals of $\Pi_1, \dots, \Pi_J$ under the graph of φ, yielding $\left(T^{(1)}_{N_1}, X^{(1)}_{N_1}\right), \dots, \left(T^{(J)}_{N_J}, X^{(J)}_{N_J}\right)$, respectively, where the N_j corresponds to the index of Π_j's first arrival under φ. Then, we select the candidate with the earliest arrival time, i.e. $j^* = \arg\min_{j \in \{1,\ldots,J\}} T^{(j)}_{N_j}$. Now, by the superposition theorem, $\left(T^{(j^*)}_{N_{j^*}}, X^{(j^*)}_{N_{j^*}}\right)$ is the first arrival of Π under φ, and hence $X^{(j^*)}_{N_{j^*}} \sim Q$. Finally, Alice encodes the sample via the tuple (j^*, N_{j^*}), i.e. which of the J processes the first arrival occurred in, and the index of the arrival in Π_{j^*}. See the middle plot in Figure 1 for an example case with $J = 2$.

Our next result shows that parallelizing GPRS results in a linear reduction in both the expectation and variance of its runtime and a more favourable codelength guarantee for channel simulation.

Theorem 3.3 (Expected runtime of parallelized GPRS)**.** *Let Q, P and r be defined as above, and let ν_j denote the random variable corresponding to the number of samples simulated by thread j in Algorithm 4 using J threads. Then, for all j,*

$$\mathbb{E}[\nu_j] = \left(\exp_2\left(D_\infty[Q\|P]\right) - 1\right) \big/ J + 1 \quad \text{and} \quad \mathbb{V}[\nu_j] \geq \left(\exp_2\left(D_\infty[Q\|P]\right) - 1\right) \big/ J + 1. \tag{6}$$

Theorem 3.4 (Expected codelength of parallelized GPRS)**.** *Let $P_{\mathbf{x},\mathbf{y}}$ be a joint distribution over correlated random variables $\mathbf{x}$ and $\mathbf{y}$, and let $\Pi_1, \ldots, \Pi_J$ be $(1/J, P_{\mathbf{x}})$-Poisson processes. Then, assuming $\log_2 J \leq I[\mathbf{x};\mathbf{y}]$, parallelized GPRS induces a channel simulation protocol such that*

$$\mathbb{H}[\mathbf{x} \mid \Pi_1, \ldots, \Pi_J] \leq I[\mathbf{x};\mathbf{y}] + \log_2\left(I[\mathbf{x};\mathbf{y}] - \log_2 J + 1\right) + 8. \tag{7}$$

See Appendix C for the proofs. Note that we can use the same parallelisation argument with the appropriate modifications to speed up A* sampling / coding too.

Branch-and-bound GPRS on $\mathbb{R}$: We briefly restrict our attention to problems on $\mathbb{R}$ when the density ratio r is unimodal, as we can exploit this additional structure to more efficiently search for Π's first arrival under φ. Consider the example in the right plot in Figure 1: we simulate the first arrival (T_1, X_1), and reject it, since $T_1 > \varphi(X_1)$. Since φ is unimodal by assumption, for $x < X_1$ we must have $\varphi(x) < \varphi(X_1)$, while the arrival time of any of the later arrivals will be larger than T_1. Therefore, none of the arrivals to the left of X_1 will fall under φ either! Hence, it is enough to simulate Π to the right of X_1. Repeating this argument for later arrivals, we obtain an efficient binary search procedure to find Π's first arrival under φ, described in Algorithm 5: We simulate the next arrival (T_B, X_B) of Π within some bounds B, starting with the first arrival in $B = \Omega$. If $T_B \leq \varphi(X_B)$ we accept; otherwise, we truncate the bound to $B \leftarrow B \cap (-\infty, X_B)$, or $B \leftarrow B \cap (X_B, \infty)$ based on where X_B falls relative to φ's mode. We repeat these two steps until we find the first arrival.

Since Algorithm 5 does not simulate every point of Π, we cannot use the index N of the accepted arrival to obtain a channel simulation protocol as before. Instead, we encode the *search path*, i.e. whether we chose the left or the right side of our current sample at each step. Similarly to A* coding, we encode the path using its *heap index* (Flamich et al., 2022): the root has index $H_{root} = 1$, and for a node with index H, its left child is assigned index $2H$ and its right child $2H + 1$. As the following theorems show, this version of GPRS is, in fact, optimally efficient.

Theorem 3.5 (Expected Runtime of GPRS with binary search)**.** *Let Q, P and r be defined as above, and let D denote the number of samples simulated by Algorithm 5. Then,*

$$\mathbb{E}[D] = D_{\mathrm{KL}}[Q\|P] + 3. \tag{8}$$

Theorem 3.6 (Expected Codelength of GPRS with binary search)**.** *Let $P_{\mathbf{x},\mathbf{y}}$ be a joint distribution over correlated random variables $\mathbf{x}$ and $\mathbf{y}$, and let Π be a $(1, P_{\mathbf{x}})$-Poisson process. Then, GPRS with binary search induces a channel simulation protocol such that*

$$\mathbb{H}[\mathbf{x} \mid \Pi] \leq I[\mathbf{x};\mathbf{y}] + \log_2\left(I[\mathbf{x};\mathbf{y}] + 1\right) + 8 \tag{9}$$

See Appendix E for details and proofs of the theorems.

Branch-and-bound GPRS with splitting functions: With some additional machinery, Algorithm 5 can be extended to much more general settings, such as $\mathbb{R}^D$ and cases where r is not unimodal, by introducing the notion of a *splitting function*. For a region of space, $B \subseteq \Omega$, a splitting function `split` simply returns a binary partition of the set, i.e. $\{L, R\} = \texttt{split}(B)$, such that $L \cap R = \emptyset$ and $L \cup R = B$. In this case, we can perform a similar tree search to Algorithm 5, captured in Algorithm 6. Starting with the whole space $B = \Omega$, we simulate the next arrival (T, X) of Π in B at each step. If we reject it, we partition B into two parts, L and R, using `split`. Then, with probability $\mathbb{P}[\tilde{X} \in R \mid \tilde{X} \in B, \tilde{T} \geq T]$, we continue searching through the arrivals of Π in R only, and in L only

```
Algorithm 6: Branch-and-bound GPRS with splitting function.
Input  : Proposal distribution $P$,
         Density ratio $r = dQ/dP$,
         Stretch function $\sigma$,
         Splitting function split.
Output : Sample $X \sim Q$ and its
         heap index $H$
$T_0, H, B \leftarrow (0, 1, \mathbb{R})$
for $d = 1, 2, \dots$ do
    $X_d \sim P|_B$
    $\Delta_d \sim \mathrm{Exp}\,(P(B))$
    $T_d \leftarrow T_{d-1} + \Delta_d$
    if $T_d < \sigma(r(X_d))$ then
        return $X_d, H$
    else
        $B_0, B_1 \leftarrow$ split$(B)$
        $\rho \leftarrow$
          $\mathbb{P}[\tilde{X} \in B_1 \mid \tilde{X} \in B, \tilde{T} \geq T_d]$
        $\beta \leftarrow \mathrm{Bernoulli}(\rho)$
        $H \leftarrow 2H + \beta$
        $B \leftarrow B_\beta$
    end
end
```

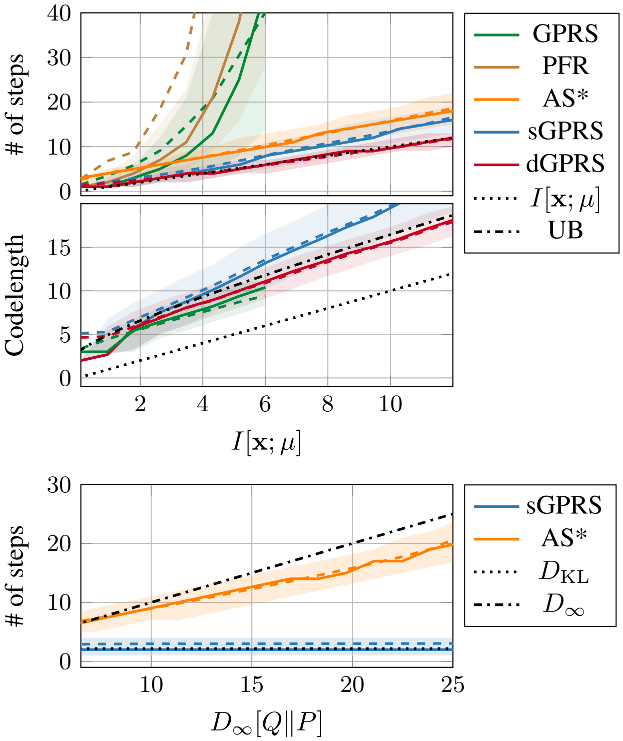

Figure 2: **Left:** Binary search GPRS with arbitrary splitting function. **Right:** Performance comparison of different channel simulation protocols. In each plot, *dashed lines* indicate the mean, *solid lines* the median and the *shaded areas* the 25 - 75 percentile region of the relevant performance metric. We computed the statistics over 1000 runs for each setting. **Top right:** Runtime comparison on a 1D Gaussian channel simulation problem $P_{\mathbf{x},\mu}$, plotted against increasing mutual information $I[\mathbf{x}; \mu]$. Alice receives $\mu \sim \mathcal{N}(0, \sigma^2)$ and encodes a sample $\mathbf{x} \mid \mu \sim \mathcal{N}(\mu, 1)$ to Bob. The abbreviations in the legend are: *GPRS* – Algorithm 3; *PFR* – Poisson functional representation / Global-bound A* coding (Li & El Gamal, 2018; Flamich et al., 2022), *AS** – Split-on-sample A* coding (Flamich et al., 2022); *sGPRS* – split-on-sample GPRS (Algorithm 5); *dGPRS* – GPRS with dyadic `split` (Algorithm 6). **Middle right:** Average codelength comparison of our proposed algorithms on the same channel simulation problem as above. *UB* in the legend corresponds to an *upper bound* of $I[\mathbf{x}; \mu] + \log_2(I[\mathbf{x}; \mu] + 1) + 2$ bits. We estimate the algorithms' expected codelengths by encoding the indices returned by GPRS using a Zeta distribution $\zeta(n \mid \lambda) \propto n^{-\lambda}$ with $\lambda = 1 + 1/I[\mathbf{x}; \mathbf{y}]$ in each case, which is the optimal maximum entropy distribution for this problem setting (Li & El Gamal, 2018). **Bottom right:** One-shot runtime comparison of sGPRS with AS* coding. Alice encodes samples from a target $Q = \mathcal{N}(m, s^2)$ using $P = \mathcal{N}(0, 1)$ as the proposal. We computed m and s^2 such that $D_{\mathrm{KL}}[Q\|P] = 2$ bits for each problem, but $D_\infty[Q\|P]$ increases. GPRS' runtime stays fixed as it scales with $D_{\mathrm{KL}}[Q\|P]$, while the runtime of A* keeps increasing.

otherwise. We show the correctness of this procedure in Appendix F, where we also derive how the quantities $\mathbb{P}[\tilde{X} \in R \mid \tilde{X} \in B, \tilde{T} \geq T]$ and σ can be computed in practice. This splitting procedure is analogous to the general version of A* sampling/coding (Maddison et al., 2014; Flamich et al., 2022), which is parameterized by the same splitting functions. Note that Algorithm 5 is a special case of Algorithm 6, where the underlying space is $\Omega = \mathbb{R}$, r is unimodal, and at each step for an interval bound $B = (a, b)$ and sample $X \in (a, b)$ we split B into $\{(a, X), (X, b)\}$.

4 Experiments

We compare the average and one-shot case efficiency of our proposed variants of GPRS and a couple of other channel simulation protocols in Figure 2. See the figure's caption and Appendix H for details of our experimental setup. The top two plots in Figure 2 demonstrate that our methods' expected runtimes and codelengths align very well with our theorems' predictions and compare favourably to other methods. Furthermore, we find that the mean performance is a *robust statistic* for the binary search-based variants of GPRS in that it lines up closely with the median, and the interquartile range is quite narrow. On the other hand, the bottom plot in Figure 2 demonstrates the most salient property of the binary search variant of GPRS: unlike all previous methods, its runtime scales with $D_{\mathrm{KL}}[Q\|P]$ and not $D_\infty[Q\|P]$. Thus, we can apply it to a larger family of channel simulation problems both in theory and practice where $D_{\mathrm{KL}}[Q\|P]$ is finite, but $D_\infty[Q\|P]$ is very large or even infinite in expectation, and other methods would not terminate.

5 Related Work

Poisson processes for channel simulation: Poisson processes were introduced to the channel simulation literature by Li & El Gamal (2018) via their construction of the *Poisson functional representation* (PFR) in their proof of the Strong Functional Representation Lemma. Flamich et al. (2022) observed that the PFR construction is equivalent to a certain variant of A* sampling (Maddison et al., 2014; Maddison, 2016). Thus, they proposed an optimized version of the PFR called A* coding, which achieves $\mathcal{O}(D_\infty[Q\|P])$ runtime for one-dimensional unimodal distributions. GPRS was mainly inspired by A* coding, and they are *dual* constructions to each other in the following sense: *Depth-limited A* coding* can be thought of as an *importance sampler*, i.e. a Monte Carlo algorithm that returns an approximate sample in fixed time. On the other hand, GPRS is a *rejection sampler*, i.e. a Las Vegas algorithm that returns an exact sample in random time.

