
Fast Relative Entropy Coding with A* coding

Gergely Flamich*¹ Stratis Markou*¹ José Miguel Hernández-Lobato^{1 2 3}

Abstract

Relative entropy coding (REC) algorithms encode a sample from a target distribution Q using a proposal distribution P , such that the expected codelength is $\mathcal{O}(D_{\text{KL}}[Q\|P])$. REC can be seamlessly integrated with existing learned compression models since, unlike entropy coding, it does not assume discrete Q or P , and does not require quantisation. However, general REC algorithms require an intractable $\Omega(e^{D_{\text{KL}}[Q\|P]})$ runtime. We introduce AS* and AD* coding, two REC algorithms based on A* sampling. We prove that, for continuous distributions over \mathbb{R} , if the density ratio is unimodal, AS* has $\mathcal{O}(D_{\infty}[Q\|P])$ expected runtime, where $D_{\infty}[Q\|P]$ is the Rényi ∞ -divergence. We provide experimental evidence that AD* also has $\mathcal{O}(D_{\infty}[Q\|P])$ expected runtime. We prove that AS* and AD* achieve an expected codelength of $\mathcal{O}(D_{\text{KL}}[Q\|P])$. Further, we introduce DAD*, an approximate algorithm based on AD* which retains its favourable runtime and has bias similar to that of alternative methods. Focusing on VAEs, we propose the IsoKL VAE (IKVAE), which can be used with DAD* to further improve compression efficiency. We evaluate A* coding with (IK)VAEs on MNIST, showing that it can losslessly compress images near the theoretically optimal limit.

1. Introduction

In recent years, there has been significant progress in compression using machine learning, an approach known as learned compression. Most of the prominent learned compression methods, including the state-of-the-art in both lossless (Townsend et al., 2019; Hoogeboom et al., 2019; Zhang

et al., 2021) and lossy compression (Ballé et al., 2017), perform non-linear transform coding (Ballé et al., 2020).

In transform coding, a datum is first mapped to a latent representation and encoded with entropy coding. Entropy coding assumes that the latent representation and the coding distribution are discrete, which requires a non-differentiable quantization step. Since gradient-based optimization requires derivatives, most state-of-the-art methods use a continuous approximation to quantization during training and switch to hard quantization only during compression time (Ballé et al., 2017). This mismatch has been argued to be harmful towards the compression efficiency of these methods (Havasi et al., 2018; Flamich et al., 2020; Theis & Agustsson, 2021).

Relative entropy coding (REC; Flamich et al., 2020) is a recently proposed alternative, which does not require quantization and avoids this mismatch. A REC algorithm uses samples from a proposal distribution P to produce a random code representing a sample from a target distribution Q , with expected length of approximately $D_{\text{KL}}[Q\|P]_2$, where the subscript specifies that the KL is measured in bits rather than nats. General-purpose REC algorithms place no restrictions on Q and P beyond that $D_{\text{KL}}[Q\|P]_2$ be finite, so they can be applied even when Q and P are continuous. Thus REC can be naturally applied to perform compression with generative models trained via gradient descent, for applications including but not limited to: (1) data compression with variational autoencoders (VAE; Kingma & Welling, 2014), where Q corresponds to a variational posterior over latent variables and P to a prior over these latent variables; (2) model compression (Havasi et al., 2018), where Q corresponds to an approximate posterior over parameters (Blundell et al., 2015), and P to a prior over those parameters.

However, REC algorithms that make no further assumptions on Q or P require $\Omega(2^{D_{\text{KL}}[Q\|P]_2})$ steps to terminate in expectation (Agustsson & Theis, 2020), which is a severe limitation in practice. Thus, in order to make progress, it is necessary to impose additional assumptions on Q and P . Universal quantisation (Ziv, 1985) can be regarded as a REC algorithm that achieves $\mathcal{O}(D_{\text{KL}}[Q\|P]_2)$ runtime and has been demonstrated to work well with the state-of-the-art VAE-based learned compression methods (Agustsson & Theis, 2020). However, it places heavy limitations on Q and P , which might be overly restrictive in many cases.

*Equal contribution. ¹Department of Engineering, University of Cambridge, Cambridge, UK ²Microsoft Research, Cambridge, UK ³Alan Turing Institute, London, UK. Correspondence to: Gergely Flamich <gf332@cam.ac.uk>, Stratis Markou <em626@cam.ac.uk>.

In this work, we introduce AS* and AD* coding, two closely related REC algorithms based on A* sampling (Maddison et al., 2014), which achieve significantly faster runtimes than existing alternatives. For Q and P over \mathbb{R} , and without further assumptions, we show that the expected codelength achieved by AS* and AD* is bounded by

$$\lambda D_{\text{KL}}[Q\|P]_2 + \lambda \log_2(D_{\text{KL}}[Q\|P]_2 + 1) + \mathcal{O}(1), \quad (1)$$

where $\lambda \approx 2.41$ and $\lambda = 1$ for AS* and AD* respectively. With the additional assumption that Q and P are continuous with unimodal density ratio, we show that the expected runtime of AS* is $\mathcal{O}(D_\infty[Q\|P])$, where $D_\infty[Q\|P] = \log \sup_{x \in \mathcal{X}} \frac{dQ}{dP}(x)$ is the Rényi ∞ -divergence. While we do not prove an analogous bound for the runtime of AD*, we conduct extensive experiments on different Q, P pairs, and observe that the runtime of AD* is also $\mathcal{O}(D_\infty[Q\|P])$. Thus AS* and AD* significantly improve upon the runtime of existing REC methods, without requiring as severe assumptions as universal quantization. While AS* and AD* require unimodality, this assumption is satisfied by many models in learnt compression, such as most VAEs.

In addition, a practical limitation of REC algorithms is that, since the codelength of a sample is random, additional bits must be used to communicate the codelength itself to ensure the message is decodable. This additional code, which corresponds to the second and third terms of eq. (1), accounts for a large portion of the overall codelength and grows linearly with the number of dimensions. To remedy this, we consider approximate REC algorithms, in which the codelength is a parameter, which is set prior to coding. This allows us to form blocks of variables which are coded using the same codelength. Thus, the additional cost must be paid only once per block, rather than once per variable, thereby greatly reducing this overhead codelength.

To this end, we first introduce an approximate variant of AD* coding which, similarly to existing approximate REC algorithms, has a tunable codelength and a provably low bias. Unlike existing methods however, it retains the favourable runtime of AD* coding. Second, we propose to parameterize latent variable distributions by explicitly specifying $D_{\text{KL}}[Q\|P]$. For example, instead of parameterizing a Gaussian Q using a mean and variance, we can specify its mean and $D_{\text{KL}}[Q\|P]$, from which the variance is uniquely determined. This allows us to construct blocks of latent variables with tied KL divergences, which can be coded with the same codelength. This codelength must be communicated once per block instead of once per variable. We consider VAE models using this parameterization, which we refer to as isoKL VAEs (IKVAEs). We present experiments on lossless image compression on MNIST which demonstrate that the performance of IKVAEs is comparable to that of VAEs, while reducing the codelength overhead.

Our contributions can be summarised as follows:

- We introduce AS* and AD* coding, two REC algorithms based on A* sampling, for coding samples from one-dimensional distributions.
- We prove that, the expected codelength of AS* and AD* is $\mathcal{O}(D_{\text{KL}}[Q\|P])$.
- We prove that if dQ/dP is bounded and unimodal, AS* achieves $\mathcal{O}(D_\infty[Q\|P])$ runtime. We present empirical evidence that the runtime of AD* is also linear in D_∞ . Therefore AS* and AD* significantly improve over the exponential runtime of existing REC algorithms. A direct consequence of the above is that A* sampling with unimodal dQ/dP also has $\mathcal{O}(D_\infty[Q\|P])$ runtime.
- We introduce an approximate variant of AD* and bound its bias. Similar to existing ones, this algorithm can code low-bias samples using fixed codelengths, but retains the favourable runtime of AD*.
- We introduce a novel modification for VAEs, in which the KL divergences across either all, or some of, the latent dimensions are tied. This modification, which we refer to as the isoKL VAE (IKVAE), can be used with any fixed-codelength approximate REC algorithm, such as our own, to greatly reduce overhead codes.
- We demonstrate the favourable performance of AS* and AD* on toy problems, comparing it with that of alternative methods. Lastly, we apply our approximate AD* algorithm to VAE and IKVAE models on image data, showing that it can losslessly compress images near the theoretically optimal ELBO.

2. Background

Relative Entropy Coding: The central problem which we tackle in this work is the REC problem, defined as follows.

Definition 1 (REC problem and algorithm). *Let Q be a target and P be a proposal distribution, with $D_{\text{KL}}[Q\|P] < \infty$, and let $S = (s_1, s_2, \dots)$ be an infinite sequence of publicly available independent fair coin tosses. Relative entropy coding (REC) is the problem of producing a uniquely decodable code C representing a sample from Q given S , such that the codelength $|C|$ satisfies*

$$\mathbb{E}[|C|] = \mathcal{O}(D_{\text{KL}}[Q\|P]), \quad (2)$$

An algorithm which solves this problem is a REC algorithm.

In practice, S is implemented by using a pseudo-random number generator (PRNG) with a publicly available seed. Crucially, REC applies to both discrete and continuous distributions, and can be integrated into learned compression pipelines, without requiring quantization. Several existing algorithms solve the REC problem without further assumptions on Q or P , however, they are impractically slow.

Poisson Functional Representation: Li & El Gamal (2018) introduced a REC algorithm for general Q and P , here referred to as Poisson functional representation (PFR) coding. Li & El Gamal showed that if T_i are the ordered arrival times of a homogeneous Poisson process on \mathbb{R}^+ (Kingman, 1992), and $X_i \sim P$, then

$$\arg \min_{i \in \mathbb{N}} \left\{ T_i \cdot \frac{dP}{dQ}(X_i) \right\} \sim Q. \quad (3)$$

Further, Li & El Gamal showed that a sample may be represented by coding the index which minimises eq. (3), and bounded the expected codelength of PFR by

$$K \leq \mathbb{E}[|C|] \leq K + \log_2(K + 1) + \mathcal{O}(1), \quad (4)$$

where $K = D_{\text{KL}}[Q||P]_2$. We note that PFR converts REC into a search problem, just like the A* sampling algorithm (Maddison et al., 2014) converts sampling into a search problem. In fact, it can be shown that the minimization in eq. (3), is equivalent to a variant of A* sampling, called Global Bound A* sampling. In particular

$$\begin{aligned} \arg \min_{i \in \mathbb{N}} \left\{ T_i \frac{dP}{dQ}(X_i) \right\} &= \arg \max_{i \in \mathbb{N}} \{-\log T_i + \log r(X_i)\} \\ &\stackrel{d}{=} \arg \max_{i \in \mathbb{N}} \{G_i + \log r(X_i)\} \end{aligned} \quad (5)$$

where $r = dQ/dP$, and G_i is sampled according to

$$G_i \sim \text{TG}(0, G_{i-1}), \quad (6)$$

where $\text{TG}(\mu, \kappa)$ denotes the Gumbel distribution with mean μ and unit scale, truncated to the interval $(-\infty, \kappa]$, and defining $G_0 \stackrel{\text{def}}{=} \infty$. The maximisation in eq. (5) is identical to the Global Bound A* sampling algorithm (see Appendix of Maddison et al.). Thus, Global Bound A* sampling and PFR are identical, with the exception that the former works in the negative log-space of the latter (eq. 5).

Unfortunately, the runtime T of PFR is so large that it renders the algorithm intractable in practice. In particular, the runtime of Global Bound A*, and thus also of PFR, can be shown (see Appendix in Maddison et al.) to be equal to

$$\mathbb{E}[T] = \exp(D_\infty[Q||P]) \geq \exp(D_{\text{KL}}[Q||P]). \quad (7)$$

where $D_\infty[Q||P] = \log \sup_{x \in \mathcal{X}} r(x)$. This bound is perhaps unsurprising, considering a more general result by Agustsson & Theis (2020), who proved that without further assumptions on Q and P , the expected runtime of any REC algorithm is $\Omega(2^{D_{\text{KL}}[Q||P]_2})$.

Additional assumptions: In order to develop a REC algorithm that is fast enough to be practical, we must make further assumptions about the target and proposal distributions. Focusing on continuous distributions over \mathbb{R} , we will

show that an assumption which enables fast runtimes is the unimodality of r . While somewhat restrictive, this assumption is satisfied by virtually all models used in learned compression. We will show that A* sampling can be modified to solve the REC problem, achieving $\mathcal{O}(D_\infty[Q||P])$ runtime whenever r is bounded and unimodal.

Henceforth, we will assume that Q and P are continuous distributions on \mathbb{R} with densities q and p . However, we note that the methods we present in this work can be generalised to arbitrary measure spaces equipped with a total ordering over their elements, using the Radon-Nikodym derivative in place of the density ratio (Grimmett & Stirzaker, 2001).

A* sampling: The A* sampling algorithm (Maddison et al., 2014) is an extension of the Gumbel-max trick (Papandreou & Yuille, 2011) to arbitrary probability spaces. A* sampling is a branch-and-bound algorithm (Land & Doig, 1960), which converts the problem of sampling from Q into the maximization in eq. (5). A* sampling builds a binary search tree, where each node n is associated with a triplet (B_n, X_n, G_n) , where: (1) $B_n \subseteq \Omega$ is a subset of the sample space; (2) $X_n \sim P|_{B_n}$ is a sample distributed according to the restriction of P to B_n ; (3) $G_n \sim \text{TG}(\log P(B_n), G_{\text{par}(n)})$ is a truncated Gumbel sample, where $\text{par}(n)$ is the parent of n . At each step, A* sampling expands the children of n , partitioning B_n into two disjoint subsets $L_n \cup R_n = B_n$, using some rule which we refer to as `partition`. A* sampling then constructs the children’s triplets following the definitions above. It then bounds the objective from eq. (5) on each of the children, and utilises the bounds to narrow the search and quickly locate the maximum. Setting `partition` to be the degenerate splitting

$$\text{partition}_{\text{GBA}^*}(B, X) \stackrel{\text{def}}{=} (\emptyset, B) \quad (8)$$

yields the Global Bound A* algorithm. Equation (8) hints at why Global Bound A*, and by extension PFR, have large runtimes. This `partition` function does not refine the search as the algorithm progresses, maintaining the same fixed global bound throughout a run. In this work we consider using two different `partition` functions, yielding the AS* and AD* coding algorithms. When applied to unimodal r , these partitioning schemes enable the algorithm to quickly refine its search and achieve a fast runtime.

Approximate REC algorithms: Other lines of work have introduced alternative algorithms, such as Minimal Random Coding (MRC; Havasi et al., 2018) and Ordered Random Coding (ORC; Theis & Yosri, 2022), which produce a code representing an approximate, instead of an exact, sample from Q . Because these algorithms produce biased samples, strictly speaking they do not satisfy the requirements of definition 1, so we refer to them as approximate REC algorithms. Both MRC and ORC accept the codelength $|C|$ as

an input parameter. Increasing $|C|$ reduces sample bias, but increases compression cost and runtime. Unfortunately, the runtime required to reduce the bias sufficiently in order to make the samples useful in practice scales as $\mathcal{O}(2^{D_{\text{KL}}[Q\|P]_2})$, making MRC and ORC as expensive as PFR.

However, one benefit of a tunable codelength is that, when communicating multiple samples, the overhead code corresponding to the second and third terms in eq. (4), can be significantly reduced. By grouping several random variables into blocks, and coding all samples of a block with the same codelength, we only need to communicate a codelength per block, as opposed to one codelength per variable. This procedure reduces the codelength overhead by a factor equal to the number of variables in each block.

3. A* Coding

A* sampling returns a node n with associated triplet (B_n, X_n, G_n) , where X_n is an exact sample from Q (Maddison et al., 2014). Therefore, the only addition we need to make to A* sampling to turn it into a REC algorithm, is a way to represent n using a uniquely decodable code C . Given such a C , we can decode the sample X_n by determining the B_n corresponding to n , and then sampling $X_n \sim P|_{B_n}$ given the public source of randomness S .

Since A* sampling may return any node in its binary search tree, we propose to use heap indexing, also known as the *ahmentafel* or *Eytzinger ordering* to code nodes. Let the parent of the root node be `nil`. Then, the heap index of a node n is

$$H_n \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } \text{par}(n) = \text{nil} \\ 2H_{\text{par}(n)} & \text{if } n \text{ is left child of } \text{par}(n) \\ 2H_{\text{par}(n)} + 1 & \text{if } n \text{ is right child of } \text{par}(n). \end{cases} \quad (9)$$

Let D_n denote the depth of node n in the binary tree, where $D_{\text{root}} = 1$. We can see that $D_n = \lfloor \log_2 H_n \rfloor + 1$. Thus, assuming that D_n is known, H_n can be encoded in D_n bits. Therefore, we can modify A* sampling to return the heap index of the optimal node, from which n can be decoded. This yields Algorithm 1, which we refer to as A* coding. Maddison et al. (2014) show that A* sampling is correct regardless of the choice of `partition`. The following theorem shows that under mild assumptions on `partition`, the expected codelength of A* coding is $\mathcal{O}(D_{\text{KL}}[Q\|P])$.

Theorem 1 (Expected codelength of A* coding). *Let Q and P be the target and proposal distributions passed to A* coding (Algorithm 1), respectively. Assume that `partition` satisfies the following property: there exists $\epsilon \in [1/2, 1)$ such that for any node n we have*

$$\mathbb{E}[P(B_n)] \leq \epsilon^{D_n}, \quad (10)$$

where the expectation is taken over the joint distribution of the samples associated with the ancestor nodes of n . Let k

be the node returned by A* coding. Then, we have

$$\mathbb{E}[D_k] \leq -\frac{1}{\log \epsilon} [D_{\text{KL}}[Q\|P] + e^{-1} + \log 2]. \quad (11)$$

In particular, when $\epsilon = 1/2$,

$$\mathbb{E}[D_k] \leq D_{\text{KL}}[Q\|P]_2 + e^{-1} \log_2 e + 1. \quad (12)$$

Proof. See Appendix A for proof. \square

Motivated by the results of Theorem 1, we examine two variants of A* coding based on particular choices for `partition`, which yield AS* and AD* coding.

AS* coding: For node n with triplet (B_n, X_n, G_n) , where $B_n = (\alpha, \beta)$, we define `partition` as

$$\text{partition}_{\text{AS}^*}(B_n, X_n) \stackrel{\text{def}}{=} (\alpha, X_n), (X_n, \beta). \quad (13)$$

In this case, the following result holds.

Lemma 1. *Let P be the proposal distribution passed to AS* coding and let `partition` be as defined in eq. (13). Then the condition in eq. (10) is satisfied with $\epsilon = 3/4$, that is*

$$\mathbb{E}[P(B_n)] \leq (3/4)^{D_n}. \quad (14)$$

Proof. See Appendix B for proof. \square

Hence, the codelength of AS* sampling is bounded by

$$\lambda D_{\text{KL}}[Q\|P]_2 + \lambda \log(D_{\text{KL}}[Q\|P] + 1) + \mathcal{O}(1), \quad (15)$$

where $\lambda = -\log 2 / \log(3/4) \approx 2.41$. Further, we have the following result.

Theorem 2 (Expected runtime of AS* coding). *Let Q and P be the target and proposal distributions passed to AS* coding (Algorithm 1), and assume $r = dQ/dP$ is quasi-concave. Let T be the number of steps AS* takes before it terminates. Then, we have*

$$\mathbb{E}[T] = \mathcal{O}(D_{\infty}[Q\|P]) + \mathcal{O}(1). \quad (16)$$

Proof. See Appendix C for proof. \square

Theorem 2 identifies a general class of target-proposal pairs where REC and A* sampling can be performed much faster than the $\Omega(\exp(D_{\text{KL}}[Q\|P]))$ and $\mathcal{O}(\exp(D_{\infty}[Q\|P]))$ bounds shown by Agustsson & Theis (2020) and Maddison et al. (2014); Maddison (2016), respectively.

AD* coding: Unfortunately, the upper bound on the codelength of AS* coding in eq. (15) is not tight enough to be optimal. However, Theorem 1 suggests this may be addressed by choosing a `partition` function for which

$\epsilon = 1/2$. One such partition function is dyadic partitioning, which splits $B_n = (\alpha, \beta)$ as

$$\text{partition}_{AD^*}(B_n, X_n) \stackrel{\text{def}}{=} (\alpha, \gamma), (\gamma, \beta), \quad (17)$$

where γ is chosen such that $P((\alpha, \gamma)) = P((\gamma, \beta))$, which is always possible for continuous P . We refer to this partitioning as dyadic, and the corresponding algorithm AD* coding, because for every node n in the search tree with $B_n = (\alpha, \beta)$, we have that $(F_P(\alpha), F_P(\beta))$ forms a dyadic subinterval of $(0, 1)$, where F_P is the CDF of P .

Algorithm 1: A* coding. **Blue parts** show modifications of A* sampling (Maddison et al., 2014).

Input : Target Q , proposal P , bounding function M , maximum search depth D_{max} .
 $(LB, X^*, k) \leftarrow (-\infty, \text{null}, 1)$
 $\Pi \leftarrow \text{PriorityQueue}$
 $(D_1, H_1) \leftarrow (1, 1)$
 $G_1 \sim \text{TG}(0, \infty)$
 $X_1 \sim P$
 $M_1 \leftarrow M(\mathbb{R})$
 $\Pi.\text{push}(1, G_1 + M_1)$

while ! $\Pi.\text{empty}()$ **and**
 $LB < \Pi.\text{topPriority}()$ **do**
 $n \leftarrow \Pi.\text{popHighest}()$
 $LB_n \leftarrow G_n + (dQ/dP)(X_n)$
 if $LB < LB_n$ **then**
 $(LB, X^*) \leftarrow (LB_n, X_n)$
 $H^* \leftarrow H_n$
 end
 if $D_n \leq D_{max}$ **then**
 $L, R \leftarrow \text{partition}(B_n, X_n)$
 for $B \in \{L, R\}$ **do**
 $(k, B_k) \leftarrow (k + 1, B)$
 $D_k \leftarrow D_n + 1$
 $H_k \leftarrow \begin{cases} 2H_n & \text{if } B = L \\ 2H_n + 1 & \text{if } B = R \end{cases}$
 $G_k \sim \text{TG}(\log P(B_k), G_n)$
 $X_k \sim P|_{B_k}$
 if $LB < G_k + M_n$ **then**
 $M_k \leftarrow M(B_k)$
 if $LB < G_k + M_k$ **then**
 $\Pi.\text{push}(k, G_k + M_k)$
 end
 end
 end
 end
end
return (X^*, H^*)

We conjecture, that the expected runtime of AD* is $\mathcal{O}(D_\infty[Q||P])$ and in Section 5 we provide thorough ex-

perimental evidence for this.

Depth-limited A* coding: There is a natural way to set up an approximate REC algorithm based on A*, which takes $|C|$ as an input parameter. Specifically, we can limit the maximal depth D_{max} to which alg. 1 is allowed to search. The number of nodes in a complete binary tree of depth D is $2^D - 1$, so setting $D_{max} = |C|$ ensures that each node can be encoded using a heap index with $|C|$ bits. By limiting the search depth, A* coding returns the optimal node up to depth D_{max} instead of the global optimum. However, if we set D_{max} large enough, then depth-limited algorithm should also be able to find the global optimum. This intuition is made precise in the following lemma.

Lemma 2. *Let Q and P be the target and proposal distributions passed to A* coding (Algorithm 1). Let H^* be the heap index returned by unrestricted A* coding and H_d^* be the index returned by its depth-limited version with $D_{max} = d$. Then, conditioned on the public random sequence S , we have $H_d^* \leq H^*$. Further, there exists $D \in \mathbb{N}$ such that for all $d > D$ we have $H_d^* = H^*$.*

Proof. See Appendix E for proof. \square

If the depth of the global optimum is larger than D_{max} , then depth-limited A* coding will not return an exact sample. As we reduce D_{max} , we force the algorithm to return increasingly sub-optimal solutions, which correspond to more biased samples. It is therefore important to quantify the trade-off between D_{max} and sample bias. Theorem 3 bounds the sample bias of depth-limited AD* coding, which we refer to as DAD*, as a function of D_{max} . This result is similar to existing bounds for MRC and ORC.

Theorem 3 (Biasedness of DAD* coding). *Let Q and P be the target and proposal distributions passed to DAD* coding (Algorithm 1). Let*

$$K \stackrel{\text{def}}{=} [D_{\text{KL}}[Q||P]_2], \quad D \stackrel{\text{def}}{=} K + t, \quad N \stackrel{\text{def}}{=} 2^D \quad (18)$$

where t is a non-negative integer, $r = dQ/dP$ and $Y \sim Q$. Let f be a measurable function and define

$$\|f\|_Q \stackrel{\text{def}}{=} \sqrt{\mathbb{E}_Q[f^2]}. \quad (19)$$

Let \tilde{Q}_D the distribution of the approximate sample returned by DAD* coding with depth-limit D . Define

$$\delta \stackrel{\text{def}}{=} \left(2^{-t/4} \sqrt{1 + \frac{1}{N}} + 2\sqrt{\mathbb{P}[\log_2 r(Y) > K + t/2]} \right)^{1/2}$$

Then,

$$\mathbb{P} \left[\left| \mathbb{E}_{\tilde{Q}_D}[f] - \mathbb{E}_Q[f] \right| \geq \frac{2\|f\|_Q\delta}{1-\delta} \right] \leq 2\delta. \quad (20)$$

Proof. See Appendix D for proof. \square

Theorem 3 says that the bit budget for DAD* should be approximately $D_{\text{KL}}[Q||P]_2$ in order to obtain reasonably low-bias samples. As we increase the budget beyond this point, we observe that in practice δ , and by extension the bias, decay quickly. In particular, as $t \rightarrow \infty$, $\delta \rightarrow 0$, and we recover exact AD*. We note that it is also possible to depth-limit other variants of A* coding, such as AS*. However, for any fixed D_{max} , the sample bias of these variants will be larger than the bias of DAD*. This is because, as shown in theorem 1, if we use a different partition with $\epsilon > 1/2$, A* coding will need to search deeper down the tree to find the global optimum. AD* achieves the lowest possible average depth out of all variants of A* coding, because it has $\epsilon = 1/2$. Equivalently, AD* can achieve the same sample quality with a lower D_{max} than any other variant of A* coding. In practice, we observed that depth limited AS* gives significantly more biased samples than AD*, in line with the above reasoning, so we did not further pursue any depth-limited variants other than DAD*.

Runtime of DAD* coding: Based on our conjecture for exact AD* and the result of Lemma 2, we conjecture that DAD* runs in $\mathcal{O}(D_\infty[Q||P])$ time. We provide experimental evidence for this in Section 5.

Using all available codewords: When $D_{\text{max}} = D$, the number of nodes in the binary search tree of DAD* is $2^D - 1$, which is one fewer than the number of items we can encode in D bits. In particular, the codeword 0 is never used, since heap indexing starts at 1. We can make AD* slightly more efficient by drawing 2 samples and arrival times at the root node instead of a single one. We found that this has a significant effect on the bias for small D , and becomes negligible for large D . In Section 5, we perform our experiments with this modified version of DAD*.

Tying codelengths: Since the codelength is a parameter of DAD*, we can code samples from different variables using the same codelength, grouping them in a block and passing $D_{\text{max}} = |C|$ to Algorithm 1 for each variable in the block. Since the variables have the same codelength, we only need to communicate this codelength once per block.

4. IsoKL layers and VAEs

Tying KL divergences: Although DAD* can be used to tie together the codelengths of different samples, Theorem 3 suggests that the search depth $|C|$ used in DAD* affects the sample bias. In order to obtain low-bias samples, we must set $|C| \geq D_{\text{KL}}[Q||P]$. If we group variables with different KL divergences and code them using the same $|C|$, one of the two following unwanted effects might occur: (1) if $D_{\text{KL}}[Q||P] \gg |C|$ for a variable, then the corresponding sample will be highly biased; (2) if $D_{\text{KL}}[Q||P] \ll |C|$ for a variable, then an excessive codelength is being used to code its sample, which is inefficient. To avoid these cases,

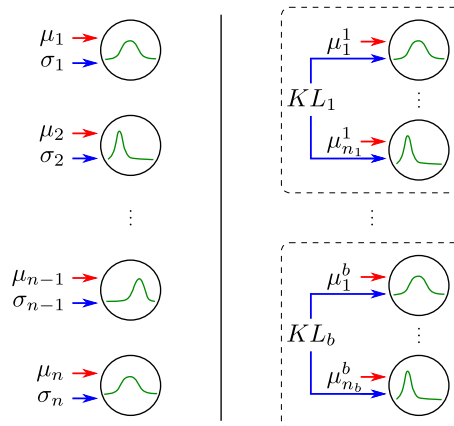


Figure 1. Two different ways of parameterizing a Gaussian variational posterior Q . **Left:** The usual mean-variance parameterization. **Right:** IsoKL layer using our proposed mean-KL parameterization where the latent dimensions are divided into b blocks, and the KLs are shared within each block.

it would be useful if $|C| \approx D_{\text{KL}}[Q||P]$ for all variables in a block. We can achieve this by constraining the KL divergences of all variables in a block to be equal.

Suppose Q and P are diagonal Gaussians over (X_1, \dots, X_N) with means (μ_1, \dots, μ_N) and (ν_1, \dots, ν_N) , and variances $(\sigma_1^2, \dots, \sigma_N^2)$ and $(\rho_1^2, \dots, \rho_N^2)$, respectively. We can parameterize Q such that

$$D_{\text{KL}}[\mathcal{N}(\mu_n, \sigma_n^2) || \mathcal{N}(\nu_n, \rho_n^2)] = \kappa \text{ and } \sigma_n < \rho_n. \quad (21)$$

for each $n = 1, \dots, N$, by setting

$$|\mu_n - \nu_n| < \rho_n \sqrt{2\kappa} \quad (22)$$

$$\sigma_n^2 = -\rho_n^2 W(-\exp(\Delta_n^2 - 2\kappa - 1)), \quad (23)$$

where $\Delta_n = (\mu_n - \nu_n)/\rho_n$ and W is the principal branch of the Lambert W function (Lambert, 1758). Note that the condition that $\sigma_n < \rho_n$ will ensure that $D_\infty[Q||P] < \infty$. While the W is not elementary, it can be computed numerically in a fast and efficient way. We refer to this as an IsoKL Gaussian layer (see Figure 1), and call a VAE model using such a Q an IsoKL VAE (IKVAE). Although we focus on Gaussians, this approach can be extended to other distributions. See Appendix F for implementation details and mathematical details on deriving the necessary quantities for IsoKL layers.