Channel simulation with dithered quantization: *Dithered quantization* (DQ; Ziv, 1985) is an alternative to rejection and importance sampling-based approaches to channel simulation. DQ exploits that for any $c \in \mathbb{R}$ and $U, U' \sim \mathrm{Unif}(-1/2, 1/2)$, the quantities $\lfloor c - U \rceil + U$ and $c + U'$ are equal in distribution. While DQ has been around for decades as a tool to model and analyze quantization error, Agustsson & Theis (2020) reinterpreted it as a channel simulation protocol and used it to develop a VAE-based neural image compression algorithm. Unfortunately, basic DQ only allows uniform target distributions, limiting its applicability. As a partial remedy, Theis & Yosri (2022) showed DQ could be combined with other channel simulation protocols to speed them up and thus called their approach hybrid coding (HQ). Originally, HQ required that the target distribution be compactly supported, which was lifted by Flamich & Theis (2023), who developed an adaptive rejection sampler using HQ. In a different vein, Hegazy & Li (2022) generalize and analyze a method proposed in the appendix of Agustsson & Theis (2020) and show that DQ can be used to realize channel simulation protocols for one-dimensional symmetric, unimodal distributions.

Greedy Rejection Coding: Concurrently, Anonymous (2023) generalize Harsha et al. (2007)'s rejection sampling algorithm to arbitrary probability spaces, which they call greedy rejection coding (GRC). Furthermore, they also introduce a space-partitioning procedure to speed up the convergence of their sampler and prove that a variant of their sampler also achieves optimal runtime for one-dimensional problems where dQ/dP is unimodal. However, the construction of their method differs significantly from ours. GRC is a direct generalization of Harsha et al. (2007)'s algorithm and, thus, a more "conventional" rejection sampler, while we base our construction on Poisson processes. Thus, our proof techniques are also significantly different. It is an interesting research question whether GRC could be formulated using Poisson processes, akin to the formulation of standard rejection sampling in Algorithm 2, as this could be used to connect the two algorithms and improve both.

6 Discussion and Future Work

Using Poisson processes, we constructed greedy Poisson rejection sampling. We proved the correctness of the algorithm and analyzed its runtime, and showed that it could be used to obtain a channel simulation protocol. We then developed several variations on it, analyzed their runtimes, and showed that they could all be used to obtain channel simulation protocols. As the most significant result of the paper, we showed that using the binary search variant of GPRS we can achieve $\mathcal{O}(D_{\mathrm{KL}}[Q\|P])$ runtime for arbitrary one-dimensional, unimodal density ratios, significantly improving upon the previous best $\mathcal{O}(D_\infty[Q\|P])$ bound by A* coding.

There are several interesting directions for future work. From a practical perspective, the most pressing question is whether efficient channel simulation algorithms exist for multivariate problems; finding an efficient channel simulation protocol for multivariate Gaussians would already have far-reaching practical consequences.

References

Eirikur Agustsson and Lucas Theis. Universally quantized neural compression. *Advances in Neural Information Processing Systems*, 33, 2020.

Anonymous. Faster relative entropy coding with greedy rejection coding. *See the Supplementary Materials*, 2023.

Robert M Corless, Gaston H Gonnet, David EG Hare, David J Jeffrey, and Donald E Knuth. On the lambert w function. *Advances in Computational mathematics*, 5:329–359, 1996.

Gergely Flamich and Lucas Theis. Adaptive greedy rejection sampling. *arXiv preprint arXiv:2304.10407*, 2023.

Gergely Flamich, Marton Havasi, and José Miguel Hernández-Lobato. Compressing images by encoding their latent representations with relative entropy coding. *Advances in Neural Information Processing Systems*, 33, 2020.

Gergely Flamich, Stratis Markou, and José Miguel Hernández-Lobato. Fast relative entropy coding with A* coding. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 6548–6577. PMLR, 17–23 Jul 2022. URL `https://proceedings.mlr.press/v162/flamich22a.html`.

Prahladh Harsha, Rahul Jain, David McAllester, and Jaikumar Radhakrishnan. The communication complexity of correlation. In *Twenty-Second Annual IEEE Conference on Computational Complexity (CCC'07)*, pp. 10–23. IEEE, 2007.

Marton Havasi, Robert Peharz, and José Miguel Hernández-Lobato. Minimal random code learning: Getting bits back from compressed model parameters. In *International Conference on Learning Representations*, 2018.

Mahmoud Hegazy and Cheuk Ting Li. Randomized quantization with exact error distribution. In *2022 IEEE Information Theory Workshop (ITW)*, pp. 350–355. IEEE, 2022.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *International Conference on Learning Representations*, 2014. URL `https://arxiv.org/abs/1312.6114`.

J.F.C. Kingman. *Poisson Processes*. Oxford Studies in Probability. Clarendon Press, 1992. ISBN 9780191591242. URL `https://books.google.co.uk/books?id=VEiM-OtwDHkC`.

Cheuk Ting Li and Abbas El Gamal. Strong functional representation lemma and applications to coding theorems. *IEEE Transactions on Information Theory*, 64(11):6967–6978, 2018.

CA Maddison. Poisson process model for Monte Carlo. *Perturbation, Optimization, and Statistics*, pp. 193–232, 2016.

Chris J Maddison, Daniel Tarlow, and Tom Minka. A* sampling. *Advances in Neural Information Processing Systems*, 27:3086–3094, 2014.

Pat Muldowney, Krzysztof Ostaszewski, and Wojciech Wojdowski. The Darth Vader rule. *Tatra Mountains Mathematical Publications*, 52(1):53–63, 2012.

A. Shah, W.-N. Chen, J. Balle, P. Kairouz, and L. Theis. Optimal compression of locally differentially private mechanisms. In *Artificial Intelligence and Statistics*, 2022. URL `https://arxiv.org/abs/2111.00092`.

L. Theis, T. Salimans, M. D. Hoffman, and F. Mentzer. Lossy compression with Gaussian diffusion. arXiv:2206.08889, 2022. URL `https://arxiv.org/abs/2206.08889`.

Lucas Theis and Noureldin Yosri. Algorithms for the communication of samples. In *International Conference on Machine Learning*, 2022.

Jacob Ziv. On universal quantization. *IEEE Transactions on Information Theory*, 31(3):344–347, 1985.

A Measure-theoretic Construction of Greedy Poisson Rejection Sampling

In this section, we provide a construction of greedy Poisson rejection sampling (GPRS) in its greatest generality. However, we first establish the notation for the rest of the appendix.

Notation: In what follows, we will always denote GPRS's target distribution as Q and its proposal distribution as P. We assume that Q and P are Borel probability measures over some Polish space Ω with Radon-Nikodym derivative $r \overset{def}{=} dQ/dP$. We denote the standard Lebesgue measure on $\mathbb{R}^n$ by λ. All logarithms are assumed to be to the base 2 denoted by $\log_2$. Similarly, we use the less common notation of $\exp_2$ to denote exponentiation with base 2, i.e. $\exp_2(x) = 2^x$. The relative entropy from Q to P is defined as $D_{\mathrm{KL}}[Q\|P] \overset{def}{=} \int_\Omega \log_2 r(x)\, dP(x)$ and the Rényi ∞-divergence as $D_\infty[Q\|P] \overset{def}{=} \operatorname{ess\,sup}_{x\in\Omega} \{\log r(x)\}$, where the essential supremum is taken with respect to P.

Restricting a Poisson process "under the graph of a function": We first consider the class of functions "under which" we will be restricting our Poisson processes. Let $\varphi : \Omega \to \mathbb{R}^+$ be a measurable function, such that $|\varphi(\Omega)| = |r(\Omega)|$, i.e. the image of φ has the same cardinality as the image of $r = dQ/dP$. This implies that there exist bijections between $r(\Omega)$ and $\varphi(\Omega)$. Since both images are subsets of $\mathbb{R}^+$, both images have a linear order. Hence, a monotonically increasing bijection $\tilde{\sigma}$ exists such that $\varphi = \tilde{\sigma} \circ r$. Since $\tilde{\sigma}$ is monotonically increasing, we can extend it to a monotonically increasing continuous function $\sigma : \mathbb{R}^+ \to \mathbb{R}^+$. This fact is significant, as it allows us to reduce questions about functions of interest from Ω to $\mathbb{R}^+$ to invertible functions on the positive reals. Since σ is continuous and monotonically increasing on the positive reals, it is invertible on the extended domain. We now consider restricting a Poisson process under the graph of φ, and work out the distribution of the spatial coordinate of the first arrival.

Let $\Pi = \{(T_n, X_n)\}_{n=1}^\infty$ be a Poisson process over $\mathbb{R}^+ \times \Omega$ with mean measure $\mu = \lambda \times P$. Let $U \overset{def}{=} \{(t, x) \in \Omega \mid t \leq \varphi(x)\}$ be the set of points "under the graph" of φ. By the restriction theorem (Kingman, 1992), $\tilde{\Pi} \overset{def}{=} \Pi \cap U$ is a Poisson process with mean measure $\tilde{\mu}(A) = \mu(A \cap U)$ for an arbitrary Borel set A. As a slight abuse of notation, we define the shorthand $\tilde{\mu}(t) = \tilde{\mu}([0, t) \times \Omega)$ for the average number of points in $\tilde{\Pi}$ up to time t. Let $\mathbf{N}_{\tilde{\Pi}}$ denote the counting measure of $\tilde{\Pi}$ and let $(\tilde{T}, \tilde{X})$ be the first arrival of $\tilde{\Pi}$, i.e. the first point of the original process Π that falls under the graph of φ, *assuming that it exists*. To develop GPRS, we first work out the distribution of $\tilde{X}$ for an arbitrary φ.

Remark: Note, that the construction below only holds if the first arrival is guaranteed to exist, otherwise the quantities below do not make sense. From a formal point of view, we would have to always condition on the event $\mathbf{N}_{\tilde{\Pi}}(\infty) \overset{def}{=} \lim_{t\to\infty} \mathbf{N}_{\tilde{\Pi}}(t) > 0$, e.g. we would need to write $\mathbb{P}[\tilde{T} \geq t \mid \mathbf{N}_{\tilde{\Pi}}(\infty) > 0]$. However, we make a "leap of faith" instead and assume that our construction is valid to obtain our "guesses" for the functions σ and φ. Then, we show that for our specific guesses the derivation below is well-defined, i.e. namely that $\mathbb{P}[\mathbf{N}_{\tilde{\Pi}}(\infty) > 0] = 1$, which then implies $\mathbb{P}[\tilde{T} \geq t \mid \mathbf{N}_{\tilde{\Pi}}(\infty) > 0] = \mathbb{P}[\tilde{T} \geq t]$. To do this, note that

$$\begin{aligned}\mathbb{P}[\mathbf{N}_{\tilde{\Pi}}(\infty) > 0] &= 1 - \mathbb{P}[\mathbf{N}_{\tilde{\Pi}}(\infty) = 0] &(10)\\ &= 1 - \lim_{t\to\infty} e^{-\tilde{\mu}(t)}. &(11)\end{aligned}$$

Hence, for the well-definition of our construction it is enough to show that for our choice of σ, $\tilde{\mu}(t) \to \infty$ as $t \to \infty$.