5. Experiments

Experiments for exact REC: We conducted experiments using PFR, AS* and AD* coding to perform REC on Gaussian, uniform and disjoint mixture of uniform distributions, where we systematically vary the problem parameters.

Figure 2 shows measurements for PFR, AS* and AD* coding. We report the number of steps executed by each algorithm rather than wall-clock time, as the former is proportional to the runtime and is unaffected by specific differences

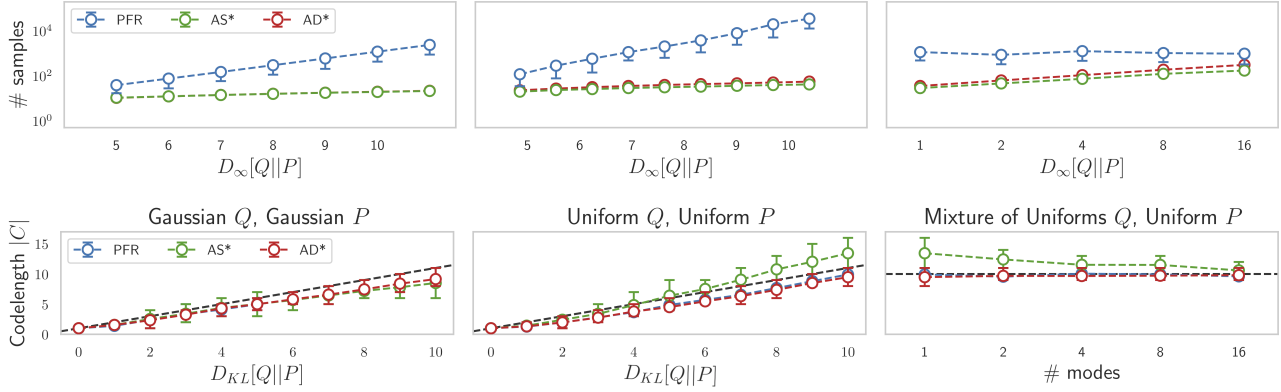


Figure 2. Number of steps (top) and codelength (bottom) for PFR, AS* and AD* coding. Reported codelengths do not include the overhead terms. Circles show mean values, and error bars show first and third quartiles. AS* and AD* are significantly quicker than PFR.

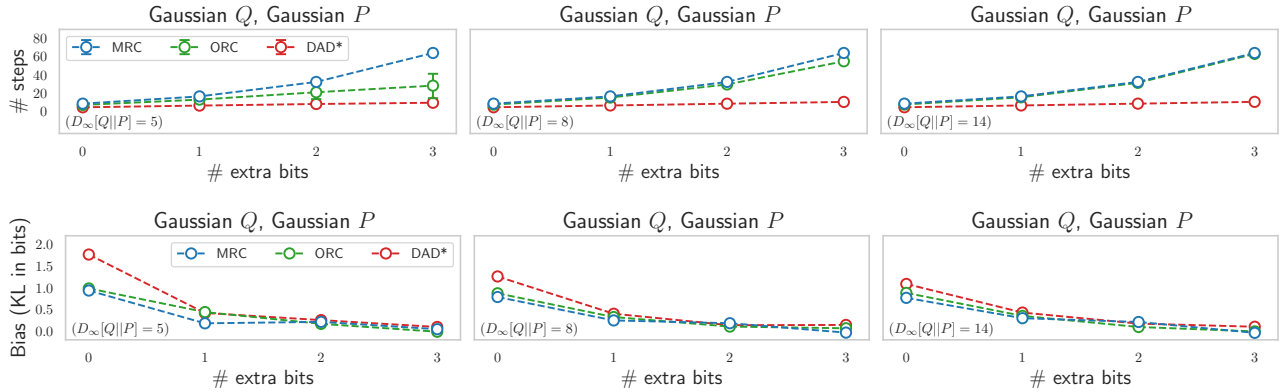


Figure 3. Number of steps (top) and bias (bottom) for MRC, ORC and DAD* coding. The bias is measured as the KL divergence from the empirical distribution of samples, to the true target (discussion in text). DAD* has similar bias to MRC and ORC, but is much faster.

in implementation. We also report the codelength, in bits, excluding the additional logarithmic and constant overhead. First, we observe that the number of steps taken by PFR scales exponentially with $D_\infty[Q||P]$, as expected. This scaling renders PFR computationally intractable for practical problems. By contrast, the number of steps taken by AS* and AD* coding increases linearly with $D_\infty[Q||P]$.

Second, we observe that the number of modes has an effect on the runtime of both AS* and AD* coding. In the top right of Figure 2 we have set Q to a mixture of uniforms with disjoint support, P to a uniform, and fixed $D_\infty[Q||P]$ while varying the number of modes in Q . For a small number of modes, AS* and AD* are both significantly faster than PFR. As the number of modes increases, so does the number of steps executed by AS* and AD*. This trend is expected, because for unimodal Q , A* coding can quickly bound the objective for large regions of the input space which have low dQ/dP throughout. As dQ/dP becomes increasingly multimodal, the bounding function M becomes large in increasingly many disconnected regions of the search space. Thus both AS* and AD* must drill down to and search increasingly many of these regions, before producing a sample, which requires a larger number of steps. By contrast, PFR is unaffected by the number of modes in dQ/dP , since

it is equivalent to Global Bound A*, which retains the entire input space in a single active search branch. These results suggest that the unimodality of dQ/dP is a key attribute for enabling fast coding with AS* and AD*.

Third, we observe that the codelengths of PFR, AS* and AD* scale linearly with $D_{\text{KL}}[Q||P]$, as expected. However, in some cases AS* produces a larger mean codelength than PFR and AD*. This can be explained by the fact that the dominant term in the codelength bounds of the three algorithms is $\lambda D_{\text{KL}}[Q||P]$, where PFR and AD* have $\lambda = 1$, while AS* has $\lambda \approx 2.41$. AS* can produce larger $|C|$ than AD* because, as Lemma 1 states, the expected rate of shrinkage of its search region is slower than AD*. Therefore, in order to refine its search by the same amount, AS* needs a greater number of steps, leading to a larger expected $|C|$.

Experiments for approximate REC: We conducted experiments using MRC, ORC and DAD* coding to perform approximate REC with Gaussian Q and P , varying the bit budget that is allowed to the coders, in addition to the baseline budget of $D_{\text{KL}}[Q||P]_2$ bits. Figure 3 shows the effect of the additional bit budget on the number of steps and the sample bias for three problems with different $D_\infty[Q||P]$. We quantify the sample bias as the KL divergence, $D_{\text{KL}}[\hat{Q}||Q]_2$, from the empirical distribution \hat{Q} of the approximate sam-

ples, to the true Q . In each case, we draw 100 samples and follow the method of Pérez-Cruz (2008) to estimate $D_{\text{KL}}[\hat{Q}||Q]_2$. We observe that in all three cases (fig. 2 bottom), while there is a slight difference in the level of bias when no extra bits are allowed, the difference in bias becomes negligible when one or more extra bits are added. However, DAD* achieves this bias far faster than MRC and ORC (fig. 2 top), making it a far more tractable method.

	# LATENT	NEG. ELBO	AD*	DAD*
VAE	20	1.43 ± 0.01	1.53 ± 0.01	—
	50	1.40 ± 0.01	1.66 ± 0.01	—
IKVAE	20	1.44 ± 0.01	1.55 ± 0.01	1.51 ± 0.01
	50	1.44 ± 0.01	1.69 ± 0.01	1.60 ± 0.01

Table 1. ELBO and lossless compression rates (bpp) on MNIST.

	# LATENT	AD*	DAD*
IKVAE	20	86.31 ± 0.01	5.00 ± 0.01
	50	195.20 ± 0.01	5.00 ± 0.01

Table 2. Overhead codelength (bits) on MNIST.

Image compression MNIST: We compared the performance of AD* and DAD* on image compression experiments on MNIST, using the feedforward VAE architecture of Townsend et al. (2018), with Gaussian Q and P . We also trained IKVAEs with the same architecture, using an IsoKL Gaussian Q . As Theorem 1 shows and as we discuss above, AS* has strictly worse expected codelength than AD* and hence we did not include it in our experiments. For DAD*, we set $\kappa = 2$ based on preliminary experiments. Table 1 shows the lossless compression rates of different model and coder combinations, from which we observe the following trends. First, the IKVAEs achieve similar ELBOs to the VAEs, suggesting that tying the KLs does not degrade model performance. Further, we observe that for the IKVAE architectures, using DAD* improves the compression rate over AD*. We do not provide results for DAD* applied to a standard VAE posterior, as it would yield a strictly worse performance than AD*. This is because the KLs in each latent dimension are different, and hence not only do we need to communicate the codelength for each dimension, but DAD* returns approximate samples from the target distribution, as opposed to AD*. This is because, as corroborated by table 2, DAD* significantly reduces codelength overheads, improving performance over AD*. We also note that in this task, the number of latent dimensions is relatively small (20 or 50) compared to the number of image pixels (784). Since the overhead costs increase with the number of latent variables, we expect the savings of DAD* to be more pronounced for larger IKVAEs. Overall, this experiment demonstrates that AD* can be effective for VAEs, while DAD* and IKVAEs further reduce coding overheads.

6. Related Work

Quantization-based approaches: Most state-of-the-art methods in both lossy and lossless compression are based on including quantization in the compression pipeline, and somehow circumventing its non-differentiability during training. Current widespread approaches in lossy compression use VAEs with a particular choice of latent distributions (Ballé et al., 2017; 2018). Instead of quantizing latent representations during training, these methods perturb the latents with uniform noise, a technique known as dithering. To perform compression, the methods switch back to hard quantization. Dithering is equivalent to using a uniform variational posterior distribution, and has been demonstrated to work well in practice (Ballé et al., 2020). However, this introduces a mismatch between the training and compression phases and it also constrains the design choices for new methods to use uniform distributions. A related variant is the work of Agustsson & Theis (2020), who propose to use universal quantization (Ziv, 1985) for compression. Universal quantisation can be regarded as a REC algorithm. While this method performs very well in the experiments of Agustsson & Theis (2020), it is limited to particular choice of distribution. Our work can be regarded as a step towards lifting the restrictions imposed by quantization based methods.

Bits-back coding: Townsend et al. (2018) introduced a practical way to combine bits-back coding (Hinton & Van Camp, 1993) and VAEs to perform lossless data compression. The method of Townsend et al. (2018) has later been applied to normalizing flows (Ho et al., 2019), which, together with discrete (van den Berg et al., 2020) and quantization based approaches (Zhang et al., 2021) represent the state-of-the-art in lossless image compression. However, bits-back coding is only applicable to perform lossless compression. Furthermore, it is only asymptotically efficient, meaning that it has a large constant codelength overhead that becomes insignificant as we compress larger batches of data. In contrast, our method is applicable to both lossy and lossless compression and can be used to perform one-shot compression.

REC and reverse channel coding: REC was first proposed by Havasi et al. (2018), who developed MRC for compressing Bayesian neural networks, and was later extended to data compression using VAEs by Flamich et al. (2020). Both of these works were inspired by reverse channel coding (Bennett et al., 2002). REC and reverse channel coding can be viewed as the worst-case and average-case approaches to the same problem. Concretely, REC requires that for fixed proposal P and public randomness S , the codelength bound in eq. (2) holds for any target Q . In contrast, reverse channel coding assumes a distribution over a family of possible targets and requires that eq. (2) holds in expectation.

The first general REC algorithm for discrete distributions was proposed by Harsha et al. (2007), however this method

has an impractically long runtime. Li & El Gamal (2018) developed PFR coding, an alternative based on Poisson processes, however this also is computationally impractical. Recently, Theis & Yosri (2022) proposed ORC, which combines some the benefits of MRC and PFR, but remains computationally impractical. These algorithms all share the limitation that their expected runtime is $\Omega(\exp(D_{\text{KL}}[Q\|P]))$.

Hybrid Coding: To improve the runtime of REC/RCC algorithms, Theis & Yosri (2022) proposed hybrid coding (HC). HC is applicable whenever the support of the target is compact, and can be combined with any of the existing REC/RCC algorithms to improve their runtime by a multiplicative factor. However, HC does not change the asymptotic complexity of the REC/RCC algorithm that it is combined with. We note that while Theis & Yosri (2022) used HC in conjunction with ORC, HC can equally well be combined with A* coding. Thus the speedup that HC provides is complementary to that of A* coding.

Efficient rejection sampling: While rejection sampling is applicable to any target Q and proposal P where $D_{\infty}[Q\|P] < \infty$ and dQ/dP can be evaluated, the basic algorithm has $\mathcal{O}(\exp(D_{\infty}[Q\|P]))$ expected runtime complexity (Maddison, 2016). Thus, a natural question is to ask under what assumptions it is possible to perform rejection sampling from a target Q using a proposal P efficiently.

Recently, Chewi et al. (2022) consider the problem of constructing an upper envelope for rejection sampling from discrete probability distributions. In particular, they study the time complexity of constructing the envelope as a function of the alphabet size. They show that shape constraints on the target distribution, such as monotonicity and log-concavity can be utilized to design algorithms whose runtime scales logarithmically in the alphabet size. Our work is complementary to theirs, as Theorem 2 provides an initial result showing that shape constraints can be leveraged to design more efficient sampling algorithms for continuous distributions as well.

7. Conclusion

Summary: In this work we proposed AS* coding and AD* coding, two algorithms based on A* sampling for performing REC with one-dimensional target and proposal distributions Q and P . We proved that the expected code-lengths of AS* and AD* are $\mathcal{O}(D_{\text{KL}}[Q\|P])$ and that, whenever dQ/dP is unimodal, the expected runtime of AS* is $\mathcal{O}(D_{\infty}[Q\|P])$. Experimental evidence suggests that the runtime of AD* is also $\mathcal{O}(D_{\infty}[Q\|P])$. This runtime significantly improves upon the existing runtimes of existing REC algorithms, without placing severe conditions on Q and P .

In addition, we proposed two methods to eliminate overhead codelength when encoding multiple samples. First, we introduced an approximate depth-limited variant of AD*

coding, DAD* coding, in which the codelength of the encoder is a tunable variable, and proved an upper bound for its bias. Second, we introduced the IsoKL parameterization, in which latent dimensions are grouped into blocks, and the KL divergences of all latent dimensions in a block are constrained to be equal. DAD* together with the IsoKL parameterization allow us to encode multiple samples with the same codelength, thereby amortising coding overheads, while maintaining low sample bias. Experimentally, we demonstrated the favourable runtimes of AS* and AD* coding on extensive toy experiments. We also showed that DAD* coding achieves levels of bias comparable to existing approximate REC algorithms, while maintaining a significantly faster runtime. On lossless image compression experiments on MNIST, DAD* together with an IsoKL VAE (IKVAE) parameterization achieved a compression rate close to the theoretically optimal ELBO.

Further work: One of the central remaining questions of this work is the runtime of AD* coding. Based on our experiments, we conjecture that the runtime of AD* coding is also $\mathcal{O}(D_{\infty}[Q\|P])$ whenever dQ/dP is unimodal, however this remains to be shown.

In general, for fixed $D_{\text{KL}}[Q\|P]$ we can have arbitrarily high $D_{\infty}[Q\|P]$, hence a second, more general question is if there exists a REC algorithm with $\mathcal{O}(D_{\text{KL}}[Q\|P])$ expected runtime. Adaptive rejection sampling (Gilks & Wild, 1992), OS* sampling (Dymetman et al., 2012) and ideas from (Theis & Yosri, 2022) could be good starting points for developing such an algorithm.

Another promising direction is to apply the IsoKL parameterization to larger VAEs, such as those used by Townsend et al. (2019), and scale our approach up to real-world compression tasks. Lastly, our methods can also be readily applied to lossy compression.

8. Author Contributions

GF discovered that A* sampling can be modified to obtain A* coding (alg. 1), which can be used to perform relative entropy coding, and provided proofs for theorems 1 and 3. SM provided a proof for theorem 2. GF and SM contributed equally to the experiments and the writing of this paper. JMH supervised and steered the project.

9. Acknowledgements

We would like to thank Rich Turner and Lennie Wells for useful feedback on an early manuscript of this paper. GF acknowledges funding from DeepMind. SM acknowledges funding from the Vice Chancellor’s & George and Marie Vergottis scholarship of the Cambridge Trust.

References

- Agustsson, E. and Theis, L. Universally quantized neural compression. *Advances in Neural Information Processing Systems*, 33, 2020.
- Ballé, J., Laparra, V., and Simoncelli, E. P. End-to-end optimized image compression. In *International Conference on Learning Representations*, 2017.
- Ballé, J., Minnen, D., Singh, S., Hwang, S. J., and Johnston, N. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations*, 2018.
- Ballé, J., Chou, P. A., Minnen, D., Singh, S., Johnston, N., Agustsson, E., Hwang, S. J., and Toderici, G. Nonlinear transform coding. *IEEE Journal of Selected Topics in Signal Processing*, 15(2):339–353, 2020.
- Bennett, C. H., Shor, P. W., Smolin, J. A., and Thapliyal, A. V. Entanglement-assisted capacity of a quantum channel and the reverse Shannon theorem. *IEEE Transactions on Information Theory*, 48(10):2637–2655, 2002.
- Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural networks. In Bach, F. and Blei, D. (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1613–1622, Lille, France, 07–09 Jul 2015. PMLR.
- Brezinski, C. Extrapolation algorithms and Padé approximations: a historical survey, 1994.
- Chatterjee, S. and Diaconis, P. The sample size required in importance sampling. *The Annals of Applied Probability*, 28(2):1099–1135, 2018.
- Chewi, S., Gerber, P. R., Lu, C., Le Gouic, T., and Rigollet, P. Rejection sampling from shape-constrained distributions in sublinear time. In *International Conference on Artificial Intelligence and Statistics*, pp. 2249–2265. PMLR, 2022.
- Corless, R. M., Gonnet, G. H., Hare, D. E., Jeffrey, D. J., and Knuth, D. E. On the lambertw function. *Advances in Computational mathematics*, 5(1):329–359, 1996.
- Dymetman, M., Bouchard, G., and Carter, S. The OS* algorithm: a joint approach to exact optimization and sampling. *arXiv preprint arXiv:1207.0742*, 2012.
- Flamich, G., Havasi, M., and Hernández-Lobato, J. M. Compressing images by encoding their latent representations with relative entropy coding. *Advances in Neural Information Processing Systems*, 33, 2020.
- Gilks, W. R. and Wild, P. Adaptive rejection sampling for gibbs sampling. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 41(2):337–348, 1992.
- Grimmett, G. and Stirzaker, D. *Probability and random processes*. Oxford University Press, U.S.A., 2001.
- Grimmett, G. and Welsh, D. *Probability: an introduction*. Oxford University Press, 2014.
- Harsha, P., Jain, R., McAllester, D., and Radhakrishnan, J. The communication complexity of correlation. In *Twenty-Second Annual IEEE Conference on Computational Complexity (CCC’07)*, pp. 10–23. IEEE, 2007.
- Havasi, M., Peharz, R., and Hernández-Lobato, J. M. Minimal random code learning: Getting bits back from compressed model parameters. In *International Conference on Learning Representations*, 2018.
- Hinton, G. E. and Van Camp, D. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the sixth annual conference on Computational learning theory*, pp. 5–13, 1993.
- Ho, J., Lohn, E., and Abbeel, P. Compression with flows via local bits-back coding. *Advances in Neural Information Processing Systems*, 32:3879–3888, 2019.
- Hoogeboom, E., Peters, J., van den Berg, R., and Welling, M. Integer discrete flows and lossless compression. *Advances in Neural Information Processing Systems*, 32:12134–12144, 2019.
- Kingma, D. P. and Welling, M. Auto-encoding variational Bayes. *International Conference on Learning Representations*, 2014.
- Kingman, J. *Poisson Processes*. Oxford Studies in Probability. Clarendon Press, 1992. ISBN 9780191591242.
- Lambert, Johann, H. Observationes variae in mathesin puram. *Acta Helvetica Physico-Mathematico-Anatomico-Botanico-Medica*, 3:128–168, 1758.
- Land, A. and Doig, A. An automatic method of solving discrete programming problems. *Econometrica*, 28(3): 497–520, 1960.
- Li, C. T. and El Gamal, A. Strong functional representation lemma and applications to coding theorems. *IEEE Transactions on Information Theory*, 64(11):6967–6978, 2018.
- Maddison, C. Poisson process model for Monte Carlo. *Perturbation, Optimization, and Statistics*, pp. 193–232, 2016.

- Maddison, C. J., Tarlow, D., and Minka, T. A* sampling. *Advances in Neural Information Processing Systems*, 27: 3086–3094, 2014.
- Markou, S. Notes on the runtime of A* sampling. *arXiv preprint arXiv.2205.15250*, 2022.
- Papandreou, G. and Yuille, A. L. Perturb-and-MAP random fields: Using discrete optimization to learn and sample from energy models. In *2011 International Conference on Computer Vision*, pp. 193–200. IEEE, 2011.
- Pérez-Cruz, F. Kullback-Leibler divergence estimation of continuous distributions. In *2008 IEEE international symposium on information theory*, pp. 1666–1670. IEEE, 2008.
- Theis, L. and Agustsson, E. On the advantages of stochastic encoders. In *Neural Compression Workshop at ICLR*, 2021.
- Theis, L. and Yosri, N. Algorithms for the communication of samples. In *International Conference on Machine Learning*, 2022.
- Townsend, J., Bird, T., and Barber, D. Practical lossless compression with latent variables using bits back coding. In *International Conference on Learning Representations*, 2018.
- Townsend, J., Bird, T., Kunze, J., and Barber, D. Hilloc: lossless image compression with hierarchical latent variable models. In *International Conference on Learning Representations*, 2019.
- van den Berg, R., Gritsenko, A. A., Dehghani, M., Sønderby, C. K., and Salimans, T. IDF++: Analyzing and improving integer discrete flows for lossless compression. In *International Conference on Learning Representations*, 2020.
- Zhang, S., Kang, N., Ryder, T., and Li, Z. iflow: Numerically invertible flows for efficient lossless compression via a uniform coder. *Advances in Neural Information Processing Systems*, 34, 2021.
- Ziv, J. On universal quantization. *IEEE Transactions on Information Theory*, 31(3):344–347, 1985.

A. Proof of Theorem 1

Notation: Throughout the appendix, we will write $[a : b] \stackrel{\text{def}}{=} [a, b] \cap \mathbb{N}$ for $a, b \in \mathbb{N}$ and $[a] \stackrel{\text{def}}{=} [1 : a]$. Furthermore, for a vector we will write $x_{1:n} \stackrel{\text{def}}{=} (x_1, \dots, x_n)$ for $n \in \mathbb{N}$.

We make use of the top-down construction of Gumbel processes (Algorithm 2; Maddison et al., 2014). The top-down construction realizes samples from a base distribution P along with their associated Gumbel values, which together form a Gumbel process. A Gumbel process can be thought of as a generalization of the Gumbel-Max trick (Papandreou & Yuille, 2011), where the log-probability of each member of a sample space is perturbed with i.i.d. Gumbel noise. Then, it can be shown that the maximum of this process is Gumbel distributed, and the argmaximum has law P . Gumbel processes can be shown to be equal in distribution to exponential races, where the time variable is mapped to its negative logarithm (Maddison, 2016). This is important, as it allows us to switch between the Gumbel and Poisson process representations, to leverage existing results in our analysis.

Algorithm 2 realizes its Gumbel process using a space partitioning binary tree construction while also recording the depths and heap indices of nodes. It is therefore an extension of Algorithm 1 in Maddison et al. (2014). Algorithm 2 can be realized using public randomness by anyone with access to the public seed S .

A* sampling can be viewed as performing a binary tree search on the Gumbel process with measure P , as realized by Algorithm 2, to search for a sample with distribution Q . The key observation is that for the search to proceed, the whole realization of the Gumbel process with measure P is not needed, and in fact it can be realized on-the-go.

A* coding (Algorithm 1) first runs the regular A* sampling procedure by simulating the Gumbel process with the proposal measure P using the publicly available randomness S . Then, it encodes the returned sample using the heap index H_n of the node n with which the sample is associated. Since any node with a given heap index can be simulated without reference to Q using Algorithm 2, the code returned by A* coding is always uniquely decodable given S , and the correctness of A* sampling (Maddison et al., 2014) will guarantee that the sample A* coding returns has the correct distribution Q .

PFR and ORC also operate on Gumbel/Poisson processes, however, they use a different encoding process. They encode the index K of a sample as opposed to its heap index H . For a sample x , K is obtained by sorting the arrival times in the Gumbel/Poisson process with measure P , and returning the index associated with x in the sorted list. PFR and ORC obtain K easily, because they realize the Gumbel/Poisson process in-order (for example, see Algorithm 3 in (Maddison et al., 2014)). Similarly, Algorithm 2 also constructs the process in order, however, it uses the top-down construction.

Theorem 1 shows that A* coding is not only correct and uniquely decodable, but its expected codelength is also optimal. However, to show this, we first show the following intermediate result, which relates the index K of a sample to its expected depth D in the top-down construction.

Lemma 3 (Average depth of nodes in a Gumbel process). *Let P be a Borel probability measure over some Polish space that is supplied to Algorithm 2. Assume that `partition` satisfies the following property: there exists $\epsilon \in [1/2, 1)$, such that for any node n we have*

$$\mathbb{E}[P(B_n)] \leq \epsilon^{D_n}, \quad (24)$$

where the expectation is taken over the joint distribution of the samples associated with the ancestor nodes of n . Let K_n be the index of a node n realized by Algorithm 2 and let D_n be its depth in the tree. Then,

$$\mathbb{E}[D_n \mid K_n] \leq -\log_\epsilon K_n. \quad (25)$$

Before delving into the proof, we clarify the allowed domain of ϵ . First note that ϵ is solely a property of `partition`. Note that fixing $\epsilon = 1$ would imply that the bounds do not shrink. The reason why $\epsilon \geq 1/2$ is because if `partition` breaks some region B into L and R with $P(L) = \epsilon_L P(B)$, then necessarily $P(R) = \epsilon_R P(B) = (1 - \epsilon_L)P(B)$, hence we need to take $\epsilon = \max\{\epsilon_L, \epsilon_R\}$. This means that the minimal ϵ is achieved when $\epsilon_L = \epsilon_R = 1/2$. Finally, since `partition` might depend on the samples drawn up to reaching the node with bound B , we need to take expectation over these.

Proof. Let $\mathcal{F}_k = (f_1, \dots, f_k)$ be the *frontier* of alg. 2 after k steps. We shall say that alg. 2 or alg. 1 *expand* a node, which means that they pop off the highest priority node from their priority queue. \mathcal{F}_k consists of all the nodes that could be expanded, starting with \mathcal{F}_1 only containing the root node. In the search literature \mathcal{F}_k is also commonly referred to as the open set of nodes. A simple inductive argument shows, that for a binary tree on k nodes will always have $k + 1$ nodes on its frontier, i.e. $|\mathcal{F}_k| = k$.

Let $\{(X_i, G_i)\}_{i=1}^{k-1}$ be the samples and Gumbels realized by alg. 2, sorted in descending order by the G_i s up to the $k-1$ largest one. Let $\mu_f = \log P(B_f)$ be the location parameter of the truncated Gumbel variate G_f for a node $f \in \mathcal{F}_k$. Maddison et al. (2014) show (see their Appendix, the section titled ‘‘Equivalence Under partition’’), that regardless of the choice of partition,

$$\forall f \in \mathcal{F}_k \quad G_f \sim \text{TG}(\mu_f, G_{k-1}). \quad (26)$$

Let $F_k \in \mathcal{F}_k$ denote the node that is expanded in step k , i.e.

$$F_k \sim \arg \max_{f \in \mathcal{F}_k} \{G_f\}. \quad (27)$$

Then a simple Gumbel-max trick-type argument shows (Maddison et al., 2014), that

$$\begin{aligned} p(F_k = f \mid G_{1:k-1}, X_{1:k-1}) &= \frac{\exp(\mu_f)}{\sum_{\phi \in \mathcal{F}_k} \exp(\mu_\phi)} \\ &= \frac{\exp(\log P(B_f))}{\sum_{\phi \in \mathcal{F}_k} \exp(\log P(B_\phi))} \\ &= \frac{P(B_f)}{\sum_{\phi \in \mathcal{F}_k} P(B_\phi)} \\ &= P(B_f), \end{aligned} \quad (28)$$

where the last equality holds because the bounds associated with the nodes on \mathcal{F}_k form a partition of the whole sample space for any k . Then,

$$\begin{aligned} \mathbb{E}_{p(F_k=f \mid G_{1:k-1}, X_{1:k-1})}[-\mu_f] &= - \sum_{f \in \mathcal{F}_k} P(B_f) \mu_f \\ &= - \sum_{f \in \mathcal{F}_k} P(B_f) \log P(B_f) \\ &= H[F_k] \\ &\leq \log k. \end{aligned} \quad (29)$$

The last equality follows, since the maximal Shannon entropy of a distribution over k items is the uniform distribution with entropy $\log k$.