Constructing σ: We wish to derive

$$\frac{d\mathbb{P}[\tilde{X} = x]}{dP} = \int_0^\infty \frac{d\mathbb{P}[\tilde{T} = t, \tilde{X} = x]}{d(\lambda \times P)}\, dt. \tag{12}$$

To do this, recall that we defined $\mathbf{N}_{\tilde{\Pi}}(t) \overset{def}{=} \mathbf{N}_{\tilde{\Pi}}([0, t) \times \Omega)$ and define $\mathbf{N}_{\tilde{\Pi}}(s, t) \overset{def}{=} \mathbf{N}_{\tilde{\Pi}}([s, t] \times \Omega)$. Similarly, let $\tilde{\mu}(t) \overset{def}{=} \mathbb{E}[\mathbf{N}_{\tilde{\Pi}}(t)]$ and $\tilde{\mu}(s,t) \overset{def}{=} \mathbb{E}[\mathbf{N}_{\tilde{\Pi}}(s,t)] = \tilde{\mu}(t) - \tilde{\mu}(s)$. Then,

$$\begin{aligned}\mathbb{P}[\tilde{X} \in A, \tilde{T} \in dt] &= \lim_{s \to t} \mathbb{P}[\tilde{X} \in A, \tilde{T} \in [s, t]] &(13)\\ &= \lim_{s \to t} \mathbb{P}[\tilde{X} \in A, \mathbf{N}_{\tilde{\Pi}}(s,t) = 1, \mathbf{N}_{\tilde{\Pi}}(s) = 0] &(14)\\ &= \lim_{s \to t} \mathbb{P}[\tilde{X} \in A \mid \mathbf{N}_{\tilde{\Pi}}(s,t) = 1]\mathbb{P}[\mathbf{N}_{\tilde{\Pi}}(s,t) = 1]\mathbb{P}[\mathbf{N}_{\tilde{\Pi}}(s) = 0], &(15)\end{aligned}$$

where the last equality holds, because $\mathbf{N}_{\tilde{\Pi}}(s) \perp \mathbf{N}_{\tilde{\Pi}}(s,t)$, since $[0, s) \times \Omega$ and $[s, t] \times \Omega$ are disjoint. By Lemma 3 from Maddison (2016),

$$\begin{aligned}\mathbb{P}[\tilde{X} \in A \mid \mathbf{N}_{\tilde{\Pi}}(s,t) = 1] &= \mathbb{P}[(\tilde{T}, \tilde{X}) \in [s,t] \times A \mid \mathbf{N}_{\tilde{\Pi}}(s,t) = 1] &(16)\\ &= \frac{\tilde{\mu}([s,t] \times A)}{\tilde{\mu}(s,t)}. &(17)\end{aligned}$$

Substituting this back into Equation (15) and using the fact that $\mathbf{N}_{\tilde{\Pi}}$ is Poisson, we find

$$\begin{aligned}\mathbb{P}[\tilde{X} \in A, \tilde{T} \in dt] &= \lim_{s \to t} \left(\frac{\tilde{\mu}([s,t] \times A)}{\tilde{\mu}(s,t)} \cdot \tilde{\mu}(s,t) e^{-\tilde{\mu}(s,t)} \cdot e^{-\tilde{\mu}(s)} \right) \Big/ (t - s) &(18)\\ &= \lim_{s \to t} \left(\tilde{\mu}([s,t] \times A) \cdot e^{-(\tilde{\mu}(t) - \tilde{\mu}(s))} \cdot e^{-\tilde{\mu}(s)} \right) \Big/ (t - s) &(19)\\ &= e^{-\tilde{\mu}(t)} \cdot \lim_{s \to t} \frac{\tilde{\mu}([s,t] \times A)}{t - s} &(20)\\ &= e^{-\tilde{\mu}(t)} \cdot \int_A \mathbf{1}[t \leq \varphi(x)]\, dP(x), &(21)\end{aligned}$$

where the last equation holds by the definition of the derivative and the fundamental theorem of calculus. From this, by inspection we find

$$\frac{d\mathbb{P}[\tilde{T} = t, \tilde{X} = x]}{d(\lambda \times P)} = \mathbf{1}[t \leq \varphi(x)] e^{-\tilde{\mu}(t)} = \mathbf{1}[t \leq \varphi(x)] \mathbb{P}[\tilde{T} \geq t]. \tag{22}$$

Finally, by marginalizing out the first arrival time, we find that the spatial distribution is

$$\begin{aligned}\frac{d\mathbb{P}[\tilde{X} = x]}{dP} &= \int_0^\infty \frac{d\mathbb{P}[\tilde{T} = t, \tilde{X} = x]}{d(\lambda \times P)}\, dt &(23)\\ &= \int_0^{\varphi(x)} \mathbb{P}[\tilde{T} \geq t]\, dt. &(24)\end{aligned}$$

Deriving φ by finding σ: Note that Equation (24) holds for any φ. However, to get a correct algorithm, we need to set it such that $\frac{d\mathbb{P}[\tilde{X} = x]}{dP} = \frac{dQ}{dP} = r$. Since we can write $\varphi = \sigma \circ r$ by our earlier argument, this problem is reduced to finding an appropriate invertible continuous function $\sigma : \mathbb{R}^+ \to \mathbb{R}^+$. Thus, we wish to find σ such that

$$\forall x \in \Omega, \quad r(x) = \int_0^{\sigma(r(x))} \mathbb{P}[\tilde{T} \geq t]\, dt. \tag{25}$$

Now, introduce $\tau = \sigma(r(x))$, so that $\sigma^{-1}(\tau) = r(x)$. Note that this substitution only makes sense for $\tau \in r(\Omega)$. However, since we are free to extend σ^{-1} to $\mathbb{R}^+$ in any we like so long as it is monotone, we may require that this equation hold for all $\tau \in \mathbb{R}^+$. Then, we find

$$\begin{aligned}\sigma^{-1}(\tau) &= \int_0^\tau \mathbb{P}[\tilde{T} \geq t]\, dt &(26)\\ \Rightarrow \left(\sigma^{-1}\right)'(\tau) &= \mathbb{P}[\tilde{T} \geq \tau] &(27)\end{aligned}$$

with $\sigma^{-1}(0) = 0$. Before we solve for σ, we define

$$\begin{aligned}w_P(h) &\overset{def}{=} \mathbb{P}_{Y \sim P}[r(Y) \geq h] &(28)\\ w_Q(h) &\overset{def}{=} \mathbb{P}_{Y \sim Q}[r(Y) \geq h]. &(29)\end{aligned}$$

Note that w_P and w_Q are supported on $[0, r^*)$, where $r^* \overset{def}{=} \operatorname{ess\,sup}_{x \in \Omega}\{r(x)\} = \exp_2(D_\infty[Q \| P])$. Now, we can rewrite Equation (27) as

$$
\begin{aligned}
\left(\sigma^{-1}\right)'(\tau) &= \mathbb{P}[\tilde{T} \geq t] && (30)\\
&= \mathbb{P}[\tilde{T} \geq t, \varphi(\tilde{X}) \geq t] + \underbrace{\mathbb{P}[\tilde{T} \geq t, \varphi(\tilde{X}) < t]}_{=0 \text{ due to mutual exclusivity}} && (31)\\
&= \underbrace{\mathbb{P}[\tilde{T} \geq 0, \varphi(\tilde{X}) \geq t]}_{=\mathbb{P}[\varphi(\tilde{X}) \geq t], \text{ since } \tilde{T} \geq 0 \text{ always}} - \mathbb{P}[\varphi(\tilde{X}) \geq t \geq \tilde{T}] && (32)\\
&= \mathbb{P}[r(\tilde{X}) \geq \sigma^{-1}(t)] - \mathbb{P}[r(\tilde{X}) \geq \sigma^{-1}(t) \geq \sigma^{-1}(\tilde{T})] && (33)\\
&= w_Q(\sigma^{-1}(t)) - \sigma^{-1}(t) w_P(\sigma^{-1}(t)) && (34)
\end{aligned}
$$

where the second term in the last equation follows by noting that

$$
\begin{aligned}
\mathbb{P}[r(\tilde{X}) \geq \sigma^{-1}(t) \geq \sigma^{-1}(\tilde{T})] &= \int_\Omega \int_0^t \underbrace{\mathbf{1}[r(x) \geq \sigma^{-1}(t)]\mathbf{1}[r(x) \geq \sigma^{-1}(\tau)]}_{=\mathbf{1}[r(x) \geq \sigma^{-1}(t)], \text{ since } \tau < t} \mathbb{P}[\tilde{T} \geq \tau]\, d\tau\, dP(x) && (35)\\
&= \int_\Omega \mathbf{1}[r(x) \geq \sigma^{-1}(t)] \underbrace{\int_0^t \mathbb{P}[\tilde{T} \geq \tau]\, d\tau}_{=\sigma^{-1}(t), \text{by eq. (26)}} dP(x) && (36)\\
&= \sigma^{-1}(t) w_P(\sigma^{-1}(t)). && (37)
\end{aligned}
$$

Now, by the inverse function theorem, we get

$$
\sigma'(h) = \frac{1}{w_Q(h) - h \cdot w_P(h)}, \tag{38}
$$

thus we finally find

$$
\sigma(h) = \int_0^h \frac{1}{w_Q(\eta) - \eta \cdot w_P(\eta)}\, d\eta. \tag{39}
$$

Thus, to recapitulate, setting $\varphi = \sigma \circ r$, where σ is given by Equation (39) will ensure that the spatial distribution of the first arrival of Π under φ is the target distribution Q.

Well-definition of GPRS and Algorithm 3 terminates with probability 1: To ensure that our construction is useful, we need to show that the first arrival under the graph $\tilde{T}$ exists and is almost surely finite, so that Algorithm 3 terminates with probability 1. First, note that for any $t > 0$, we have $\tilde{\mu}(t) \leq \mu(t) = t < \infty$. Since $\mathbb{P}[\mathbf{N}_{\tilde{\Pi}}(t) = 0] = e^{-\tilde{\mu}(t)}$, this implies that for any finite time t, $\mathbb{P}[\mathbf{N}_{\tilde{\Pi}}(t) = 0] > 0$. Second, note that for $h \in [0, r^*]$,

$$
w_Q(h) - h \cdot w_P(h) = \int_\Omega \mathbf{1}[r(x) \geq h](r(x) - h)\, dP(x), \tag{40}
$$

from which

$$
\begin{aligned}
w_Q(r^*) - r^* w_P(r^*) &= \int_\Omega \mathbf{1}[r(x) \geq r^*](r(x) - r^*)\, dP(x) && (41)\\
&= \int_\Omega \mathbf{1}[r(x) = r^*](r(x) - r^*)\, dP(x) && (42)\\
&= 0. && (43)
\end{aligned}
$$

Thus, in particular, we find that

$$
\lim_{h \to r^*} e^{-\tilde{\mu}(\sigma(h))} = \lim_{h \to r^*} \mathbb{P}[\mathbf{N}_{\tilde{\Pi}}(\sigma(h)) = 0] \overset{\text{eq. (34)}}{=} \lim_{h \to r^*} (w_Q(h) - h \cdot w_P(h)) = 0. \tag{44}
$$

By continuity, this can only hold if $\tilde{\mu}(\sigma(h)) \to \infty$ as $h \to r^*$. Since $\tilde{\mu}(t)$ is finite for all $t > 0$ and σ is monotonically increasing, this also implies that $\sigma(h) \to \infty$ as $h \to r^*$. Thus, we have shown a couple of facts:

- $\sigma(h) \to \infty$ as $h \to r^*$, hence φ is always unbounded at points in Ω that achieve the supremum r^*. Furthermore, this implies that
$$\sigma^{-1}(t) \to r^* \quad \text{as} \quad t \to \infty. \tag{45}$$
Since σ^{-1} is increasing, this implies that it is bounded from above by r^*.
- $\tilde{\mu}(t) < \infty$ for all $t > 0$, but $\tilde{\mu}(t) \to \infty$ as $t \to \infty$. Hence, by Equation (11) we have $\mathbb{P}[\mathbf{N}_{\tilde{\Pi}}(\infty) > 0] = 1$. Thus, the first arrival of $\tilde{\Pi}$ exists almost surely, and our construction is well-defined. In particular, it is meaningful to write $\mathbb{P}[\tilde{T} \geq t]$.
- $\mathbb{P}[\tilde{T} \geq t] = \mathbb{P}[\mathbf{N}_{\tilde{\Pi}}(t) = 0] \to 0$ as $t \to \infty$, which shows that the first arrival time is finite with probability 1. In turn, this implies that Algorithm 3 will terminate with probability one, as desired.

Note, that w_P and w_Q can be computed in many practically relevant cases, see Appendix G.

B Analysis of Greedy Poisson Rejection Sampling

Now that we constructed a correct sampling algorithm in Appendix A, we turn our attention to deriving the expected first arrival time $\tilde{T}$ in the restricted process $\tilde{\Pi}$, the expected number of samples N before Algorithm 3 terminates and bound on N's variance, and an upper bound on its coding cost. The proofs below showcase the true advantage of formulating the sampling algorithm using the language of Poisson processes: the proofs are all quite short and elegant.