Now, taking expectations over $G_{1:k-1}, X_{1:k-1}$, we get

$$\mathbb{E}_{p(F_k=f, G_{1:k-1}, X_{1:k-1} \mid k)}[-\mu_f] \leq \log k. \quad (30)$$

Finally,

$$\begin{aligned} \log k &\geq \mathbb{E}_{p(F_k=f, G_{1:k-1}, X_{1:k-1} \mid k)}[-\mu_f] \\ &= \mathbb{E}_{p(F_k=f, G_{1:k-1}, X_{1:k-1} \mid k)}[-\log P(B_f)] \\ &\geq -\log \mathbb{E}_{p(F_k=f, G_{1:k-1}, X_{1:k-1} \mid k)}[P(B_f)] \\ &\geq -\log \epsilon^{D_k} \\ &= -D_k \log \epsilon, \end{aligned} \quad (31)$$

where the second inequality holds by Jensen’s inequality and the third inequality holds by our assumption on partition. Since $0 < -\log \epsilon$, rearranging the two sides of the inequality gives the desired result. \square

With this result in mind, we are now ready to prove Theorem 1, which we state again for completeness:

Theorem 4 (Expected codelength of A* coding). *Let Q and P be the target and proposal measures passed to A* coding (Algorithm 1). Assume that partition satisfies the following property: there exists $\epsilon \in [1/2, 1)$ such that for any node n we have*

$$\mathbb{E}[P(B_n)] \leq \epsilon^{D_n}, \quad (32)$$

where the expectation is taken over the joint distribution of the samples associated with the ancestor nodes of n . Let k be the node returned by A* coding. Then, we have

$$\mathbb{E}[D_k] \leq -\frac{1}{\log \epsilon} [D_{\text{KL}}[Q||P] + e^{-1} + \log 2]. \quad (33)$$

In particular, when $\epsilon = 1/2$,

$$\mathbb{E}[D_k] \leq D_{\text{KL}}[Q||P]_2 + e^{-1} \log_2 e + 1. \quad (34)$$

Proof. Let K be the random variable that represents the index (not the heap index) of the sample returned by A* coding. Maddison (2016) show that K is equal in distribution to the index returned by running global bound A* sampling, which is equal in distribution to the index returned by PFR coding. In Li & El Gamal (2018), Appendix, section A it is shown, that

$$\mathbb{E}[\log K] \leq D_{\text{KL}}[Q||P] + e^{-1} + \log 2. \quad (35)$$

Putting this together with Lemma 3, we get the desired result. \square

B. Proof of Lemma 1

In the main text, the result is stated for AS* sampling. However, the result does not depend on the target Q , only the realization of the Gumbel process with base measure P in alg. 2, hence we restate the lemma here as follows:

Lemma 4. *Let P be a non-atomic proposal measure over a 1-dimensional sample space, passed to Algorithm 2, and let partition be as defined in eq. (13). Then the condition in eq. (10) is satisfied with $\epsilon = 3/4$, that is*

$$\mathbb{E}[P(B_n)] \leq \left(\frac{3}{4}\right)^{D_n}. \quad (36)$$

Proof. Let F denote the CDF of the measure P . We will prove the claim by induction. For the base case, note, that depth $D = 0$ can be associated with not having drawn any samples yet, i.e. the next node that is expanded by alg. 2 will be the root node. Since the sample is drawn from the whole space, we will have $P(B_1) = P(\Omega) = 1 = (3/4)^0$. For the hypothesis, assume the claim holds for $D = d$. Let $D = d + 1$. Fix a node n such that $D_n = d + 1$. Let $\mathcal{A}(n) = (n_1, \dots, n_{D_n-1})$ denote the ancestors of n , where n_1 is the root of the tree and $n_{D_n-1} = \text{par}(n)$ is the direct parent node of n in the tree constructed by alg. 2. Then, by the law of iterated expectations, we have

$$\begin{aligned} \mathbb{E}_{p(X_{\mathcal{A}(n)})}[P(B_n)] &= \\ &= \mathbb{E}_{p(X_{\mathcal{A}(\text{par}(n))})}[\mathbb{E}_{p(X_{\text{par}(n)}|X_{\mathcal{A}(\text{par}(n))})}[P(B_n)]]. \end{aligned} \quad (37)$$

Focusing on the inner expectation, let $B_{\text{par}(n)} = (a, b)$. Then, by the definition of partition, $L = (a, X_n)$, $R = (X_n, b)$. Note, that since B_n is either L or R , we have $P(B_n) \leq \max\{P(L), P(R)\}$. Furthermore, since the space is 1 dimensional, we get $P(L) = F(X_n) - F(a)$ and $P(R) = F(b) - F(X_n)$. Let $\mathcal{U}(c, d)$ denote the uniform density on (c, d) . Let $\alpha = F(a)$, $\beta = F(b)$. Then, by the generalized probability integral transform, we find, that $U \stackrel{\text{def}}{=} F(X_n) \sim \mathcal{U}(\alpha, \beta)$. Thus, by the law of the unconscious statistician, the inner expectation of eq. (37) can be rewritten as

$$\begin{aligned} \mathbb{E}_{p(X_{\text{par}(n)}|X_{\mathcal{A}(\text{par}(n))})}[P(B_n)] &\leq \mathbb{E}_{p(U)}[\max\{U - \alpha, \beta - U\}] \\ &= \int_{\alpha}^{\beta} \frac{\max\{u - \alpha, \beta - u\}}{\beta - \alpha} du \\ &= \frac{3}{4}(\beta - \alpha) \\ &= \frac{3}{4}P(B_{\text{par}(n)}). \end{aligned} \quad (38)$$

Now, by the induction hypothesis eq. (37) becomes

$$\frac{3}{4}\mathbb{E}_{p(X_{\mathcal{A}(\text{par}(n))})}[P(B_n)] \leq \left(\frac{3}{4}\right)^{d+1}, \quad (39)$$

which concludes the proof. \square

C. Proof of Theorem 2

Note: Our original argument for the linear runtime of A* coding contained an error. [Markou \(2022\)](#) provided a proof for theorem 2, which we reproduce here.

Overview: The proof breaks down the execution of AS* coding into two stages. For the first stage, we consider how AS* shrinks its search bounds, until it obtains a sufficiently good candidate sample. Here, a sufficiently good sample is a sample which falls within a predefined super-level set of the density ratio. Lemma 8 gives an upper bound on the expected number of steps in this first stage of the algorithm.

For the second stage, we quantify how many additional steps AS* must subsequently make until it terminates, after obtaining a good candidate sample in the first stage. Lemma 11 gives an upper bound on the expected number of steps in this second stage of the algorithm. Putting lemmas 8 and 11 together, we obtain an upper bound on the runtime of AS*, stated in corollary 1. This bound depends on, and holds for any, super-level set. Therefore, we can minimise this bound over all super-level sets of the density ratio. Lastly, we show that even for the worst case density ratios of this bound, this minimum results in a runtime is linear in the ∞ -divergence, resulting in theorem 2.

Notation: In this section, all indices to random variables are integers corresponding to the depth of the variable within the binary tree being searched. This is in contrast to other sections where the random variables are indexed by the node of the binary tree to which they belong. We found this notational overloading makes the exposition clearer, and the meaning of the indexing should be clear from the context.

Assumption 1 (Continuous distributions, finite D_∞). *We assume that measures Q and P describe continuous random variables, so their densities q and p exist. Since $P \gg Q$, the Radon-Nikodym derivative $r(x) = (dQ/dP)(x)$ also exists. We also assume $r(x)$ is unimodal and satisfies*

$$D_\infty[Q||P] = \log \sup_{x \in \mathbb{R}} \frac{dQ}{dP}(x) = \log r_{max} < \infty. \quad (40)$$

Without loss of generality, we can also assume P to be the uniform measure on $[0, 1]$, as shown by the next lemma. This is because we can push P and Q through the CDF of P to ensure P is uniform, while leaving the Radon-Nikodym derivative unimodal and the ∞ -divergence unchanged.

Lemma 5 (Without loss of generality, P is uniform). *Suppose Q is a target measure and P a proposal measure as specified in Assumption 1. Let Φ be the CDF associated with P and consider the measures $P', Q' : [0, 1] \rightarrow [0, \infty)$ defined as*

$$P' = P \circ \Phi^{-1} \text{ and } Q' = Q \circ \Phi^{-1}. \quad (41)$$

Then, P' is the uniform measure on $[0, 1]$. Further, the Radon-Nikodym derivative $dQ'/dP'(x)$ is unimodal, and

$$\log \sup_{z \in [0,1]} \frac{dQ'}{dP'}(z) = \log \sup_{x \in \mathbb{R}} \frac{dQ}{dP}(x). \quad (42)$$

Proof. First, P' is the uniform measure on $[0, 1]$ since for any $z \in [0, 1]$

$$P'([0, z]) = P \circ \Phi^{-1}([0, z]) = P((-\infty, \Phi^{-1}(z)]) = [0, z]. \quad (43)$$

Now, let the densities of Q and P be q and p , and the densities of Q' and P' be q' and p' . Then by the change of variables formula

$$p'(z) = p(\Phi^{-1}(z)) (\Phi^{-1})'(z) \text{ and } q'(z) = q(\Phi^{-1}(z)) (\Phi^{-1})'(z). \quad (44)$$

Therefore, we have

$$\frac{dQ'}{dP'}(z) = \frac{q'(z)}{p'(z)} = \frac{q(\Phi^{-1}(z))}{p(\Phi^{-1}(z))} = \frac{dQ}{dP} \circ \Phi^{-1}(z), \quad (45)$$

Now, since $dQ/dP(x)$ is a unimodal function of x and $\Phi^{-1}(z)$ is increasing in z , the function $(dQ/dP) \circ \Phi^{-1}(z)$ is unimodal in z . Also, by taking the the supremum and logarithm of both sizes

$$\log \sup_{z \in [0,1]} \frac{dQ'}{dP'}(z) = \log \sup_{z \in [0,1]} \frac{dQ}{dP} \circ \Phi^{-1}(z) = \log \sup_{x \in \mathbb{R}} \frac{dQ}{dP}(x), \quad (46)$$

arriving at the result. \square

We now define the super-level sets of the density ratio, and super-level set width functions, on which the argument relies.

Definition 2 (Superlevel set, width). *We define the superlevel-set function $S : [0, 1] \rightarrow 2^{[0,1]}$ as*

$$S(\gamma) = \{x \in [0, 1] \mid r(x) \geq \gamma r_{max}\}, \quad (47)$$

And let $x_{max} \in \{x \in [0, 1] \mid r(x) \leq r(x_{max})\}$ be an arbitrary maximiser of the density ratio. We also define the superlevel-set width function $w : [0, 1] \rightarrow [0, 1]$ as

$$w(\gamma) = \inf\{\delta \in [0, 1] \mid \exists z \in [0, 1], S(\gamma) \subseteq [z, z + \delta]\}. \quad (48)$$

Because width functions are defined in terms of a ratio of probability densities, they satisfy certain properties, stated in lemma 6 and proved below. We use these properties later to prove lemma 12.

Lemma 6 (Properties of w). *The width function $w(\gamma)$ is non-increasing in γ and satisfies*

$$\int_0^1 w(\gamma) d\gamma = \frac{1}{r_{max}} \text{ and } w(0) \geq \frac{1}{r_{max}}. \quad (49)$$

Proof. First, we note that if $\gamma_1 \leq \gamma_2$, then $S(\gamma_2) \subseteq S(\gamma_1)$ which implies $w(\gamma_2) \leq w(\gamma_1)$. Therefore

$$\gamma_1 \leq \gamma_2 \implies w(\gamma_2) \leq w(\gamma_1), \quad (50)$$

so $w(\gamma)$ is decreasing in γ . Second, let $A = [0, 1] \times [0, r_{max}]$, define $B = \{(x, y) \in A \mid y \leq r(x)\}$ and consider the integral

$$I = \int_A \mathbb{1}(z \in B) dz. \quad (51)$$

Since this the integrand is a non-negative measurable function, by Fubini's theorem, we have

$$I = \int_0^1 \int_0^{r_{max}} \mathbb{1}((x, y) \in B) dy dx = \int_0^{r_{max}} \int_0^1 \mathbb{1}((x, y) \in B) dx dy \quad (52)$$

$$\int_0^1 r(x) dx = \int_0^{r_{max}} w(y/r_{max}) dy \quad (53)$$

$$\int_0^1 q(x) dx = \int_0^1 w(\gamma) r_{max} d\gamma \quad (54)$$

$$\int_0^1 w(\gamma) d\gamma = r_{max}^{-1}. \quad (55)$$

Last, since w is non-increasing, we have $w(0) \geq r_{max}^{-1}$, because otherwise $\int_0^1 w(\gamma) d\gamma < r_{max}^{-1}$. \square

Now we define the two stages in which we break down the execution AS* coding. In particular, we define $N(\gamma)$ as the number of steps required until AS* gives a sample in the superlevel set $S(\gamma)$, and we define $K(\gamma)$ as the number of subsequent steps required for AS* to terminate.

Definition 3 (# steps to $S(\gamma)$, # residual steps). *Suppose AS* is applied to a target-proposal pair Q, P satisfying Assumption 1, producing a sequence of samples X_1, X_2, \dots . We use $T \in \mathbb{Z}$ to denote the total number of steps taken by AS* until it terminates and define the random variables*

$$N(\gamma) = \min\{n \in \mathbb{Z} \mid X_n \in S(\gamma)\} \text{ and } K(\gamma) = \max\{0, T - N(\gamma)\}. \quad (56)$$

Because the bounds of AS* shrink exponentially quickly, we can bound the probability that AS* in the first stage, by a quantity which also shrinks exponentially, as stated in lemma 7.

Lemma 7 (Upper bound on the probability of $P(B_n) \geq w(\gamma)$). *Let $Z_n = P(B_n)$. Then*

$$\mathbb{P}(Z_n \geq w(\gamma)) \leq \frac{1}{w(\gamma)} \left(\frac{3}{4}\right)^{n-1} \quad (57)$$

Proof. Let $Z_n = P(B_n)$. Noting that $Z_n \geq 0$, we apply Markov's inequality and lemma 4 to get

$$\mathbb{P}(Z_n \geq w(\gamma)) \leq \frac{1}{w(\gamma)} \mathbb{E}[Z_n] \leq \frac{1}{w(\gamma)} \left(\frac{3}{4}\right)^{n-1}, \quad (58)$$

as required. □

Now using lemma 7 we can upper bound the expectation over $N(\gamma)$, which depends on the logarithm of the width $w(\gamma)$.

Lemma 8 (Bound on expected $N(\gamma)$). *The random variable $N(\gamma)$ satisfies*

$$\mathbb{E}[N(\gamma)] \leq \alpha \log \frac{1}{w(\gamma)} + 6, \text{ where } \alpha = \left(\log \frac{4}{3}\right)^{-1}. \quad (59)$$

Proof. Let $N_0 = \left\lceil \frac{\log w(\gamma)}{\log(3/4)} \right\rceil + 1$. Also let B_0, B_1, \dots be the bounds produced by AS*. Noting that by the unimodality of r , $S(\gamma)$ is an interval with $x_{max} \in S(\gamma)$, and $x_{max} \in B_n$, we have

$$P(B_n) < w(\gamma) \implies N(\gamma) \leq n, \quad (60)$$

that is, the event $P(B_n) < w(\gamma)$ implies the event $N(\gamma) \leq n$. From this it follows that

$$\mathbb{P}(P(B_n) < w(\gamma)) \leq \mathbb{P}(N(\gamma) \leq n) \implies \mathbb{P}(P(B_n) \geq w(\gamma)) \geq \mathbb{P}(N(\gamma) \geq n + 1). \quad (61)$$

Using this together with lemma 7, we can write

$$\mathbb{E}_{B_0:\infty}[N(\gamma)] = \sum_{n=1}^{\infty} \mathbb{P}(N(\gamma) = n) n \quad (62)$$

$$= \sum_{n=1}^{\infty} \mathbb{P}(N(\gamma) \geq n) \quad (63)$$

$$\leq N_0 + \sum_{n=N_0+1}^{\infty} \mathbb{P}(N(\gamma) \geq n) \quad (64)$$

$$= N_0 + \sum_{n=1}^{\infty} \mathbb{P}(N(\gamma) \geq N_0 + n) \quad (65)$$

$$\leq N_0 + \sum_{n=1}^{\infty} \mathbb{P}(B_{N_0+n-1} \geq w(\gamma)) \quad (66)$$

$$\leq N_0 + \sum_{n=1}^{\infty} \frac{1}{w(\gamma)} \left(\frac{3}{4}\right)^{N_0+n-1} \quad (67)$$

$$\leq N_0 + \sum_{n=1}^{\infty} \left(\frac{3}{4}\right)^n \quad (68)$$

$$= N_0 + 4 \quad (69)$$

$$\leq \frac{\log w(\gamma)}{\log 3/4} + 6, \quad (70)$$

where the equality of 62 and 63 is a standard identity (Grimmett & Welsh, 2014), 63 to 64 follows by the fact that probabilities are bounded above by 1, 64 to 65 follows by relabelling the indices, 65 to 66 follows from eq. (61), 66 to 67 follows from follows from lemma 7 and 66 to 67 follows from our definition of N_0 . \square

Now we turn to bounding the expectation of $K(\gamma)$. For this, we must consider how the difference between the upper and lower bounds maintained by the search shrinks. To do so, we will use lemma 10. Lemma 9 is an intermediate result, which we use to show lemma 10.

Lemma 9 (Exponentials and Truncated Gumbels). *Let $T \sim \text{Exp}(\lambda)$ and $T_0 \geq 0$. Then*

$$Z \stackrel{\text{def}}{=} -\log(T + T_0) \stackrel{d}{=} G \text{ where } G \sim \text{TG}(\log \lambda, -\log T_0). \quad (71)$$

Proof. Let $T \sim \text{Exp}(\lambda)$, $T_0 \geq 0$ and define

$$Z \stackrel{\text{def}}{=} -\log(T + T_0). \quad (72)$$

We note that $Z \leq -\log T_0$. For $Z \leq -\log T_0$, we can apply the change of variables formula to obtain the density of Z . Let p_Z and p_T be the densities of Z and T . Then

$$p_Z(z) = p_T(t) \left| \frac{dt}{dz} \right| \quad (73)$$

$$= \lambda e^{-\lambda t} \left| \frac{d}{dz}(e^{-z} - T_0) \right| \quad (74)$$

$$\propto e^{-\lambda e^{-z}} e^{-z} \quad (75)$$

$$\propto e^{-z - e^{-(z-\kappa)}}, \quad (76)$$

where $\kappa = \log \lambda$. Therefore Z has distribution $\text{TG}(\log \lambda, -\log T_0)$. \square

Lemma 10 (Mean of exponentiated negative truncated Gumbel). *Let B_1, \dots, B_N be the first N bounds produced by AS*, let G_N be the N^{th} Gumbel produced by AS* and define $E_N = e^{-G_N}$. Then*

$$\mathbb{E}[E_N | B_{1:N}] = \sum_{n=1}^N \frac{1}{P(B_n)}. \quad (77)$$

Proof. Define $E_n = e^{-G_n}$ for $n = 1, 2, \dots$. By the definition of AS*, we have

$$G_N | G_{N-1}, B_{1:N} \sim \text{TG}(\log P(B_N), G_{N-1}). \quad (78)$$

Negating G_N and G_{N-1} , taking exponentials and applying Jensen's inequality together with ineq. (9), we obtain

$$E_N | T_{N-1}, B_{0:N} \stackrel{d}{=} \tau_N + T_{N-1}, \text{ where } \tau_{N-1} \sim \text{Exp}(P(B_N)). \quad (79)$$

Repeating this step and taking expectations, we have

$$\mathbb{E}[E_N | B_{0:N}] = \mathbb{E} \left[\sum_{n=1}^N \tau_n | B_{0:N} \right] = \sum_{n=1}^N \frac{1}{P(B_n)} \quad (80)$$

as required. \square

Using lemma 10 we can bound the expectation over $K(\gamma)$, as stated and proved in lemma 11. Note that this bound does not depend on $N(\gamma)$, which has been marginalised out.

Lemma 11 (Bound on expected $K(\gamma)$). *The random variable $K(\gamma)$ satisfies*

$$\mathbb{E}[K(\gamma)] \leq \alpha \left(\log \frac{1}{\gamma} + \log \frac{1}{w(\gamma)} \right) + 16, \text{ where } \alpha = \left(\log \frac{4}{3} \right)^{-1}. \quad (81)$$

Proof. Let the global upper and lower bounds of AS* at step n be U_n and L_n respectively. Then, by the definition of the upper bound of AS* coding

$$U_{N(\gamma)} = \log r_{max} + G_{N(\gamma)}, \quad (82)$$

and also, by the definition of the lower bound of AS* coding

$$L_{N(\gamma)} = \max_{n \in [1:N(\gamma)]} \{ \log r(x_n) + G_n \} \geq \log r(x_{N(\gamma)}) + G_{N(\gamma)} \geq \log \gamma r_{max} + G_{N(\gamma)}. \quad (83)$$

Now for $k = 0, 1, 2, \dots$, we have

$$U_{N(\gamma)+k} - L_{N(\gamma)+k} \leq 0 \implies T \leq N(\gamma) + k, \quad (84)$$

that is, the event $U_{N(\gamma)+k} - L_{N(\gamma)+k} \leq 0$ implies the event $T \leq N(\gamma) + k$. This is because if $U_{N(\gamma)+k} - L_{N(\gamma)+k} \leq 0$, then the algorithm has terminated by step $N(\gamma) + k$, so it follows that $T \leq N(\gamma) + k$. Further

$$U_{N(\gamma)+k} - L_{N(\gamma)+k} \leq U_{N(\gamma)+k} - L_{N(\gamma)} \quad (85)$$

$$\leq \log r_{max} + G_{N(\gamma)+k} - \log \gamma r_{max} - G_{N(\gamma)} \quad (86)$$

$$= \log \frac{1}{\gamma} + G_{N(\gamma)+k} - G_{N(\gamma)}. \quad (87)$$

Therefore, we have

$$G_{N(\gamma)+k} - G_{N(\gamma)} \leq \log \gamma \implies T \leq N(\gamma) + k \implies K(\gamma) \leq k, \quad (88)$$

that is, the event $G_{N(\gamma)+k} - G_{N(\gamma)} \leq \log \gamma$ implies the event $K(\gamma) \leq k$. This holds because if $G_{N(\gamma)+k} - G_{N(\gamma)} \leq \log \gamma$, then $U_{N(\gamma)+k} - L_{N(\gamma)+k} \leq 0$, which in turn implies $K(\gamma) \leq k$. Therefore

$$\mathbb{P}(G_{N(\gamma)+k} - G_{N(\gamma)} \leq \log \gamma) \leq \mathbb{P}(K(\gamma) \leq k) \implies \mathbb{P}(G_{N(\gamma)+k} - G_{N(\gamma)} \geq \log \gamma) \geq \mathbb{P}(K(\gamma) \geq k + 1). \quad (89)$$

Equation (89) upper bounds the probability that the second stage of the algorithm has not terminated, by the probability that the Gumbel values have decreased sufficiently. To proceed, we turn to lower bounding the probability of the complementary event $G_{N(\gamma)+k} - G_{N(\gamma)} \leq \log \gamma$. Let $\Phi_{TG}(g; \mu, \kappa)$ denote the CDF of a truncated Gumbel distribution with location parameter μ and unit scale parameter, truncated at κ . Then

$$\mathbb{P}(G_{N(\gamma)+n} - G_{N(\gamma)} \leq \log \gamma \mid N(\gamma), G_{N(\gamma)}, B_{0:N(\gamma)+n}) = \quad (90)$$

$$= \mathbb{E}_{G_{N(\gamma)+n-1}} [\mathbb{P}(G_{N(\gamma)+n} - G_{N(\gamma)} \leq \log \gamma \mid N(\gamma), G_{N(\gamma)}, G_{N(\gamma)+n-1}, B_{0:N(\gamma)+n})] \quad (91)$$

$$= \mathbb{E}_{G_{N(\gamma)+n-1}} [\Phi_{TG}(\log \gamma + G_{N(\gamma)}; \log P(B_{N(\gamma)+n}), G_{N(\gamma)+n-1})] \quad (92)$$

$$\geq \Phi_{TG}(\log \gamma + G_{N(\gamma)}; \log P(B_{N(\gamma)+n}), \infty) \quad (93)$$

$$= e^{-e^{-(\log \gamma + G_{N(\gamma)} - \log P(B_{N(\gamma)+n}))}} \quad (94)$$

$$= e^{-\frac{1}{\gamma} P(B_{N(\gamma)+n})} e^{-G_{N(\gamma)}}. \quad (95)$$

Taking an expectation over $G_{N(\gamma)}$ and $B_{0:N(\gamma)+n}$, we have

$$\mathbb{P}(G_{N(\gamma)+n} - G_{N(\gamma)} \leq \log \gamma \mid N(\gamma)) \geq \mathbb{E}_{G_{N(\gamma)}, B_{0:N(\gamma)+n}} \left[e^{-\frac{1}{\gamma} P(B_{N(\gamma)+n})} e^{-G_{N(\gamma)}} \mid N(\gamma) \right] \quad (96)$$

$$\geq e^{-\frac{1}{\gamma} \mathbb{E}_{G_{N(\gamma)}, B_{0:N(\gamma)+n}} [P(B_{N(\gamma)+n}) e^{-G_{N(\gamma)}} \mid N(\gamma)]} \quad (97)$$

Focusing on the term in the exponent, we have

$$\mathbb{E}_{G_{N(\gamma)}, B_{0:N(\gamma)+n}} [P(B_{N(\gamma)+n}) e^{-G_{N(\gamma)}} | N(\gamma)] = \quad (98)$$

$$= \mathbb{E}_{B_{0:N(\gamma)-1}} \left[\mathbb{E}_{G_{N(\gamma)}, B_{N(\gamma):N(\gamma)+n}} [P(B_{N(\gamma)+n}) e^{-G_{N(\gamma)}} | B_{0:N(\gamma)-1}, N(\gamma)] \mid N(\gamma) \right] \quad (99)$$

$$\leq \mathbb{E}_{B_{0:N(\gamma)-1}} \left[\mathbb{E}_{G_{N(\gamma)}} \left[\left(\frac{3}{4} \right)^{n+1} P(B_{N(\gamma)-1}) e^{-G_{N(\gamma)}} \mid B_{0:N(\gamma)-1} \right] \mid N(\gamma) \right] \quad (100)$$

$$= \mathbb{E}_{B_{0:N(\gamma)-1}} \left[\left(\frac{3}{4} \right)^{n+1} P(B_{N(\gamma)-1}) \sum_{n=0}^{N(\gamma)-1} \frac{1}{P(B_n)} \mid N(\gamma) \right] \quad (101)$$

$$\leq \left(\frac{3}{4} \right)^{n+1} N(\gamma). \quad (102)$$

Substituting eq. (102) into eq. (97), we obtain

$$\mathbb{P}(G_{N(\gamma)+n} - G_{N(\gamma)} \leq \log \gamma \mid N(\gamma)) \geq e^{-\frac{N(\gamma)}{\gamma} \left(\frac{3}{4} \right)^{n+1}} \quad (103)$$

and applying ineq. (69) to this we obtain

$$\mathbb{P}(G_{N(\gamma)+n} - G_{N(\gamma)} \leq \log \gamma) \geq e^{-\frac{1}{\gamma} \left(\frac{3}{4} \right)^{n+1} (N_0+4)}, \quad (104)$$

arriving at a deterministic lower bound on which does not depend on any random quantities. Now we also have

$$\log \frac{1}{\gamma} + \log(N_0 + 4) = \log \frac{1}{\gamma} + \log \left(\left\lceil \frac{\log w(\gamma)}{\log(3/4)} \right\rceil + 5 \right) \quad (105)$$

$$\leq \log \frac{1}{\gamma} + \log \left(\frac{\log w(\gamma)}{\log(3/4)} + 6 \right) \quad (106)$$

$$\leq \log \frac{1}{\gamma} + \log \frac{1}{w(\gamma)} + 2, \quad (107)$$

where going from 106 to 107 can be verified numerically. Therefore, letting $K_0 = \left\lceil \frac{\log(1/\gamma) + \log(1/w(\gamma)) + 2}{\log(4/3)} \right\rceil$, we have

$$\mathbb{E}[K(\gamma)] = \sum_{k=0}^{\infty} \mathbb{P}(K(\gamma) = k) k \quad (108)$$

$$= \sum_{k=0}^{\infty} \mathbb{P}(K(\gamma) \geq k) \quad (109)$$

$$\leq K_0 + \sum_{k=K_0+1}^{\infty} \mathbb{P}(K(\gamma) \geq k) \quad (110)$$

$$= K_0 + \sum_{k=1}^{\infty} \mathbb{P}(K(\gamma) \geq K_0 + k) \quad (111)$$

$$\leq K_0 + \sum_{k=1}^{\infty} \mathbb{P}(G_{N(\gamma)+K_0+k-1} - G_{N(\gamma)} > \log \gamma) \quad (112)$$

$$\leq K_0 + \sum_{k=1}^{\infty} \left(1 - e^{-\left(\frac{3}{4} \right)^k} \right) \quad (113)$$

$$\leq K_0 + 4 \quad (114)$$

$$\leq \frac{\log \gamma}{\log(3/4)} + \frac{\log w(\gamma)}{\log(3/4)} + 16. \quad (115)$$

where the equality of 108 and 109 is a standard identity (Grimmett & Stirzaker, 2001), 109 to 110 follows because probabilities are bounded above by 1, 110 to 111 follows by relabelling the indices, 111 to 112 follows by ineq. (89), 112 to 113 follows by ineq. (104) and the definition of K_0 , 113 to 114 can be verified by evaluating the sum using numerical means and 114 to 115 follows by the definition of K_0 . \square

Putting lemmas 8 and 11 together, we obtain corollary 1, which is a bound on the expected runtime of AS*. This holds for any width function w and any $\gamma \in [0, 1]$. Note that whenever $\gamma = 0$ or $w(\gamma) = 0$ this bound becomes vacuous.

Corollary 1 (Upper bound on T for given w). *For any $\gamma \in [0, 1]$, the total number of steps, T , satisfies*

$$\mathbb{E}[T] \leq 2\alpha \left(\log \frac{1}{w(\gamma)} + 2 \log \frac{1}{\gamma} \right) + 22, \text{ where } \alpha = \left(\log \frac{4}{3} \right)^{-1} \quad (116)$$

Proof. By the definition of $N(\gamma)$ and $K(\gamma)$, we have

$$T \leq N(\gamma) + K(\gamma) \implies \mathbb{E}[T] \leq \mathbb{E}[N(\gamma) + K(\gamma)], \quad (117)$$

for all $\gamma \in [0, 1]$. From lemma 8 and lemma 11, we have

$$\mathbb{E}[T] \leq \alpha \left(2 \log \frac{1}{w(\gamma)} + \log \frac{1}{\gamma} \right) + 22 \leq 2\alpha \left(\log \frac{1}{w(\gamma)} + 2 \log \frac{1}{\gamma} \right) + 22, \quad (118)$$

where $\alpha = \log(4/3)^{-1}$ as required. \square

Note that in eq. (118) we obtain a bound which we intentionally make looser. This step results in a looser bound but facilitates subsequent manipulations easier. While a more careful analysis may result in a tighter bound, it can only improve our bound by a scaling factor, and we leave this as a point for further work.