B.1 The Expected First Arrival Time

At the end of the previous section, we showed that $\tilde{T}$ is finite with probability one, which implies that $\mathbb{E}[\tilde{T}] < \infty$. Now, we derive the value of this expectation exactly. Since $\tilde{T}$ is a positive random variable, by the Darth Vader rule (Muldowney et al., 2012), we may write its expectation as
$$\mathbb{E}\left[\tilde{T}\right] = \int_0^\infty \mathbb{P}[\tilde{T} \geq t]\, dt \overset{\text{eq. (26)}}{=} \lim_{t\to\infty} \sigma^{-1}(t) \overset{\text{eq. (45)}}{=} r^*. \tag{46}$$

B.2 The Expectation and Variance of the Runtime

Expectation of N: First, let $\tilde{\Pi}^C \overset{def}{=} \Pi \setminus \tilde{\Pi}$ be the set of points in Π above the graph of φ. Since $\tilde{\Pi}$ and $\tilde{\Pi}^C$ are defined on complementary sets, they are independent. Since by definition $\tilde{T}$ is the first arrival of $\tilde{\Pi}$ and the Nth arrival of Π, it must mean that the first $N-1$ arrivals of Π occurred in $\tilde{\Pi}^C$. Thus, conditioned on $\tilde{T}$, we have $N - 1 = |\{(T, X) \in \tilde{\Pi} \mid T < \tilde{T}\}|$ is Poisson distributed with mean
$$\mathbb{E}\left[N - 1 \mid \tilde{T}\right] = \int_0^{\tilde{T}} \int_\Omega \mathbf{1}[t \geq \varphi(x)]\, dP(x)\, dt \tag{47}$$
$$= \int_0^{\tilde{T}} \int_\Omega 1 - \mathbf{1}[t < \varphi(x)]\, dP(x)\, dt \tag{48}$$
$$= \tilde{T} - \tilde{\mu}\left(\tilde{T}\right). \tag{49}$$
By the law of iterated expectations,
$$\mathbb{E}[N-1] = \mathbb{E}_{\tilde{T}}\left[\mathbb{E}\left[N - 1 \mid \tilde{T}\right]\right] \tag{50}$$
$$= \mathbb{E}\left[\tilde{T}\right] - \mathbb{E}\left[\tilde{\mu}\left(\tilde{T}\right)\right]. \tag{51}$$
Focusing on the second term, we find
$$\mathbb{E}\left[\tilde{\mu}\left(\tilde{T}\right)\right] = \int_0^\infty \tilde{\mu}(t) \cdot \mathbb{P}[\tilde{T} \in dt]\, dt \tag{52}$$
$$= \int_0^\infty \tilde{\mu}(t) \cdot \tilde{\mu}'(t) e^{-\tilde{\mu}(t)}\, dt \tag{53}$$
$$= \int_0^\infty u e^{-u}\, du = 1, \tag{54}$$

where the third equality follows by substituting $u = \tilde{\mu}(t)$. Finally, plugging the above and Equation (46) into Equation (51), we find

$$\mathbb{E}[N] = 1 + \mathbb{E}[N-1] = 1 + r^* - 1 = r^*. \tag{55}$$

Variance of N: We now show that the distribution of N is *super-Poissonian*, i.e. $\mathbb{E}[N] \leq \mathbb{V}[N]$. Similarly to the above, we begin with the law of iterated variances to find

$$\begin{aligned}\mathbb{V}[N] = \mathbb{V}[N-1] &= \mathbb{E}_{\tilde{T}}[\mathbb{V}[N-1 \mid \tilde{T}]] + \mathbb{V}_{\tilde{T}}[\mathbb{E}[N-1 \mid \tilde{T}]] &(56)\\ &= \mathbb{E}\left[\tilde{T}\right] - \mathbb{E}\left[\tilde{\mu}\left(\tilde{T}\right)\right] + \mathbb{V}\left[\tilde{T}\right] + \mathbb{V}\left[\tilde{\mu}\left(\tilde{T}\right)\right], &(57)\end{aligned}$$

where the second equality follows from the fact that the variance of $N-1$ matches its mean conditioned on $\tilde{T}$, since it is a Poisson random variable. Focussing on the last term, we find

$$\begin{aligned}\mathbb{V}\left[\tilde{\mu}\left(\tilde{T}\right)\right] &= \mathbb{E}\left[\tilde{\mu}\left(\tilde{T}\right)^2\right] - \mathbb{E}\left[\tilde{\mu}\left(\tilde{T}\right)\right]^2 &(58)\\ &= \int_0^\infty \tilde{\mu}(t)^2 \cdot \mathbb{P}[\tilde{T} \in dt]\, dt - 1 &(59)\\ &= \int_0^\infty u^2 e^{-u}\, dt - 1 = 1, &(60)\end{aligned}$$

where the third equality follows from a similar u-substitution as in Equation (54). Thus, putting everything together, we find

$$\mathbb{V}[N] = r^* - 1 + \mathbb{V}\left[\tilde{T}\right] + 1 = r^* + \mathbb{V}\left[\tilde{T}\right] \geq \mathbb{E}[N]. \tag{61}$$

B.3 The Codelength of the Index

As discussed in Section 3, Alice and Bob can realize a one-shot channel simulation protocol using GPRS for a pair of correlated random variables $\mathbf{x}, \mathbf{y} \sim P_{\mathbf{x},\mathbf{y}}$ if they have access to shared randomness and can simulate samples from $P_{\mathbf{x}}$. In particular, after receiving $\mathbf{y} \sim P_{\mathbf{y}}$, Alice runs GPRS with proposal distribution $P_{\mathbf{x}}$ and target $P_{\mathbf{x}|\mathbf{y}}$, and the shared randomness to simulate the Poisson process Π. She then encodes the index N of the first arrival of Π under the graph of φ. Bob can decode Alice's sample $\mathbf{x} \sim P_{\mathbf{x}|\mathbf{y}}$ by simulating the same N samples from Π using the shared randomness. The question is, how efficiently can Alice encode N? We answer this question by following the approach of Li & El Gamal (2018). Namely, we first bound the conditional expectation $\mathbb{E}[\log_2 N \mid \mathbf{y} = y]$, after which we average over $\mathbf{y}$ to bound $\mathbb{E}[\log_2 N]$. Then, we use the maximum entropy distribution subject to the constraint of fixed $\mathbb{E}[\log_2 N]$ to bound $\mathbb{H}[N]$. Finally, noting that $\mathbf{x}$ is a function of Π and N, we get

$$\begin{aligned}\mathbb{H}[\mathbf{x}, N \mid \Pi] &= \underbrace{\mathbb{H}[\mathbf{x} \mid N, \Pi]}_{=0} + \mathbb{H}[N \mid \Pi] &(62)\\ &= \underbrace{\mathbb{H}[N \mid \mathbf{x}, \Pi]}_{=0} + \mathbb{H}[\mathbf{x} \mid \Pi], &(63)\end{aligned}$$

from which

$$\mathbb{H}[N] \geq \mathbb{H}[N \mid \Pi] = \mathbb{H}[\mathbf{x} \mid \Pi] \geq \mathbb{H}[\mathbf{x} \mid \Pi], \tag{64}$$

which will finish the proof.

Bound on the conditional expectation: Fix $\mathbf{y}$ and set $Q = P_{\mathbf{x}|\mathbf{y}}$ and $P = P_{\mathbf{x}}$ as GPRS's target and proposal distribution, respectively. Let $(t, x) \in \mathbb{R}^+ \times \Omega$ be a point under the graph of φ, i.e. $\sigma^{-1}(t) \leq r(x)$. Then,

$$\begin{aligned}r(x) &\geq \sigma^{-1}(t) &(65)\\ &\overset{\text{eq. (27)}}{=} \int_0^t \mathbb{P}[\tilde{T} \geq \tau]\, d\tau &(66)\\ &\geq t \cdot \inf_{\tau \in [0,t]} \left\{\mathbb{P}[\tilde{T} \geq \tau]\right\} &(67)\\ &= t \cdot \mathbb{P}[\tilde{T} \geq t]. &(68)\end{aligned}$$

From this, we get

$$t + 1 \leq \frac{r(x)}{\mathbb{P}[\tilde{T} \geq t]} + 1 \leq \frac{r(x) + 1}{\mathbb{P}[\tilde{T} \geq t]}. \tag{69}$$

Next, conditioning on the first arrival $(\tilde{T}, \tilde{X})$ we get

$$\begin{aligned}
\mathbb{E}[\log_2 N \mid \tilde{T} = t, \tilde{X} = x] &\leq \log_2 \left(\mathbb{E}[N - 1 \mid \tilde{T} = t, \tilde{X} = x] + 1 \right) && (70)\\
&\overset{\text{eq. (49)}}{=} \log_2 \left(t - \tilde{\mu}\left(t\right) + 1\right) && (71)\\
&\leq \log_2 \left(t + 1\right) && (72)\\
&\overset{\text{eq. (69)}}{\leq} \log_2 \left(r(x) + 1\right) - \log_2 \mathbb{P}[\tilde{T} \geq t] && (73)\\
&= \log_2 \left(r(x) + 1\right) + \tilde{\mu}\left(t\right) \cdot \log_2 e, && (74)
\end{aligned}$$

where the first inequality follows by Jensen's inequality.

Now, by the law of iterated expectations, we find

$$\begin{aligned}
\mathbb{E}[\log_2 N] &= \mathbb{E}_{\tilde{X}} \left[\mathbb{E}_{\tilde{T} \mid \tilde{X}} \left[\mathbb{E}[\log_2 N \mid \tilde{T}, \tilde{X}] \right] \right] && (75)\\
&\overset{\text{eq. (74)}}{\leq} \mathbb{E}_{\tilde{X}} \left[\log_2 \left(r\left(\tilde{X}\right) + 1 \right) \right] + \underbrace{\mathbb{E}_{\tilde{T}} \left[\tilde{\mu}\left(\tilde{T}\right) \right]}_{=1 \text{ by eq. (54)}} \cdot \log_2 e && (76)\\
&= D_{\mathrm{KL}}[Q \| P] + \mathbb{E}_{\tilde{X}} \left[\log_2 \left(1 + \frac{1}{r\left(\tilde{X}\right)} \right) \right] + \log_2 e && (77)\\
&\leq D_{\mathrm{KL}}[Q \| P] + \mathbb{E}_{\tilde{X}} \left[\log_e \left(e^{-r(\tilde{X})} \right) \right] \cdot \log_2 e + \log_2 e && (78)\\
&= D_{\mathrm{KL}}[Q \| P] + 2 \cdot \log_2 e, && (79)
\end{aligned}$$

where the second inequality follows from switching to the natural base and using a first-order Taylor approximation for e^{-x}.

Bound on the marginal expectation and entropy: Equation (79) is a one-shot bound, which yields

$$\mathbb{E}[\log_2 N \mid \mathbf{y}] \leq D_{\mathrm{KL}}[P_{\mathbf{x}|\mathbf{y}} \| P_{\mathbf{x}}] + 2 \cdot \log_2 e. \tag{80}$$

Taking expectation over $\mathbf{y} \sim P_{\mathbf{y}}$, we get

$$\mathbb{E}[\log_2 N] \leq I[\mathbf{x}; \mathbf{y}] + 2 \cdot \log_2 e. \tag{81}$$

Finally, following Proposition 4 in Li & El Gamal (2018), the maximum entropy distribution for N subject to a constraint on $\mathbb{E}[\log_2 N]$ obeys

$$\mathbb{H}[N] \leq \mathbb{E}[\log_2 N] + \log\left(\mathbb{E}[\log_2 N] + 1\right) + 1. \tag{82}$$

Plugging in Equation (81) into the above, we get

$$\begin{aligned}
\mathbb{H}[N] &\leq I[\mathbf{x}; \mathbf{y}] + \log_2(I[\mathbf{x}; \mathbf{y}] + 2 \log_2 e + 1) + 2 \log_2 e + 1 && (83)\\
&\leq I[\mathbf{x}; \mathbf{y}] + \log_2(I[\mathbf{x}; \mathbf{y}] + 1) + 2 \log_2 e + 1 + \log_2(2 \log_2 e + 1) && (84)\\
&< I[\mathbf{x}; \mathbf{y}] + \log_2(I[\mathbf{x}; \mathbf{y}] + 1) + 6. && (85)
\end{aligned}$$

Similarly to Li & El Gamal (2018), we can encode N using a Zeta distribution $\zeta(n \mid s) \propto n^{-s}$ with

$$s \overset{def}{=} \frac{1}{I[\mathbf{x}; \mathbf{y}] + 2 \log_2 e}. \tag{86}$$

With this choice of the coding distribution, the expected codelength of N is upper bounded by $I[\mathbf{x}; \mathbf{y}] + \log_2(I[\mathbf{x}; \mathbf{y}] + 1) + 7$ bits.

C Analysis of Parallel GPRS

Parallel GPRS (PGPRS) is a general technique to accelerate GPRS when parallel computing power is available and is presented in Section 3.1. Assuming that J parallel threads are available, for target distribution Q and proposal P, PGPRS simulates J Poisson processes $\Pi_1, \dots, \Pi_J$ in parallel, all of them with mean measure $\mu_i \overset{def}{=} \frac{\lambda}{J} \times P$. Assuming $\left\{\left(T_{N_1}^{(1)}, X_{N_1}^{(1)}\right), \dots, \left(T_{N_J}^{(J)}, X_{N_J}^{(J)}\right)\right\}$ are the first arrivals of $\Pi_1, \dots, \Pi_J$ under φ, respectively, PGPRS selects the arrival with the overall smallest arrival time $J^* = \arg\min_{j \in \{1,\dots,J\}} \left\{T_{N_j}^{(j)}\right\}$. By the superposition theorem (Kingman, 1992), $\cup_{j=1}^{J} \Pi_j = \Pi$ is a Poisson process with mean measure $\mu = \lambda \times P$, hence $\left(T_{N_{J^*}}^{(J^*)}, X_{N_{J^*}}^{(J^*)}\right)$ will be the first arrival of Π under the graph of φ and the measure-theoretic construction in Appendix A therefore guarantees the correctness of PGPRS.

Expected runtime: We will proceed similarly to the analysis in Appendix B.2. Concretely, let $\tilde{T} = T_{N_{J^*}}^{(J^*)}$ be the first arrival time of Π under the graph of φ. Let

$$\nu_j \overset{def}{=} \left|\left\{(T, X) \in \Pi_j \mid T < \tilde{T}, T > \varphi(X)\right\}\right| \tag{87}$$

be the number of points in Π_j that are rejected before the global first arrival occurs. By an independence argument analogous to the one given in Appendix B.2, the $\nu_1, \dots, \nu_J$ are independent Poisson random variables, each distributed with mean

$$\mathbb{E}\left[\nu_j \mid \tilde{T}\right] = \frac{1}{J}\left(\tilde{T} - \tilde{\mu}\left(\tilde{T}\right)\right). \tag{88}$$

Hence, by the law of iterated expectations, we find that we get

$$\mathbb{E}\left[\nu_j\right] \overset{\text{eq. (51)}}{=} \frac{r^* - 1}{J} \tag{89}$$

rejections in each of the J threads on average before the global first arrival occurs. Now, a thread j terminates either when the global first arrival occurs in it or when the thread's current arrival time $T_{n_j}^{(j)}$ provably exceeds the global first arrival time. Thus, on average, each thread will have one more rejection compared to Equation (89). Hence the average runtime of a thread is

$$\mathbb{E}[\nu_j + 1] = \frac{r^* - 1}{J} + 1, \tag{90}$$

and the average number of samples simulated by PGPRS across all its threads is

$$J \cdot \left(\frac{r^* - 1}{J} + 1\right) = r^* + J - 1. \tag{91}$$

Variance of Runtime: Once again, similarly to Appendix B.2, we find by the law of iterated variances that the variance of the runtime in each of the j threads is

$$\mathbb{V}[\nu_j + 1] = \mathbb{V}[\nu_j] = \frac{r^* - 1}{J} + \frac{1}{J^2}\left(\mathbb{V}\left[\tilde{T}\right] + 1\right), \tag{92}$$

meaning that we make an $\mathcal{O}(1/J)$ reduction in the variance of the runtime compared to regular GPRS.

Codelength: PGPRS can also realize a one-shot channel simulation protocol for a pair of correlated random variables $\mathbf{x}, \mathbf{y} \sim P_{\mathbf{x},\mathbf{y}}$. For a fixed $\mathbf{y} \sim P_{\mathbf{y}}$, Alice applies PGPRS to the target $Q = P_{\mathbf{x}|\mathbf{y}}$ and proposal $P = P_{\mathbf{x}}$, and encodes the two-part code (J^*, N_{J^*}). Bob can then simulate N_{J^*} samples from Π_{J^*} and recover Alice's sample.

Encoding J^*: By the symmetry of the setup, the global first arrival will occur with equal probability in each subprocess Π_j. Hence J^* follows a uniform distribution on $\{1, \dots J\}$. Therefore, Alice can encode J^* optimally using $\lceil \log_2 J \rceil$ bits.

Encoding N_{J^*}: We can develop a bound using an almost identical argument to the one in Appendix B.3. In particular, by adapting the conditional bound in Equation (74) appropriately using Equation (90), we get

$$\mathbb{E}\left[\log_2 N_{J^*} \mid \tilde{T} = t, \tilde{X} = x, J^*\right] \leq \log_2(r(x) + 1) + \tilde{\mu}(t) \cdot \log_2 e - \log_2 J. \tag{93}$$

Then, using this conditional bound and adapting Equation (79), we find

$$\mathbb{E}\left[\log_2 N_{J^*} \mid \mathbf{y}, J^*\right] \leq D_{\mathrm{KL}}[P_{\mathbf{x}|\mathbf{y}} \| P_{\mathbf{x}}] + 2 \cdot \log_2 e - \log_2 J \tag{94}$$

to obtain a one-shot bound. Taking expectation over $\mathbf{y} \sim P_{\mathbf{y}}$, we get

$$\mathbb{E}\left[\log_2 N_{J^*} \mid J^*\right] \leq I[\mathbf{x}; \mathbf{y}] + 2 \cdot \log_2 e - \log_2 J, \tag{95}$$

hence, by adapting the maximum entropy bound in Equation (85), we find

$$\mathbb{H}[N_{J^*} \mid J^*] < I[\mathbf{x}; \mathbf{y}] - \log_2 J + \log(I[\mathbf{x}; \mathbf{y}] - \log_2 J + 1) + 6. \tag{96}$$

Thus, we finally find that the entropy of the two-part code (J^*, N_{J^*}) is upper bounded by

$$\mathbb{H}[J^*, N_{J^*}] < I[\mathbf{x}; \mathbf{y}] + \log(I[\mathbf{x}; \mathbf{y}] - \log_2 J + 1) + 7. \tag{97}$$

Using a Zeta distribution $\zeta(n \mid s) \propto n^{-s}$ to encode $N_{J^*} \mid J^*$ with

$$s \overset{def}{=} \frac{1}{I[\mathbf{x}; \mathbf{y}] + 2 \cdot \log_2 e - \log_2 J}, \tag{98}$$

we find that the expected codelength of the two-part code is upper bounded by

$$\mathbb{H}[J^*, N_{J^*}] < I[\mathbf{x}; \mathbf{y}] + \log(I[\mathbf{x}; \mathbf{y}] - \log_2 J + 1) + 8 \text{ bits}. \tag{99}$$

D Simulating Poisson Processes Using Tree-Structured Partitions

In this section, we examine an advanced simulation technique for Poisson processes, which is required to formulate the binary search-based variants of GPRS. We first recapitulate the tree-based simulation technique from (Flamich et al., 2022) and some important results from their work. Then, we present Algorithm 7, using which we can finally formulate the optimally efficient variant of GPRS in Appendix E. **Note:** For simplicity, we present the ideas for Poisson processes whose spatial measure P is non-atomic. These ideas can also be extended to atomic spatial measures with appropriate modifications.

Splitting functions: Let Π be a spatiotemporal Poisson process on some space Ω with non-atomic spatial measure P. In this paragraph, we will not deal with Π itself yet, but the space on which it is defined and the measure P. Now, assume that there is a function `split`, which for any given Borel set $B \subseteq \Omega$, produces a (possibly random) *P-essential partition* of B consisting of two Borel sets, i.e.

$$\texttt{split}(B) = \{L, R\}, \text{ such that } L \cap R = \emptyset \text{ and } P(L \cup R) = P(B). \tag{100}$$

The last condition simply allows `split` to discard some points from B that will not be included in either the left or right split. For example, if we wish to design a splitting function for a subset of $B \subseteq \mathbb{R}^d$ to split B along some hyperplane, we do not need to include the points on the hyperplane in either set. The splitting function that always exists is the *trivial* splitting function, which just returns the original set and the empty set:

$$\texttt{split}_{\text{trivial}}(B) = \{B, \emptyset\}. \tag{101}$$

A more interesting example when $\Omega = \mathbb{R}$ is the *on-sample* splitting function, where for a $\mathbb{R}$-valued random variable $X \sim P|_B$ it splits B as

$$\texttt{split}_{\text{on-samp}}(B) = \{(-\infty, X) \cap B, (X, \infty) \cap B\}. \tag{102}$$

This function is used by Algorithm 5 and AS* coding (Flamich et al., 2022). Another example is the *dyadic* splitting function operating on a bounded interval (a, b), splitting as

$$\texttt{split}_{\text{dyad}}((a, b)) = \left\{\left(a, \frac{a+b}{2}\right), \left(\frac{a+b}{2}, b\right)\right\}. \tag{103}$$

We call this dyadic because when we apply this splitting function to $(0, 1)$ and subsets produced by applying it, we get the set of all intervals with dyadic endpoints on $(0, 1)$, i.e. $\{(0, 1), (0, 1/2), (1/2, 1), (0, 1/4), \ldots\}$. This splitting function is used by Algorithm 6 and AD* coding. As one might imagine, there might be many more possible splitting functions than the two examples we give above, all of which might be more or less useful in practice.

Algorithm 7: Simulating a tree construction with another

```
Input  : Proposal distribution $P$
         Target splitting function $\texttt{split}_{\text{target}}$,
         Simulating splitting function $\texttt{split}_{\text{sim}}$,
         Target heap index $H_{\text{target}}$
Output : Arrival $(T, X)$ with heap index $H_{\text{target}}$ in $\mathcal{T}_{\text{target}}$, and heap index $H_{\text{sim}}$ of the arrival in
         $\mathcal{T}_{\text{sim}}$.
$\mathcal{P} \leftarrow \text{PriorityQueue}$
$B \leftarrow \Omega$
$K \leftarrow \lfloor \log_2 H_{\text{target}} \rfloor$
$X \sim P$
$T \sim \text{Exp}(1)$
$\mathcal{P}.\texttt{push}(T, X, \Omega, 1)$
for $k = 0, \ldots, K$ do
    repeat
        $T, X, C, H \leftarrow \mathcal{P}.\texttt{pop}()$
        $L, R \leftarrow \texttt{split}_{\text{sim}}(C)$
        if $B \cap L \neq \emptyset$ then
            $\Delta_L \sim \text{Exp}(P(L))$
            $T_L \leftarrow T + \Delta_L$
            $X_L \sim P|_L$
            $\mathcal{P}.\texttt{push}(T_L, X_L, L, 2H)$
        end
        if $B \cap R \neq \emptyset$ then
            $\Delta_R \sim \text{Exp}(P(R))$
            $T_R \leftarrow T + \Delta_R$
            $X_R \sim P|_R$
            $\mathcal{P}.\texttt{push}(T_R, X_R, R, 2H + 1)$
        end
    until $X \in B$            /* Exit loop when we find first arrival in $B$.  */
    if $k < K$ then
        $\{B_0, B_1\} \leftarrow \texttt{split}_{\text{target}}(B)$
        $d \leftarrow \left\lceil \frac{H_{\text{target}}}{2^{K-k}} \right\rceil \mod 2$
        $B \leftarrow B_d$
    end
end
return $T, X, H$
```

Split-induced binary space partitioning tree: Every splitting function on a space Ω induces an infinite set of subsets by repeatedly applying the splitting function to the splits it produces. These sets can be organised into an infinite *binary space partitioning tree (BSP-tree)*, where each node in the tree is represented by a set produced by `split` and an unique index. Concretely, let the root of the tree be represented by the whole space Ω and the index $H_{\text{root}} = 1$. Now we recursively construct the rest of the tree as follows: Let (B, H) be a node in the tree, with B a Borel set and H its index, and let $\{L, R\} = \texttt{split}(B)$ the left end right splits of B. Then, we set $(L, 2H)$ as the node's left child and $(R, 2H + 1)$ as its right child. We refer to the index associated with each node as its *heap index*.

Heap-indexing the points in Π and a strict heap invariant: As we saw in Section 2.1, each point in Π can be uniquely identified by their *time index* N, i.e. if we time-order the points of Π, N represents the Nth arrival in the ordered list. However, we can also uniquely index each point in Π using a splitting function-induced BSP-tree as follows.

We extend each node in the BSP-tree with a point from Π, such that the extended tree satisfies a strict *heap invariant*: First, we extend the root node $(\Omega, 1)$ by adjoining Π's first arrival (T_1, X_1) and the first arrival index 1 to get $(T_1, X_1, \Omega, 1, 1)$. Then, for every other extended node (T, X, B, H, N) with parent node $(T', X', B', \lfloor H/2 \rfloor, M)$ we require that (T, X) is the first arrival in the restricted process $\Pi \cap ((T', \infty) \times B)$. This restriction enforces that we always have that $T' < T$ and, therefore,

$M < N$. Furthermore, it is strict because there are no other points of Π in B between those two arrival times.

Notation for the tree structure: Let us denote the extended BSP tree $\mathcal{T}$ on Π induced by `split`. Each node $\nu \in \mathcal{T}$ is a tuple $\nu = (T, X, B, H, N)$ consisting of the *arrival time* T, *spatial coordinate* X, its *bounds* B, `split`*-induced heap index* H and *time index* N. As a slight overload of notation, for a node ν, we use it as subscript to refer to its elements, e.g. T_ν is the arrival time associated with node ν. We now prove the following important result, which ensures that we do not lose any information by using $\mathcal{T}$ to represent Π. An essentially equivalent statement was first proven by for Gumbel processes (Appendix, "Equivalence under *partition*" subsection; Maddison et al., 2014).

Lemma D.1. *Let* Π*,* P*,* `split` *and* $\mathcal{T}$ *be as defined above. Then,* P*-almost surely,* $\mathcal{T}$ *contains every point of* Π.

Proof. We give an inductive argument.

Base case: the first arrival of Π is by definition contained in the root node of $\mathcal{T}$.

Inductive hypothesis: assume that the first N arrivals of Π are all contained in $\mathcal{T}$.

Case for $N + 1$*:* We begin by observing that if the first N nodes are contained in $\mathcal{T}$, they must form a connected subtree $\mathcal{T}_N$. To see this, assume the contrary, i.e. that the first N arrivals form a subgraph $\Gamma_N \subseteq \mathcal{T}$ with multiple disconnected components. Let $\nu \in \Gamma_N$ be a node contained in a component of Γ_N that is *disconnected* from the component of Γ_N containing the root node. Since $\mathcal{T}$ is a tree, there is a unique path π from the root node to ν in $\mathcal{T}$, and since ν and the root are disconnected in Γ_N, π must contain a node $c \notin \Gamma_N$. However, since c is an ancestor of ν, by the heap invariant of $\mathcal{T}$ we must have that the time index of c is $N_c < N_\nu \leq N$ hence $c \in \Gamma_N$, a contradiction.