Since corollary 1 holds for any $\gamma \in [0, 1]$, we can minimise the right hand side with respect to γ to make the bound as tight as possible. This results in a bound that is a function of w , however we are interested in producing a bound that holds for all w . Therefore, after minimising with respect to γ , we will consider the worst possible width functions which maximise the resulting quantity, and show that even for these worst-case width functions, the bound is linear in r_{max} . Definition 4 introduces the family of these worst-case width functions for a given r_{max} .

Definition 4 (Bound functions f, g, h , worst-case width set W^*). *We define*

$$f(\gamma, w) = \log \frac{1}{w(\gamma)}, \quad g(\gamma) = 2 \log \frac{1}{\gamma} \text{ and } h(\gamma, w) = f(\gamma, w) + g(\gamma). \quad (119)$$

For fixed r_{max} , let $W(r_{max})$ be the set of all possible width functions. We define the set W^* of worst-case width functions as

$$W^* = \left\{ w^* \in W(r_{max}) \mid \inf_{\gamma'} h(\gamma', w^*) \geq \inf_{\gamma'} h(\gamma', w) \forall w \in W(r_{max}) \right\}. \quad (120)$$

We refer to members of this set as worst-case width functions.

Next, for a given r_{max} , we define a width function \tilde{w} with a particular form, and show that $\tilde{w} \in W(r_{max})$. We also show that if $w \in W(r_{max})$ is any other width function, then

$$\inf_{\gamma} h(\gamma, \tilde{w}) \geq \inf_{\gamma} h(\gamma, w), \quad (121)$$

from which it follows that \tilde{w} is a worst case width function, that is $\tilde{w} \in W^*$.

Lemma 12 (An explicit worst case width function). *The function*

$$\tilde{w}(\gamma) = \begin{cases} 1 & \text{for } 0 \leq \gamma \leq \tilde{\gamma} \\ (\tilde{\gamma}/\gamma)^2 & \text{for } \tilde{\gamma} < \gamma \leq 1 \end{cases}, \quad (122)$$

where $\tilde{\gamma} = 1 - \sqrt{1 - r_{max}^{-1}}$, is a width function and $\tilde{w} \in W(r_{max})$. Further, if $w \in W(r_{max})$ then

$$\inf_{\gamma} h(\gamma, \tilde{w}) \geq \inf_{\gamma} h(\gamma, w). \quad (123)$$

Proof. Suppose $w \in W(r_{max})$ and let

$$m = \inf_{\gamma} h(\gamma, \tilde{w}), \quad (124)$$

let γ_m be the point where g equals m , that is

$$g(\gamma_m) = 2 \log \frac{1}{\gamma_m} = m \implies \gamma_m = e^{-m/2}. \quad (125)$$

Define $v : [0, 1] \times [0, 1] \rightarrow [0, 1]$ as

$$v(\gamma, \gamma') = \begin{cases} 1 & 0 \leq \gamma \leq \gamma' \\ (\gamma'/\gamma)^2 & \gamma' < \gamma \leq 1 \end{cases}, \quad (126)$$

and consider $v(\gamma, \gamma_m)$ as a function of γ . Note that $v(\gamma, \gamma_m)$ may not be in $W(r_{max})$ because, while it is non-increasing and continuous, it may not integrate to r_{max}^{-1} . In particular it holds that

$$h(\gamma, v(\gamma, \gamma_m)) \leq h(\gamma, \tilde{w}) \text{ for all } \gamma \in [0, 1] \implies v(\gamma, \gamma_m) \geq w(\gamma) \implies \int_0^1 v(\gamma, \gamma_m) d\gamma \geq r_{max}^{-1}. \quad (127)$$

Now, note that

$$\int_0^1 v(\gamma, \gamma') d\gamma = 2\gamma' - (\gamma')^2. \quad (128)$$

By the intermediate value theorem, there exists some $0 < \tilde{\gamma} \leq \gamma_m$ such that $2\tilde{\gamma} - \tilde{\gamma}^2 = r_{max}^{-1}$. For this $\tilde{\gamma}$, we define $\tilde{w}(\gamma) = v(\gamma, \tilde{\gamma})$, which is a width function because it is decreasing and integrates to 1. Specifically, $\tilde{w}(\gamma)$ is in $W(r_{max})$ because the probability density function

$$q(x) = r_{max} \min \left\{ 1, \tilde{\gamma} x^{-1/2} \right\}, \quad (129)$$

has $\tilde{w}(\gamma)$ as its width function. In addition note that $\tilde{\gamma} \leq \gamma_m$ so we have

$$\tilde{\gamma} \leq \gamma_m \implies \inf_{\gamma} h(\gamma, \tilde{w}(\gamma)) = \inf_{\gamma} h(\gamma, v(\gamma, \tilde{\gamma})) \geq \inf_{\gamma} h(\gamma, v(\gamma, \gamma_m)) = \inf_{\gamma} h(\gamma, w(\gamma)). \quad (130)$$

Therefore it holds that

$$w \in W(r_{max}) \implies \inf_{\gamma} h(\gamma, \tilde{w}) \geq \inf_{\gamma} h(\gamma, w), \quad (131)$$

from which it follows that $\tilde{w} \in W^*$ is a width function. \square

Last, we can put corollary 1 together with lemma 12 to arrive at the main result.

Theorem (AS* runtime upper bound). *Let T be the total number of steps taken by AS* until it terminates. Then*

$$\mathbb{E}[T] \leq 2\alpha \log r_{max} + 2\alpha \log 2 + 22. \quad (132)$$

Proof. Suppose AS* is applied to a target Q and proposal P with $D_{\infty}[Q||P] = r_{max}$, and corresponding width function $w \in W(r_{max})$. Now consider the worst case width function \tilde{w} defined in lemma 12, and note that

$$\tilde{\gamma} = 1 - \sqrt{1 - r_{max}^{-1}} \geq \frac{1}{2r_{max}}. \quad (133)$$

Then we have

$$\mathbb{E}[T] \leq 2\alpha \inf_{\gamma} h(\gamma, w) + 22 \leq 2\alpha \inf_{\gamma} h(\gamma, \tilde{w}) + 22 \leq 2\alpha h(\tilde{\gamma}, \tilde{w}) + 22, \quad (134)$$

and substituting the expression for h we obtain

$$\mathbb{E}[T] \leq 2\alpha \left(\log \frac{1}{\tilde{w}(\tilde{\gamma})} + 2 \log \frac{1}{\tilde{\gamma}} \right) + 22 \leq 4\alpha \log r_{max} + 4\alpha \log 2 + 22, \quad (135)$$

arriving at the result. \square

D. Proof of Theorem 3

Before we state the precise form of Theorem 3, we clarify what the precise form the approximate distribution \tilde{Q}_D of the output of DAD* is for target Q and proposal P for depth limit $D_{max} = D$. This means that there are $N = 2^D - 1$ nodes in the binary tree constructed by alg. 1 with associated Gumbel values and samples $\{(G_i, X_i)\}_{i=1}^N$. Let $r = \frac{dQ}{dP}$ and $r_i = r(X_i)$. Then, DAD* searches for $I \stackrel{\text{def}}{=} \arg \max_{i \in [N]} \{\log r(X_i) + G_i\}$, which can therefore be interpreted as simply performing the Gumbel-max trick on N atoms. Hence, we know, that

$$\begin{aligned} w_i \stackrel{\text{def}}{=} p(I = i \mid X_{1:N}) &= \frac{\exp(\log r_i)}{\sum_{j=1}^N \exp(\log r_j)} \\ &= \frac{r_i}{\sum_{j=1}^N r_j}. \end{aligned} \quad (136)$$

Hence, we finally get

$$\tilde{q}_D(X) = \sum_{i=1}^N w_i \delta(X - X_i) \quad (137)$$

where δ denotes the Dirac delta function.

Then, the \tilde{Q}_D -expectation of a measurable function f is

$$\mathbb{E}_{\tilde{Q}_D}[f] = \sum_{i=1}^N w_i f(X_i) = \frac{\sum_{i=1}^N r_i f(X_i)}{\sum_{j=1}^N r_j}. \quad (138)$$

Now, define

$$\mathcal{I}_D(f) \stackrel{\text{def}}{=} \frac{1}{N} \sum_{i=1}^N f(X_i) r_i. \quad (139)$$

Note, that $\mathcal{I}_D(f)$ looks very similar to the usual importance sampling estimator for Q -expectation the function f using N samples with distribution P . However, in this case the X_i s used in $\mathcal{I}_D(f)$ are not identically distributed, though they are independent. In particular, let n be a node in the tree realized by alg. 2 with $D_{max} = D$. Then, $X_n \sim P|_{B_n}$. The importance of $\mathcal{I}_D(f)$ is that the \tilde{Q}_D -expectation of f in eq. (138) can be written as

$$\mathbb{E}_{\tilde{Q}_D}[f] = \frac{\mathcal{I}_D(f)}{\mathcal{I}_D(1)}, \quad (140)$$

where 1 in the above is the constant function identically equal to 1. Hence, we begin by investigating the properties of $\mathcal{I}_D(f)$. First, we show that it is unbiased:

Lemma 13 (Unbiasedness of $\mathcal{I}_D(f)$). *Let Q , P , r , f and $\mathcal{I}_D(f)$ be defined as above. Then,*

$$\mathbb{E}_{P(X_{1:N})}[\mathcal{I}_D(f)] = \mathbb{E}_Q[f]. \quad (141)$$

Proof. Let \mathcal{T}_D denote the set of all nodes in the binary tree constructed by alg. 2 with $D_{max} = D$. We first note, that for a node $n \in \mathcal{T}_D$, we know $X_n \sim P|_{B_n}$, hence $dP(X_n)|_{B_n} = \frac{dP(X_n)}{P(B_n)}$ for $X_n \in B_n$. Furthermore, since partition is dyadic, we know that for a fixed depth $1 \leq d \leq D$, for every node n with $D_n = d$ it holds, that $P(B_n) = 2^{-(d-1)}$.

Furthermore, by construction, the bounds at a given depth are disjoint and form a partition of the whole space. Therefore,

$$\begin{aligned}
 \mathbb{E}_{p(X_{1:N})}[\mathcal{I}_D(f)] &= \\
 &= \mathbb{E}_{p(X_{1:N})} \left[\frac{1}{N} \sum_{n \in \mathcal{T}_D} r_n f(X_n) \right] \\
 &= \mathbb{E}_{p(X_{1:N})} \left[\frac{1}{N} \sum_{d=1}^D \sum_{\substack{n \in \mathcal{T}_D \\ D_n=d}} r_n f(X_n) \right] \\
 &= \frac{1}{N} \sum_{d=1}^D \sum_{\substack{n \in \mathcal{T}_D \\ D_n=d}} \int_{B_n} f(X_n) \frac{dQ}{dP}(X_n) \frac{dP(X_n)}{P(B_n)} \\
 &= \frac{1}{N} \sum_{d=1}^D 2^{d-1} \sum_{\substack{n \in \mathcal{T}_D \\ D_n=d}} \int_{B_n} f(X_n) dQ(X_n) \\
 &= \frac{1}{N} \sum_{d=1}^D 2^{d-1} \int_{\Omega} f(X_n) dQ(X_n) \\
 &= \mathbb{E}_Q[f] \frac{1}{N} \sum_{d=1}^D 2^{d-1} \\
 &= \mathbb{E}_Q[f],
 \end{aligned} \tag{142}$$

as required. \square

Given the unbiasedness of $\mathcal{I}_D(f)$, the rest of the proof follows mostly that of [Chatterjee & Diaconis \(2018\)](#) with appropriate modifications in the necessary places. Hence, we begin with the following lemma:

Lemma 14 (Mean absolute deviation bound for $\mathcal{I}_D(f)$). *Let Q, P, r, f and $\mathcal{I}_D(f)$ be defined as above. Let*

$$K \stackrel{\text{def}}{=} \lfloor D_{\text{KL}}[Q||P]_2 \rfloor, \quad D \stackrel{\text{def}}{=} K + t, \quad N \stackrel{\text{def}}{=} 2^D, \tag{143}$$

where t is a non-negative integer, and $Y \sim Q$. Define

$$\|f\|_Q \stackrel{\text{def}}{=} \sqrt{\mathbb{E}_Q[f^2]}. \tag{144}$$

Let

$$b \stackrel{\text{def}}{=} \left(2^{-t/4} \sqrt{1 + \frac{1}{N}} + 2\sqrt{\mathbb{P}[\log_2 r(Y) > K + t/2]} \right) \tag{145}$$

Then,

$$\mathbb{E}[|I_{\mathcal{T}_P(D)}(f) - \mathbb{E}_Q[f]|] \leq b \|f\|_Q. \tag{146}$$

Proof. Let $a = 2^{K+t/2}$, and define

$$h(x) = f(x) \mathbb{1}[r(x) \leq a], \tag{147}$$

where $\mathbb{1}[\cdot]$ is an indicator function. Let $\phi, \psi \in L^2(Q)$, and define

$$\langle \phi, \psi \rangle_Q \stackrel{\text{def}}{=} \int_{\Omega} \phi \psi dQ. \tag{148}$$

By the triangle inequality,

$$\begin{aligned}
 |I_D(f) - \mathbb{E}_Q[f]| &\leq \\
 &|\mathbb{E}_Q[f] - \mathbb{E}_Q[h]| + |\mathcal{I}_D(f) - \mathcal{I}_D(h)| + |I_D(h) - \mathbb{E}_Q[h]|.
 \end{aligned} \tag{149}$$

We will now proceed to bound each term on the right hand side of the inequality. First,

$$\begin{aligned}
 |\mathbb{E}_Q[f] - \mathbb{E}_Q[h]| &= |\mathbb{E}_Q[f - h]| \\
 &\leq \mathbb{E}_Q[|f| \mathbb{1}[r > a]] \\
 &= \langle |f|, \mathbb{1}[r > a] \rangle_Q \\
 &\leq \|f\|_Q \sqrt{\mathbb{P}[r(Y) > a]},
 \end{aligned} \tag{150}$$

where the first inequality follows by Jensen's inequality and the second follows by Cauchy-Schwarz. Next,

$$\begin{aligned}
 \mathbb{E}[|\mathcal{I}_D(f) - \mathcal{I}_D(h)|] &= \mathbb{E}[|\mathcal{I}_D(f - h)|] \\
 &\leq \mathbb{E}[\mathcal{I}_D(|f - h|)] \\
 &= \mathbb{E}_Q[|f| \mathbb{1}[r > a]] \\
 &= \langle |f|, \mathbb{1}[r > a] \rangle_Q \\
 &\leq \|f\|_Q \sqrt{\mathbb{P}[r(Y) > a]},
 \end{aligned} \tag{151}$$

where the first inequality holds by Jensen's inequality, the second equality follows by Lemma 13 and the second inequality follows by Cauchy-Schwarz. Finally,

$$\begin{aligned}
 \mathbb{E}[|I_D(h) - \mathbb{E}_Q[h]|]^2 &\leq \\
 &\leq \mathbb{E}[(I_D(h) - \mathbb{E}_Q[h])^2] \\
 &= \mathbb{V}[I_D(h)] \\
 &= \mathbb{V} \left[\frac{1}{N} \sum_{d=1}^D \sum_{\substack{n \in \mathcal{T}_D \\ D_n=d}} h(X_n) \frac{dQ}{dP}(X_n) \right] \\
 &= \frac{1}{N^2} \sum_{d=1}^D \sum_{\substack{n \in \mathcal{T}_D \\ D_n=d}} \mathbb{V}_{P|_{B_n}} \left[h(X_n) \frac{dQ}{dP}(X_n) \right],
 \end{aligned} \tag{152}$$

where the first inequality holds by Jensen's inequality. Now, examining a single variance term in the above sum:

$$\begin{aligned}
 \mathbb{V}_{P|_{B_n}} \left[h(X_n) \frac{dQ}{dP}(X_n) \right] &\leq \\
 &\leq \mathbb{E}_{P|_{B_n}} \left[h(X_n)^2 \frac{dQ}{dP}(X_n)^2 \right] \\
 &= \int_{B_n} h^2(X_n) \frac{dQ}{dP}(X_n)^2 \frac{dP(X_n)}{P(B_n)} \\
 &= 2^{d-1} \int_{B_n} h^2(X_n) r(X_n) dQ(X_n) \\
 &= 2^{d-1} \int_{B_n} f^2(X_n) \mathbb{1}[r(X_n) \leq a] r(X_n) dQ(X_n) \\
 &= a 2^{d-1} \int_{B_n} f^2(X_n) dQ(X_n).
 \end{aligned} \tag{153}$$