Thus, let $\mathcal{T}_N$ represent the subtree of $\mathcal{T}$ containing the first N arrivals of Π. Now, let $\mathcal{F}_N$ represent the *frontier* of $\mathcal{T}_N$, i.e. the leaf nodes' children:

$$\mathcal{F}_N \overset{def}{=} \{\nu \in \mathcal{T} \mid \nu \notin \mathcal{T}_N, \mathrm{parent}(\nu) \in \mathcal{T}_N\}, \tag{104}$$

where parent retrieves the parent of a node in $\mathcal{T}$. Let

$$\bar{\Omega}_N \overset{def}{=} \bigcup_{\nu \in \mathcal{F}_N} B_\nu \tag{105}$$

be the union of all the bounds of the nodes in the frontier. A simple inductive argument shows that for all N the nodes in $\mathcal{F}_N$ provide a P-essential partition of Ω, from which $P(\bar{\Omega}_N) = 1$. Let T_N be the Nth arrival time of Π. Now, by definition, the arrival time associated with every node in the frontier $\mathcal{F}_N$ must be later than T_N. Finally, consider the first arrival time across the nodes in the frontier:

$$T_N^* \overset{def}{=} \min_{\nu \in \mathcal{F}_N} T_\nu. \tag{106}$$

Then, conditioned on $\mathcal{T}_N$, T_N^* is the first arrival of Π restricted to $(T_N, \infty) \times \bar{\Omega}$, thus it is P-almost surely the $N + 1$st arrival in Π, as desired. □

A connection between the time index an heap index of a node: Now that we have two ways of uniquely identifying each point in Π it is natural to ask whether there is any relation between them. In the next paragraph we adapt an argument from Flamich et al. (2022) to show that under certain circumstances, the answer is yes.

First, we need to define two concepts, the *depth* of a node in an extended BSP-tree and a *contractive* splitting function. Thus, let $\mathcal{T}$ be an extended BSP-tree over Π induced by `split`. Let $\nu \in \mathcal{T}$. Then, the depth D of ν in $\mathcal{T}$ is defined as the distance from ν to the root. A simple argument shows that the heap index H_ν of every node ν at depth D in $\mathcal{T}$ is between $2^D \leq H_\nu < 2^{D+1}$. Thus, we can easily obtain the depth of a node ν from the heap index via the formula $D_\nu = \lfloor \log_2 H_\nu \rfloor$. Next, we say that `split` is *contractive* if **all** the bounds it produces shrink on average. More formally, let $\epsilon < 1$, and for a node $\nu \in \mathcal{T}$, let $\mathcal{A}_\nu$ denote the set of ancestor nodes of ν. Then, `split` is contractive if for every non-root node $\nu \in \mathcal{T}$ we have

$$\mathbb{E}_{\mathcal{T}_{\mathcal{A}_\nu} \mid D_\nu}[P(B_\nu)] \leq \epsilon^{D_\nu}, \tag{107}$$

where $\mathcal{T}_{\mathcal{A}_\nu}$ and denotes the subtree of $\mathcal{T}$ containing the arrivals of the ancestors of ν.

Note that $\epsilon \geq 1/2$ for any `split` function. This is because if $\{L, R\} = \texttt{split}(B)$ for some set B, then $P(R) = P(B) - P(L)$. Thus, if $P(R) = \alpha P(B)$, then $P(L) = (1-\alpha)P(B)$, and by definition $\epsilon = \max\{\alpha, 1-\alpha\}$, which is minimized when $\alpha = 1/2$, from which $\epsilon = 1/2$.

For example, $\texttt{split}_{\text{dyad}}$ is contractive with $\epsilon = 1/2$, while $\texttt{split}_{\text{trivial}}$ is not contractive. By Lemma 1 from Flamich et al. (2022), $\texttt{split}_{\text{on-samp}}$ is also contractive with $\epsilon = 3/4$. We now strengthen this result using a simple argument, and show that $\texttt{split}_{\text{on-samp}}$ is contractive with $\epsilon = 1/2$.

Lemma D.2. *Let ν, D be defined as above, let P be a non-atomic probability measure over $\mathbb{R}$ with CDF F_P. Let Π a $(1, P)$-Poisson process and $\mathcal{T}$ be the BSP-tree over Π induced by $\texttt{split}_{\textit{on-samp}}$. Then,*

$$\mathbb{E}_{\mathcal{T}_{\mathcal{A}_\nu} \mid D_\nu}[P(B_\nu)] = 2^{-D_\nu}. \tag{108}$$

Proof. Fix $D_\nu = d$, and let $\mathcal{N}_d = \{n \in \mathcal{T} \mid D_n = d\}$ be the set of nodes in $\mathcal{T}$ whose depth is d. Note, that $|\mathcal{N}_d| = 2^d$. We will show that

$$\forall n, m \in \mathcal{N}_d: \quad P(B_n) \overset{\mathrm{d}}{=} P(B_m), \tag{109}$$

i.e. that the distributions of the bound sizes are all the same. From this, we will immediately have $\mathbb{E}[P(B_n)] = \mathbb{E}[P(B_\nu)]$ for every $n \in \mathcal{N}_d$. Then, using the fact, that the nodes in $\mathcal{N}_d$ for a P-almost partition of Ω, we get:

$$1 = \mathbb{E}\left[P\left(\bigcup_{n \in \mathcal{N}_d} B_n\right)\right] = \mathbb{E}\left[\sum_{n \in \mathcal{N}_d} P(B_n)\right] = \sum_{n \in \mathcal{N}_d} \mathbb{E}\left[P(B_n)\right] = |\mathcal{N}_d| \cdot \mathbb{E}\left[P(B_\nu)\right]. \tag{110}$$

Dividing the very left and very right by $|\mathcal{N}_d| = 2^d$ yields the desired result.

To complete the proof, we now show that by symmetry, Equation (109) holds. We begin by exposing the fundamental symmetry of $\texttt{split}_{\text{on-samp}}$: for a node ν with left child L and right child R, the left and right bound sizes are equal in distribution:

$$P(B_L) \overset{\mathrm{d}}{=} P(B_r) \mid P(B_\nu). \tag{111}$$

First, note that by definition, all involved bounds will be intervals. Namely, assume that $B_\nu = (a, b)$ for some $a < b$ and X_ν is the sample associated with ν. Then, $B_L = (a, X_\nu)$ and $B_R = (X_\nu, b)$ and hence $P(B_L) = F_P(X_\nu) - F_P(a)$. Since $X_\nu \sim P|_{B_\nu}$, by the probability integral transform, we have $F(X_\nu) \sim \mathrm{Unif}(F_P(a), F_P(b))$, from which $P(B_L) \sim \mathrm{Unif}(0, F_P(b) - F_P(a)) = \mathrm{Unif}(0, P(B_\nu))$. Since $P(B_R) = P(B_\nu) - P(B_L)$, we similarly have $P(B_R) \sim \mathrm{Unif}(0, P(B_\nu))$, which establishes our claim.

Now, to show Equation (109), fix d and fix $n \in \mathcal{N}_d$. Let $\mathcal{A}_n$ denote the ancestor nodes of n. As we saw in the paragraph above,

$$P(B_n) \mid P(B_{\mathrm{parent}(n)}) \overset{\mathrm{d}}{=} P(B_{\mathrm{parent}(n)}) \cdot U, \quad U \sim \mathrm{Unif}(0,1), \tag{112}$$

regardless of whether n is a left or a right child of its parent. We can apply this d times to the ancestors of n find the marginal:

$$P(B_n) \overset{\mathrm{d}}{=} \prod_{i=1}^{d} U_i, \quad U_i \sim \mathrm{Unif}(0,1). \tag{113}$$

Since the choice of n was arbitrary, all nodes in $\mathcal{N}_d$ have this distribution, which is what we wanted to show. $\square$

Now, we have the following result.

Lemma D.3. *Let* `split` *be a contractive splitting function for some $\epsilon \in [1/2, 1)$. Then, for every node ν in $\mathcal{T}$ with time index N_ν and depth D_ν, we have*

$$\mathbb{E}_{D_\nu \mid N_\nu}[D_\nu] \leq -\log_\epsilon N_\nu. \tag{114}$$

Proof. Let us examine the case $N_\nu = 1$ first. In this case, ν is the root of $\mathcal{T}$ and has depth $D_\nu = 0$ by definition. Thus, $0 \leq -\log_\epsilon 1$ holds trivially.

Now, fix $\nu \in \mathcal{T}$ with time index $N_\nu = N > 1$. Let $\mathcal{T}_{N-1}$ be the subtree of $\mathcal{T}$ containing the first $N-1$ arrivals, $\mathcal{F}_{N-1}$ be the frontier of $\mathcal{T}_{N-1}$ and T_{N-1} the $(N-1)$st arrival time. Then, as we saw in Lemma D.1, the Nth arrival in Π occurs in one of the nodes in the frontier $\mathcal{F}_{N-1}$, after T_{N-1}. In particular, conditioned on $\mathcal{T}_{N-1}$, the arrival times associated with each node $f \in \mathcal{F}_{N-1}$ will be shifted exponentials $T_f = T_{N-1} + \mathrm{Exp}(P(B_f))$, and the Nth arrival time in Π is the minimum of these: $T_\nu = T_N = \min_{f \in \mathcal{F}_{N-1}} T_f$. It is a standard fact (see e.g. Lemma 6 in Maddison (2016)) that the index of the minimum

$$F_N = \operatorname*{arg\,min}_{f \in \mathcal{F}_{N-1}} T_f \tag{115}$$

is independent of $T_\nu = T_{F_N}$ and $\mathbb{P}[F_N = f \mid \mathcal{T}_{N-1}] = P(B_f)$. A simple inductive argument shows that the number of nodes on the frontier $|\mathcal{F}_{N-1}| = N$. Thus, we have a simple upper bound on the entropy of F:

$$\mathbb{E}_{F_N \mid \mathcal{T}_{N-1}, N_\nu = N}[-\log_2 P(B_\nu)] = \mathbb{H}[F_N \mid \mathcal{T}_{N-1}, N_\nu = N] \leq \log_2 N. \tag{116}$$

Thus, taking expectation over $\mathcal{T}_{N-1}$, we find

$$\begin{align}
\log_2 N &\overset{\text{eq. (116)}}{\geq} \mathbb{E}_{F_N, \mathcal{T}_{N-1} \mid N_\nu = N}[-\log_2 P(B_\nu)] \tag{117}\\
&= \mathbb{E}_{D_\nu \mid N_\nu = N}\left[\mathbb{E}_{F_N, \mathcal{T}_{N-1} \mid D_\nu, N_\nu = N}[-\log_2 P(B_\nu)]\right] \tag{118}\\
&\geq \mathbb{E}_{D_\nu \mid N_\nu = N}\left[-\log_2 \mathbb{E}_{F_N, \mathcal{T}_{N-1} \mid D_\nu, N_\nu = N}[P(B_\nu)]\right] \tag{119}\\
&\overset{\text{eq. (107)}}{\geq} \mathbb{E}_{D_\nu \mid N_\nu = N}\left[-\log_2 \epsilon^{D_\nu}\right] \tag{120}\\
&= (-\log_2 \epsilon) \cdot \mathbb{E}_{D_\nu \mid N_\nu = N}[D_\nu]. \tag{121}
\end{align}$$

The second inequality holds by Jensen's inequality. In the third inequality, we apply Equation (107), and one might worry about conditioning on N_ν here. However, this is not an issue because

$$\begin{align}
&\mathbb{E}_{F_N, \mathcal{T}_{N-1} \mid D_\nu = d, N_\nu = N}[P(B_\nu)] \tag{122}\\
&\quad = \mathbf{1}[d \leq N-1] \cdot \sum_{f \in \mathcal{F}_{N-1}} \mathbb{E}_{\mathcal{T}_{A_f} \mid F_N = f, D_\nu = d}[P(B_f)] \cdot \mathbb{P}[F_N = f \mid D_f = d] \tag{123}\\
&\quad \overset{\text{eq. (107)}}{\leq} \mathbf{1}[d \leq N-1] \cdot \sum_{f \in \mathcal{F}_{N-1}} \epsilon^d \cdot \mathbb{P}[F_N = f \mid D_f = d] \tag{124}\\
&\quad = \mathbf{1}[d \leq N-1] \cdot \epsilon^d. \tag{125}
\end{align}$$

Thus, we finally get

$$(-\log_2 \epsilon) \cdot \mathbb{E}_{D_\nu \mid N_\nu = N}[D_\nu] \leq \log_2 N \quad \Rightarrow \quad \mathbb{E}_{D_\nu \mid N_\nu}[D_\nu] \leq -\log_\epsilon N_\nu \tag{126}$$

by dividing both sides by $-\log_2 \epsilon$ and we obtain the desired result. ☐

Converting between different heap indices: Assume now that we have two splitting functions, $\texttt{split}_{\text{target}}$ and $\texttt{split}_{\text{sim}}$, which induce their own BSP-ordering on Π, $\mathcal{T}_{\text{target}}$ and $\mathcal{T}_{\text{sim}}$. Now, given a $\texttt{split}_{\text{target}}$-induced heap index H, Algorithm 7 presents a method for simulating the appropriate node $\nu \in \mathcal{T}_{\text{target}}$ by simulating nodes from $\mathcal{T}_{\text{sim}}$. In other words, given a node with some heap index induced by a splitting function, Algorithm 7 lets us find the heap index of the same arrival induced by a different splitting function. The significance of Algorithm 7 is that it lets us develop convenient search methods using a given splitting function, but it might be more efficient to encode the heap index induced by another splitting function.