Plugging this back into eq. (152), we get

$$\begin{aligned}
 \mathbb{E}[|I_D(h) - \mathbb{E}_Q[h]|]^2 &\leq \\
 &\leq \frac{a}{N^2} 2^{d-1} \sum_{d=1}^D \sum_{\substack{n \in \mathcal{T}_D \\ D_n=d}} \int_{B_n} f^2(X_n) dQ(X_n) \\
 &= \frac{a}{N^2} \sum_{d=1}^D 2^{d-1} \int_{\Omega} f^2(X_n) dQ(X_n) \\
 &= \frac{a \|f\|_Q^2}{N^2} \sum_{d=1}^D 2^{d-1} \\
 &= \frac{a \|f\|_Q^2}{N}.
 \end{aligned} \tag{154}$$

Taking square roots of the very left and very right, we get

$$\mathbb{E}[|I_D(h) - \mathbb{E}_Q[h]|] \leq \|f\|_Q 2^{-t/4} \sqrt{1 + \frac{1}{N}}. \tag{155}$$

Putting the three bounds together gives us the desired result. \square

We are now ready to state our result on the \tilde{Q}_D -expectation

Theorem 5 (Biasedness of DAD* coding). *Let $Q, P, r, f, \mathcal{I}, \tilde{Q}_D, K, D, N$ and Y be defined as in Lemma 14. Define*

$$\delta \stackrel{\text{def}}{=} \left(2^{-t/4} \sqrt{1 + \frac{1}{N}} + 2\sqrt{\mathbb{P}[\log_2 r(Y) > K + t/2]} \right)^{1/2}$$

Then,

$$\mathbb{P} \left[\left| \mathbb{E}_{\tilde{Q}_D}[f] - \mathbb{E}_Q[f] \right| \geq \frac{2\|f\|_Q \delta}{1 - \delta} \right] \leq 2\delta. \tag{156}$$

Proof. The proof is mutatis mutandis the same as the proof of Theorem 1.2 of Chatterjee & Diaconis (2018), as it only relies on eq. (140) and Lemma 14. We simply repeat it here for completeness.

Let $a = 2^{K+t/2}$ and

$$b = \left(2^{-t/4} \sqrt{1 + \frac{1}{N}} + 2\sqrt{\mathbb{P}[\log_2 r(Y) > K + t/2]} \right). \tag{157}$$

Then, by Markov's inequality and Lemma 14, for $\delta \in (0, 1)$ we have

$$\begin{aligned}
 \mathbb{P}[|\mathcal{I}_D(f) - 1| \geq \delta] &\leq \frac{\mathbb{E}[|\mathcal{I}_D(f) - 1|]}{\delta} \\
 &\leq \frac{b}{\delta},
 \end{aligned} \tag{158}$$

and for $\alpha \in (0, 1)$ we get

$$\begin{aligned}
 \mathbb{P}[|\mathcal{I}_D(f) - \mathbb{E}_Q[f]| \geq \alpha] &\leq \frac{\mathbb{E}[|\mathcal{I}_D(f) - \mathbb{E}_Q[f]|]}{\alpha} \\
 &\leq \frac{\|f\|_Q}{\alpha}.
 \end{aligned} \tag{159}$$

Now, if $|\mathcal{I}_D(f) - \mathbb{E}_Q[f]| < \alpha$ and $|\mathcal{I}_D(f) - 1| < \delta$, then

$$\begin{aligned}
 & |\mathbb{E}_{\tilde{Q}_D}[f] - \mathbb{E}_Q[f]| \\
 & \leq \left| \frac{\mathcal{I}_D(f)}{\mathcal{I}_D(1)} - \mathbb{E}_Q[f] \right| \\
 & \leq \frac{|\mathcal{I}_D(f) - \mathbb{E}_Q[f]| + |\mathbb{E}_Q[f]| |1 - \mathcal{I}_D(1)|}{\mathcal{I}_D(1)} \\
 & < \frac{\alpha + |\mathbb{E}_Q[f]| \delta}{1 - \delta}.
 \end{aligned} \tag{160}$$

Finally, setting $\delta = \sqrt{b}$ and $\alpha = \|f\|_Q \delta$ gives the desired result. \square

E. Proof of Lemma 2

Lemma 15. *Let Q and P be the target and proposal distributions passed to A* coding (Algorithm 1). Let H^* be the heap index returned by unrestricted A* coding and H_d^* be the index returned by its depth-limited version with $D_{max} = d$. Then, conditioned on the public random sequence S , we have $H_d^* \leq H^*$. Further, there exists $D \in \mathbb{N}$ such that for all $d > D$ we have $H_d^* = H^*$.*

Proof. Let $n, m \in \mathcal{T}$ be two nodes in the binary tree representation \mathcal{T} of the Gumbel process with base measure P realized by alg. 2 simulated using the public random sequence S , such that $D_m < D_n$. It follows from the definition of heap indexing that $H_m < H_n$. Given S , A* and its depth-limited version search over the same tree \mathcal{T} , with the difference that the depth-limited version only searches $\mathcal{T}_d \subset \mathcal{T}$, the tree truncated after depth $D_{max} = d$. Let

$$\begin{aligned}
 n^* &= \arg \max_{n \in \mathcal{T}} \{G_n + \log r(X_n)\} \\
 n_d^* &= \arg \max_{n \in \mathcal{T}_d} \{G_n + \log r(X_n)\},
 \end{aligned} \tag{161}$$

the nodes from \mathcal{T} returned by unrestricted A* coding and its depth-limited version, respectively. Clearly $D_{n_d^*} \leq d$. Then, we have the following two cases.

Case 1: $d < D_{n^*}$. In this case, we have $D_{n_d^*} < D_{n^*}$, hence $H_d^* < H^*$.

Case 2: $d \geq D_{n^*}$. In this case, depth-limited A* coding finds and returns n^* , and since it is the global maximum, increasing the budget further will not make a difference in the returned node. Thus, for this case we get $H_d^* = H^*$, as required. \square

F. KL Parameterization for Distributions

In this section, we demonstrate how Gaussian and uniform target distributions can be parameterized by their KL divergence to some reference distribution, and give some details on how to implement these parameterizations in practice.

F.1. KL-Mean Parameterization for Gaussians

Given a reference Gaussian distribution $P = \mathcal{N}(\nu, \rho^2)$, we want to parameterize $Q = \mathcal{N}(\mu, \sigma^2)$ such that

$$D_{\text{KL}}[Q||P] = \kappa \text{ and } \sigma < \rho. \tag{162}$$

The condition that $\sigma < \rho$ incorporates the additional inductive bias, that since P in practical applications acts as a prior and Q will be representing a variational approximation to a posterior, the posterior should have less uncertainty than the prior. It is also a condition required by A* coding, as this condition is necessary and sufficient to ensure $D_\infty[Q||P] < \infty$.

The KL divergence from Q to P is

$$D_{\text{KL}}[Q||P] = \log \frac{\rho}{\sigma} + \frac{\sigma^2 + (\mu - \nu)^2}{2\rho^2} - \frac{1}{2} = \kappa. \tag{163}$$

Algorithm 2: Priority queue-based top-down construction of a Gumbel process.

Input : Base distribution P with density p , number of samples to realize K , maximum search depth D_{max} .

Output : Next realization from the Gumbel process $\mathcal{GP}_P = (X_k, G_k)_{k=1}^\infty$.

 $k, D_1, H_1 \leftarrow 1, 1, 1$
 $Q \leftarrow \text{PriorityQueue}$
 $G_1 \sim \text{Gumbel}(0)$
 $X_1 \sim P(\cdot)$
 $\Pi.\text{pushWithPriority}(1, G_1)$
while $!\Pi.\text{empty}()$ **do**
 $i \leftarrow \Pi.\text{popHighest}()$
if $D_i \leq D_{max}$ **then**
 $L, R \leftarrow \text{partition}(B_p, X_i)$
for $C \in \{L, R\}$ **do**
 $k \leftarrow k + 1$
 $B_k \leftarrow C$
 $D_k \leftarrow D_p + 1$

$$H_k \leftarrow \begin{cases} 2H_p & \text{if } C = L \\ 2H_p + 1 & \text{if } C = R \end{cases}$$
 $G_k \sim \text{TruncGumbel}(\log P(B_k), G_p)$
 $X_k \sim P(\cdot)|_{B_k}/P(B_k)$
 $\Pi.\text{pushWithPriority}(k, G_k)$
end
end
yield X_i, G_i, H_i, D_i
end

We observe that the largest value that $|\mu - \nu|$ can take while satisfying this equality and the constraint $0 < \sigma < \rho$, occurs when $\sigma \rightarrow \rho$, in which case we have

$$\frac{(\mu - \nu)^2}{2\rho^2} = \kappa \implies |\mu - \nu| < \rho \sqrt{2\kappa}. \quad (164)$$

Defining $\Delta = (\mu - \nu)/\rho$, we can rearrange eq. (163) to

$$\frac{\sigma^2}{\rho^2} e^{-\sigma^2/\rho^2} = e^{\Delta^2 - 2\kappa - 1}. \quad (165)$$

We can rearrange this equation using the Lambert W function (Lambert, 1758), into the form

$$\sigma^2 = -\rho^2 W\left(-e^{\Delta^2 - 2\kappa - 1}\right). \quad (166)$$

While the Lambert W does not have an expression in terms of elementary functions, it can be estimated numerically (Corless et al., 1996). While a numerical solver for the W function is supported in Tensorflow, we found it computationally faster and numerically stabler method to use a Pade approximant of W in practice (Brezinski, 1994). We make this approximant method available in our code repository.

Implementing a Gaussian IsoKL layer: Note, that the parameters derived above are in a constrained domain. Thus, given $P = \mathcal{N}(\nu, \rho^2)$, we can reparameterize κ and μ to an unconstrained domain as follows:

1. Let α, β be real numbers.
2. Set $\kappa \leftarrow \exp(\alpha)$. This will ensure that $\kappa \geq 0$.
3. Set $\mu \leftarrow \nu + \rho\sqrt{2\kappa} \tanh(\beta)$. This ensures the inequality on $|\mu - \nu|$ in Equation (164) is satisfied.
4. Set $\sigma^2 \leftarrow -\rho^2 W\left(-\exp(\Delta^2 - 2\kappa - 1)\right)$.

F.2. KL-Infinity Divergence Parameterization for Gaussians

Assume now, instead of just controlling the KL, we wish to control the Rényi ∞ divergence as well. Concretely, for a given reference Gaussian distribution $P = \mathcal{N}(\nu, \rho^2)$, we want to parameterize $Q = \mathcal{N}(\mu, \sigma^2)$ such that

$$D_{\text{KL}}[Q\|P] = K \quad \text{and} \quad D_{\infty}[Q\|P] = R. \quad (167)$$

We know, that

$$\begin{aligned} K &= D_{\text{KL}}[Q\|P] = \frac{1}{2} [\mu^2 + \sigma^2 - \log \sigma^2 - 1] \\ R &= \log \sup_{x \in \mathbb{R}} \left\{ \frac{dQ}{dP}(x) \right\} = \frac{\mu^2}{2(1 - \sigma^2)} - \log \sigma. \end{aligned} \quad (168)$$

From these, we get that

$$\begin{aligned} \mu^2 &= 2K - \sigma^2 + \log \sigma^2 + 1 \\ \mu^2 &= 2(1 - \sigma^2)(R + \log \sigma) \end{aligned} \quad (169)$$

Setting these equal to each other

$$\begin{aligned} 2K - \sigma^2 + \log \sigma^2 + 1 &= 2R + \log \sigma^2 - 2\sigma^2 R - \sigma^2 \log \sigma^2 \\ \sigma^2 \log \sigma^2 - \sigma^2 + 2\sigma^2 R &= 2R - 2K - 1 \\ \sigma^2 \log \sigma^2 + \sigma^2(2R - 1) &= A \\ \sigma^2(\log \sigma^2 + B) &= A \\ \sigma^2 \log(\sigma^2 e^B) &= A \\ e^B \sigma^2 \log(\sigma^2 e^B) &= Ae^B \\ e^{\log(\sigma^2 e^B)} \log(\sigma^2 e^B) &= Ae^B \\ \log(\sigma^2 e^B) &= W(Ae^B) \\ \sigma^2 &= e^{W(Ae^B) - B}, \end{aligned} \quad (170)$$

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

Note: This parameterization is only unique up to the sign of the target mean μ due to the symmetry of the Gaussian distribution.

F.3. KL-Mean parameterization for Uniforms

Given a uniform reference distribution $P = \mathcal{U}(\nu, \rho)$ with mean ν and width ρ on the interval $[\nu - \rho/2, \nu + \rho/2]$, we want to parameterize $Q = \mathcal{U}(\mu, \sigma)$ such that

$$D_{\text{KL}}[Q||P] = \kappa. \quad (171)$$

Note, that we must have

$$\nu - \rho/2 \leq \mu - \sigma/2 < \mu + \sigma/2 \leq \nu + \rho/2 \quad (172)$$

to ensure $Q \ll P$. Now, we have

$$\kappa = D_{\text{KL}}[Q||P] = \log \frac{\rho}{\sigma}, \quad (173)$$

from which we get

$$\sigma = \rho \exp(-\kappa). \quad (174)$$

Implementing a Uniform IsoKL layer: Note, that the parameters derived above are in a constrained domain. Thus, given $P = \mathcal{U}(\nu, \rho)$, we can reparameterize κ and μ to an unconstrained domain as follows:

1. Let α, β be real numbers.
2. Set $\kappa \leftarrow \exp(\alpha)$. This will ensure that $\kappa \geq 0$.
3. Set $\sigma \leftarrow \rho \exp(-\kappa)$.
4. Set $\mu \leftarrow \nu + \frac{\rho - \sigma}{2} \tanh(\beta)$. This will ensure $Q \ll P$.