Theorem D.4. *Let Π, $\texttt{split}_{\textit{target}}$, $\texttt{split}_{\textit{sim}}$, $\mathcal{T}_{\textit{target}}$ and $\mathcal{T}_{\textit{sim}}$ be as above. Let $\nu \in \mathcal{T}_{\textit{target}}$ with and let $(T_{\textit{sim}}, X_{\textit{sim}}, H_{\textit{sim}})$ be the arrival and its heap index output by Algorithm 7 given the above as input as well as H_ν as the target index. Then, Algorithm 7 is correct, in the sense that*

$$T_\nu \overset{d}{=} T_{\textit{sim}} \quad \textit{and} \quad X_\nu \overset{d}{=} X_{\textit{sim}}, \tag{127}$$

and $H_{\textit{sim}}$ is the heap index of $(T_{\textit{sim}}, X_{\textit{sim}})$ in $\mathcal{T}_{\textit{sim}}$.

Proof. First, observe that when $\texttt{split}_{\text{target}} = \texttt{split}_{\text{sim}}$, Algorithm 7 collapses onto just the extended BSP tree construction for Π and simply returns the arrival with the given heap index H_{target} in $\mathcal{T}_{\text{target}}$. In particular, the inner loop will always exit after one iteration, and every time one and only one of the conditional blocks will be executed. In other words, in this case, the algorithm becomes equivalent to Algorithm 2 in Flamich et al. (2022).

Let us now consider the case when $\texttt{split}_{\text{target}} \neq \texttt{split}_{\text{sim}}$. Denote the depth of the required node by $D_\nu = \lfloor \log_2 H_\nu \rfloor$. Now, we give an inductive argument for correctness.

Base case: Consider $D_\nu = 0$. In this case, the target bounds $B = \Omega$, and the first sample we draw $X \sim P$ is guaranteed to fall in Ω. Hence, for $D_\nu = 0$ the outer loop only runs for one iteration. Furthermore, during that iteration, the inner loop will also exit after one iteration, and Algorithm 7 returns the sample (T, X) sampled before the outer loop with heap index 1. Since $T \sim \mathrm{Exp}(1)$ and $X \sim P$, this will be a correctly distributed output with the appropriate heap index.

Inductive hypothesis: Assume Algorithm 7 is correct heap indices with depths up to $D_\nu = d$.

Case $D_\nu = d + 1$: Let $\rho \in \mathcal{T}_{\text{target}}$ be the parent node of ν with arrival (T_ρ, X_ρ). Then, $D_\rho = d$, hence by the inductive hypothesis, Algorithm 7 will correctly simulate a branch $\mathcal{T}_{\text{target}}$ up to node ρ. At the end of the dth iteration of the outer loop Algorithm 7 sets the target bounds $B \leftarrow B_\nu$. Then, in the final, $d + 1$st iteration, the inner loop simply realizes $\mathcal{T}_{\text{sim}}$ and accepts the first sample after X that falls inside B_ν whose time index $T > T_\rho$. Due to the priority queue, the loop simulates the nodes of $\mathcal{T}_{\text{sim}}$ in time order; hence the accepted sample will also be the one with the earliest arrival time. Furthermore, Algorithm 7 only ever considers nodes of $\mathcal{T}_{\text{sim}}$ whose bounds intersect the target bounds B_ν, hence the inner loop is guaranteed to terminate, which finishes the proof. ☐

E GPRS with Binary Search

We now utilise the machinery we developed in Appendix D to analyze Algorithm 5.

Correcntess of Algorithm 5: Observe that Algorithm 5 constructs the extended BSP tree for the on-sample splitting function. Thus, we will now focus on performing a binary tree search using the extended BSP tree induced by the on-sample splitting function, which we denote by $\mathcal{T}$. The first step of the algorithm matches GPRS's first step (Algorithm 3). Hence it is correct for the first step. Now consider the algorithm in an arbitrary step k before termination, where the candidate sample (T, X) is rejected, i.e. $T > \varphi(X)$. By assumption, the density ratio r is unimodal, and since σ is monotonically increasing, $\varphi = \sigma \circ r$ is unimodal too. Thus, let $x^* \in \Omega$ be such that $r(x^*) = r^*$, where $r^* = \exp_2(D_\infty[Q\|P])$. Assume for now that $X < x^*$, the case $X > x^*$ follows by a symmetric argument. By the unimodality assumption, since $X < x^*$, it must hold that for all $y < X$, we have $\varphi(y) < \varphi(X)$. Consider now the arrival (T_L, X_L) of Π in the current node's left child. Then, we will have $T < T_L$ and $X_L < X$ by construction. Thus, finally, we get

$$\varphi(X_L) < \varphi(X) < T < T_L, \tag{128}$$

meaning that the current node's left child is also guaranteed to be rejected. This argument can be easily extended to show that any left-descendant of the current node will be rejected, and it is sufficient to search its right-descendants only. By a similar argument, when $X > x^*$, we find that it is sufficient only to check the current node's left-descendants. Finally, since both algorithms simulate Π and search for its first arrival under φ, by the construction in Appendix A, the sample returned by both algorithms will follow the desired target distribution.

Expected runtime and codelength: Since Algorithm 5 simulates a single branch of $\mathcal{T}_{\text{on-sample}}$, its runtime is equal to the runtime of simulating that single branch. Assume that the accepted sample's time index is N and its depth is D. Then, since Algorithm 5 draws one sample per depth, its runtime will be exactly D steps. Then, by putting together lemmas D.2 and D.3, we get

$$\mathbb{E}_{D|N}[D] \leq \log_2 N. \tag{129}$$

Then, by taking expectation over N, by Equation (79) we get

$$\mathbb{E}[D] \leq D_{\mathrm{KL}}[Q\|P] + 2 \cdot \log_2 e < D_{\mathrm{KL}}[Q\|P] + 3, \tag{130}$$

thus in the one-shot case, the runtime of Algorithm 5 is linear in the KL divergence, which establishes Theorem 3.5.

For the codelength result, let $\mathbf{x}, \mathbf{y} \sim P_{\mathbf{x},\mathbf{y}}$, fix $\mathbf{y}$ and set $Q \leftarrow P_{\mathbf{x}|\mathbf{y}}$ and $P \leftarrow P_{\mathbf{x}}$. Let H denote the heap index of the sample returned by Algorithm 5, and D its depth. Recall, that $\lfloor \log_2 H \rfloor = D$, hence by Equation (130), we get

$$\mathbb{E}_H[\log_2 H] < \mathbb{E}_H[\lfloor \log_2 H \rfloor] + 1 = \mathbb{E}_D[D] + 1 \leq D_{\mathrm{KL}}[P_{\mathbf{x}|\mathbf{y}} \| P_{\mathbf{x}}] + 1 + 2\log_2 e. \tag{131}$$

Taking expectation over $\mathbf{y}$, we get

$$\mathbb{E}[\log_2 H] \leq I[\mathbf{x}; \mathbf{y}] + 1 + 2\log_2 e. \tag{132}$$

Therefore, using similar arguments to the ones in Appendix B.3, we use a Zeta distribution to encode H with

$$\lambda = 1 + \frac{1}{I[\mathbf{x}; \mathbf{y}] + 1 + 2\log_2 e}. \tag{133}$$

This gives an upper bound of

$$\begin{aligned}
\mathbb{H}[\mathbf{x} \mid \Pi] &\leq \mathbb{H}[H] && (134)\\
&\leq I[\mathbf{x}; \mathbf{y}] + \log_2(I[\mathbf{x}; \mathbf{y}] + 2\log_2 e + 2) + 2\log_2 e + 2 && (135)\\
&\leq I[\mathbf{x}; \mathbf{y}] + \log_2(I[\mathbf{x}; \mathbf{y}] + 1) + 2\log_2 e + 2 + \log_2(2\log_2 e + 2) && (136)\\
&\leq I[\mathbf{x}; \mathbf{y}] + \log_2(I[\mathbf{x}; \mathbf{y}] + 1) + 8 && (137)
\end{aligned}$$

F General GPRS with Binary Search

We finally present a generalized version of branch-and-bound GPRS (Algorithm 6) for more general spaces and remove the requirement that r be unimodal.

Decomposing a Poisson process into a mixture of processes: Let Π be a process over $\mathbb{R}^+ \times \Omega$ as before, with spatial measure P, and let Q be a target measure, $\varphi = \sigma \circ r$ and $U = \{(t, x) \in \mathbb{R}^+ \times \Omega \mid t \leq \varphi(x)\}$ as before. Let $\tilde{\Pi}$ be $\Pi = \Pi \cap U$ restricted under φ and $(\tilde{T}_1, \tilde{X}_1)$ its first arrival. Let $\{L, R\}$ form an P-essential partiton of Ω, i.e. $L \cap R = \emptyset$ and $P(L \cup R) = 1$, and let $\Pi_L = \Pi \cap \mathbb{R}^+ \times L$ and $\Pi_R = \Pi \cap \mathbb{R}^+ \times R$ be the restriction of Π to L and R, respectively. Let $\tilde{\Pi}_L = \Pi_L \cap U$ and $\tilde{\Pi}_R = \Pi_R \cap U$ be the restrictions of the two processes under φ as well. Let $\tilde{\mu}_L(t)$ and $\tilde{\mu}_R(t)$ be the mean measures of these processes. Thus, the first arrival times in these processes have survival functions

$$\mathbb{P}[\tilde{T}_1^L \geq t] = e^{-\tilde{\mu}_L(t)} \tag{138}$$

$$\mathbb{P}[\tilde{T}_1^R \geq t] = e^{-\tilde{\mu}_R(t)}. \tag{139}$$

Note, that by the superposition theorem, $\Pi_L \cup \Pi_R = \Pi$, hence the first arrival $(\tilde{T}_1, \tilde{X}_1)$ occurs in either Π_L or Π_R. Assume now that we have already searched through the points of Π up to time τ without finding the first arrival. At this point, we can ask: will the first arrival occur in Π_L, given that $\tilde{T}_1 \geq \tau$? Using Bayes' rule, we find

$$\mathbb{P}[\tilde{X}_1 \in L \mid \tilde{T}_1 \geq \tau] = \frac{\mathbb{P}[\tilde{X}_1 \in L, \tilde{T}_1 \geq \tau]}{\mathbb{P}[\tilde{T}_1 \geq \tau]}. \tag{140}$$

More generally, assume that the first arrival of Π occurs in some set $A \subseteq \Omega$, and we know that the first arrival time is larger than τ. Then, what is the probability that the first arrival occurs in some set $B \subseteq A$? Similarly to the above, we find

$$\begin{aligned}
\mathbb{P}[\tilde{X}_1 \in B \mid \tilde{T}_1 \geq \tau, \tilde{X}_1 \in A] &= \frac{\mathbb{P}[\tilde{X}_1 \in B, \tilde{T}_1 \geq \tau, \tilde{X}_1 \in A]}{\mathbb{P}[\tilde{T}_1 \geq \tau, \tilde{X}_1 \in A]} && (141)\\
&= \frac{\mathbb{P}[\tilde{T}_1 \geq \tau, \tilde{X}_1 \in B]}{\mathbb{P}[\tilde{T}_1 \geq \tau, \tilde{X}_1 \in A]}. && (142)
\end{aligned}$$

Let $(\tilde{T}_L, \tilde{X}_L)$ be the first arrival of Π_L. Then, the crucial observation is that

$$(\tilde{T}_L, \tilde{X}_L) \overset{d}{=} \tilde{T}_1, \tilde{X}_1 \mid \tilde{X}_1 \in L. \tag{143}$$

This enables us to search for the first arrival of Π under the graph of φ using an extended BSP tree construction. At each node, if we reject, we draw a Bernoulli random variable b with mean equal to the probability that the first arrival occurs within the bounds associated with the right child node. Then, if $b = 1$, we continue the search along the right branch. Otherwise, we search along the left branch.

Note, however, that in a restricted process Π_A, the spatial measure no longer integrates to 1. Furthermore, our target Radon-Nikodym derivative is $r(x) \cdot \mathbf{1}[x \in A]/Q(A)$. This means we need to change the graph φ to some new graph φ_A to ensure that the spatial distribution of the returned sample is still correct. Therefore, for a set A we define the restricted versions of previous quantities:

$$\tilde{\mu}_A(t) \overset{def}{=} \int_0^t \int_A \mathbf{1}[\tau \leq \varphi_A(x)]\, dP(x)\, dt \tag{144}$$

$$w_P(h \mid A) \overset{def}{=} \int_A \mathbf{1}[h \leq r(x)]\, dP(x) \tag{145}$$

Then, via analogous arguments to the ones in Appendix A, we find

$$\frac{d\mathbb{P}[\tilde{T}_A = t, \tilde{X}_A = x]}{d(\lambda \times P)} = \mathbf{1}[x \in A]\mathbf{1}[t \leq \varphi_A(x)]\mathbb{P}[\tilde{T}_A \geq t] \tag{146}$$

$$\frac{d\mathbb{P}[\tilde{X}_A = x]}{dP} = \mathbf{1}[x \in A] \int_0^{\varphi_A(x)} \mathbb{P}[\tilde{T}_A \geq t]\, dt \tag{147}$$

$$\varphi_A = \sigma_A \circ r. \tag{148}$$

Similarly, setting $\frac{d\mathbb{P}[\tilde{X}_A = x]}{dP} = \mathbf{1}[x \in A] \cdot r(x)/Q(A)$, and setting $\tau = \sigma_A(r(x))$, we get

$$\sigma_A^{-1}(\tau) = Q(A) \int_0^{\tau} \mathbb{P}[\tilde{T}_A \geq t]\, dt \tag{149}$$

$$\Rightarrow \quad \left(\sigma_A^{-1}\right)'(\tau) = \mathbb{P}[\tilde{T}_A \geq \tau] \tag{150}$$

$$= \mathbb{P}[\tilde{T} \geq \tau, \tilde{X} \in A]. \tag{151}$$

From this, again using similar arguments to the ones in Appendix A, we find

$$\left(\sigma_A^{-1}\right)'(\tau) = w_Q(\sigma_A^{-1}(t) \mid A) - \sigma_A^{-1}(t) \cdot w_P(\sigma_A^{-1}(t) \mid A). \tag{152}$$

G Necessary Quantities for Implementing GPRS in Practice

Ultimately, given a target-proposal pair (Q, P) with density ratio r, we would want an easy-to-evaluate expression for the appropriate stretch function σ or σ^{-1} to plug directly into our algorithms. Computing σ requires computing the integral in Equation (3) and finding σ^{-1} requires solving the non-linear ODE in Equation (2), neither of which is usually possible in practice. Hence, we usually resort to computing σ^{-1} numerically by using an ODE solver for Equation (2).

In any case, we need to compute w_P and w_Q, which are analytically tractable in the practically relevant cases. Hence, in this section, we give closed-form expressions for w_P and w_Q for all the examples we consider and give closed-form expressions for σ and σ^{-1} for some of them. If we do not give a closed-form expression of σ, we use numerical integration to compute σ^{-1} instead.

G.1 Uniform-Uniform Case

Let P be the uniform distribution over an arbitrary space Ω and Q a uniform distribution supported on some subset $\mathcal{X} \subset \Omega$, with $P(\mathcal{X}) = C$ for some $C \leq 1$ Then,

$$r(x) = \frac{1}{C} \cdot \mathbf{1}[x \in \mathcal{X}] \tag{153}$$

$$w_P(h) = C \tag{154}$$

$$w_Q(h) = 1 \tag{155}$$

$$\sigma(h) = -\frac{1}{C} \log\left(1 - C \cdot h\right) \tag{156}$$

$$\sigma^{-1}(t) = \frac{1}{C}\left(1 - \exp(-C \cdot h)\right). \tag{157}$$

Note that using GPRS in the uniform-uniform case is somewhat overkill, as in this case, it is simply equivalent to standard rejection sampling.

G.2 Triangular-Uniform Case

Let $P = \mathrm{Unif}(0, 1)$ and for some numbers $0 < a < c < b < 1$, let Q be the triangular distribution, defined by the PDF

$$q(x) = \begin{cases} 0 & \text{if } x < a \\ \frac{2(x-a)}{(b-a)(c-a)} & \text{if } a \le x < c \\ \frac{2}{b-a} & \text{if } x = c \\ \frac{2(b-x)}{(b-a)(b-c)} & \text{if } c < x \le b \\ 0 & \text{if } b < x. \end{cases} \tag{158}$$

For convenience, let $\ell = b - a$. Then,

$$r(x) = q(x) \tag{159}$$

$$w_P(x) = \ell - \frac{\ell^2}{2} \cdot h \tag{160}$$

$$w_Q(x) = 1 - \frac{\ell^2}{4} \cdot h^2 \tag{161}$$

$$\sigma(h) = \frac{2h}{2 - \ell \cdot h} \tag{162}$$

$$\sigma^{-1}(t) = \frac{2t}{2 + \ell \cdot t} \tag{163}$$

G.3 Finite Discrete-Discrete / Piecewise Constant Case

Without loss of generality, let Q be a discrete distribution over a finite set with K elements defined by the probability vector $(r_1 < r_2 < \ldots < r_K)$ and $P = \mathrm{Unif}([1 : K])$ the uniform distribution on K elements. Note that any discrete distribution pair can be rearranged into this setup. Now, we can embed this distribution into $[0, 1]$ by mapping P to $\hat{P} = \mathrm{Unif}(0, 1)$ the uniform distribution over $[0, 1]$ and mapping Q to $\hat{Q}$ with density

$$\hat{q}(x) = \sum_{k=1}^{K} \mathbf{1}\left[\lfloor K \cdot x \rfloor = k - 1\right] \cdot K \cdot r_k. \tag{164}$$

Then, we can draw a sample x from Q by simulating a sample $\hat{x} \sim \hat{Q}$ and computing $x = K \cdot \lfloor \hat{x}/K \rfloor + 1$. Hence, we can instead reduce the problem to sampling from a piecewise constant distribution. Thus, let us now instead present the more general case of arbitrary piecewise constant distributions over $[0, 1]$, with $\tilde{Q}$ defined by probabilities $(q_1 < q_2 < \ldots < q_K)$ and corresponding piece widths $(w_1 < w_2 < \ldots < w_K)$. We require, that $\sum_{k=1} q_k = \sum_{k=1}^{K} w_k = 1$. Then, the density is

$$\tilde{q}(x) = \sum_{k=1}^{K} \mathbf{1}\left[\sum_{j=1}^{k-1} w_j \le x \le \sum_{j=1}^{k} w_j\right] \cdot \frac{q_k}{w_k} \tag{165}$$

Define $r_k = q_k / w_k$. Then,

$$w_P(h) = \sum_{k=1}^{K} \mathbf{1}[h \le r_k] \cdot w_k \tag{166}$$

$$w_Q(h) = \sum_{k=1}^{K} \mathbf{1}[h \le r_k] \cdot w_k \cdot r_k \tag{167}$$

$$\sigma(h) = \sum_{k=1}^{K} \mathbf{1}[h \ge r_{k-1}] \cdot \frac{1}{B_k} \cdot \log\left(\frac{A_k - B_k r_{k-1}}{A_k - B_k \min\{r_k, h\}}\right) \tag{168}$$

where we defined

$$A_k = \sum_{j=k}^{K} w_j r_j = \sum_{j=k}^{K} q_j \tag{169}$$

$$B_k = \sum_{j=k}^{K} w_j \tag{170}$$

G.4 Diagonal Gaussian-Gaussian Case

Without loss of generality, let $Q = \mathcal{N}(\mu, \sigma^2 I)$ and $P = \mathcal{N}(0, I)$ be d-dimensional Gaussian distributions with diagonal covariance. As a slight abuse of notation, let $\mathcal{N}(x \mid \mu, \sigma^2 I)$ denote the probability density function of a Gaussian random variable with mean μ and covariance $\sigma^2 I$ evaluated at x. Then, when $\sigma^2 < 1$, we have

$$r(x) = Z \cdot \mathcal{N}\left(x \mid m, s^2 I\right) \tag{171}$$

$$m = \frac{\mu}{1 - \sigma^2} \tag{172}$$

$$s^2 = \frac{\sigma^2}{1 - \sigma^2} \tag{173}$$

$$Z = \frac{(1 - \sigma^2)^{-d}}{\mathcal{N}(\mu \mid 0, (1 - \sigma^2) I)} \tag{174}$$

$$w_P(h) = \mathbb{P}\left[\chi^2\left(d, \|m\|^2\right) \leq -2s^2 \ln h + C\right] \tag{175}$$

$$w_Q(h) = \mathbb{P}\left[\chi^2\left(d, \left\|\frac{m - \mu}{s}\right\|^2\right) \leq -2s^2 \ln h + C\right] \tag{176}$$

$$C = s^2\left(2 \ln Z - d \ln(2\pi s^2)\right). \tag{177}$$

Unfortunately, in this case, it is unlikely that we can solve for the stretch function analytically, so in our experiments, we solved for it numerically using Equation (2).

H Experimental Details

Comparing Algorithm 3 versus Global-bound A* coding: We use a setup similar to the one used by Theis & Yosri (2022). Concretely, we assume the following model for correlated random variables $\mathbf{x}, \mu$:

$$P_\mu = \mathcal{N}(0, 1) \tag{178}$$

$$P_{\mathbf{x}|\mu} = \mathcal{N}(\mu, \sigma^2). \tag{179}$$

From this, we find that the marginal on $\mathbf{x}$ must be $P_{\mathbf{x}} = \mathcal{N}(0, \sigma^2 + 1)$. The mutual information is $I[\mathbf{x}; \mu] = \frac{1}{2} \log_2\left(1 + \sigma^2\right)$ bits, which is what we plot as $I[\mathbf{x}; \mu]$ in the top two panels in Figure 2.

For the bottom panel in Figure 2, we fixed a standard Gaussian prior $P = \mathcal{N}(0, 1)$, fixed $K = D_{\mathrm{KL}}[Q\|P]$ and linearly increased $R = D_\infty[Q\|P]$. To find a target that achieves the desired values for these given divergences, we set its mean and variances as

$$\sigma^2 = \exp\left(W(A \cdot \exp(B)) - B\right. \tag{180}$$

$$\mu = 2K - \sigma^2 + \ln \sigma^2 + 1 \tag{181}$$

$$A = 2 \ln R - 2K - 1 \tag{182}$$

$$B = 2 \ln R - 1, \tag{183}$$

where W is the principal branch of the Lambert W-function (Corless et al., 1996).

We can derive this formula by assuming we wish to find μ and σ^2 such that for fixed numbers K and R, and $q(x) = \mathcal{N}(x \mid \mu, \sigma^2), p(x) = \mathcal{N}(x \mid 0, 1)$. Then, we have that

$$D_{\mathrm{KL}}[q\|p] = K \quad \text{and} \quad \sup_{x \in \mathbb{R}} \left\{\frac{q(x)}{p(x)}\right\} = R. \tag{184}$$

We know that

$$\begin{aligned} K &= D_{\mathrm{KL}}[q\|p] = \frac{1}{2}\left[\mu^2 + \sigma^2 - \log\sigma^2 - 1\right] \\ \log R &= \log \sup_{x\in\mathbb{R}}\left\{\frac{q(x)}{p(x)}\right\} = \frac{\mu^2}{2(1-\sigma^2)} - \log\sigma. \end{aligned} \tag{185}$$

From these, we get that

$$\begin{aligned} \mu^2 &= 2K - \sigma^2 + \log\sigma^2 + 1 \\ \mu^2 &= 2(1-\sigma^2)(\log R + \log\sigma) \end{aligned} \tag{186}$$

Setting these equal to each other

$$\begin{aligned} 2K - \sigma^2 + \log\sigma^2 + 1 &= 2\log R + \log\sigma^2 - 2\sigma^2\log R - \sigma^2\log\sigma^2 \\ \sigma^2\log\sigma^2 - \sigma^2 + 2\sigma^2\log R &= 2\log R - 2K - 1 \\ \sigma^2\log\sigma^2 + \sigma^2(2\log R - 1) &= A \\ \sigma^2(\log\sigma^2 + B) &= A \\ \sigma^2\log(\sigma^2 e^B) &= A \\ e^B\sigma^2\log(\sigma^2 e^B) &= Ae^B \\ e^{\log(\sigma^2 e^B)}\log(\sigma^2 e^B) &= Ae^B \\ \log(\sigma^2 e^B) &= W(Ae^B) \\ \sigma^2 &= e^{W(Ae^B) - B}, \end{aligned} \tag{187}$$

where we made the substitutions $A = 2\log R - 2K - 1$ and $B = 2\log R - 1$.

I Rejection sampling index entropy lower bound

Assume that we have a pair of correlated random variables $\mathbf{x}, \mathbf{y} \sim P_{\mathbf{x},\mathbf{y}}$ and Alice and Bob wish to realize a channel simulation protocol using standard rejection sampling as given by, e.g. Algorithm 2. Thus, when Alice receives a source symbol $\mathbf{y} \sim P_{\mathbf{y}}$, she sets $Q = P_{\mathbf{x}|\mathbf{y}}$ as the target and $P = P_{\mathbf{x}}$ as the proposal for the rejection sampler. Let N denote the index of Alice's accepted sample, which is also the number of samples she needs to draw before her algorithm terminates. Since each acceptance decision is an independent Bernoulli trial in standard rejection sampling, N follows a geometric distribution whose mean equals the upper bound M used for the density ratio (Maddison, 2016). The lower bound on the optimal coding cost for N is given by its entropy

$$\mathbb{H}[N] = -(M-1)\log_2\left(1 - \frac{1}{M}\right) + \log_2 M \geq \log_2 M, \tag{188}$$

where the inequality follows by omitting the first term, which is guaranteed to be positive since $x \mapsto -x\log_2 x$ is positive on $(0,1)$. Hence, by using the optimal upper bound on the density ratio $M^* = \exp_2(D_\infty[Q\|P])$ and plugging it into the formula above, we find that

$$D_\infty[Q\|P] \leq \mathbb{H}[N]. \tag{189}$$

Now, taking expectation over $\mathbf{y}$, we find

$$\mathbb{E}_{\mathbf{y}\sim P_{\mathbf{y}}}\left[D_\infty[P_{\mathbf{x}|\mathbf{y}}\|P_{\mathbf{x}}]\right] \leq \mathbb{H}[N]. \tag{190}$$