

Schemes for Bidirectional Modeling of Discrete Stationary Sources

Jiming Yu, *Student Member, IEEE*, and Sergio Verdú, *Fellow, IEEE*

Abstract—We develop adaptive schemes for bidirectional modeling of unknown discrete stationary sources. These algorithms can be applied to statistical inference problems such as noncausal universal discrete denoising that exploit bidirectional dependencies. Efficient algorithms for constructing those models are developed and we compare their performance to that of the DUDE algorithm for universal discrete denoising.

Index Terms—Bidirectional modeling, discrete stationary sources, universal algorithms, universal discrete denoising.

I. INTRODUCTION

A. Motivation

BIDIRECTIONAL models are concerned with estimating the marginal conditional distribution of the current symbol given the rest of the data, in contrast to unidirectional models that care about the conditional distribution of the current symbol given the past symbols. Noncausal universal discrete denoising [1] is a typical application of bidirectional models. The main application of sequentially available unidirectional models such as context tree weighting [2], [3], is arithmetic coding based data compression, where the decompressor maintains at each point the same model as the compressor.

To be concrete, we explain why bidirectional models are needed for noncausal discrete denoising, which is a major application of bidirectional modeling. Denoising is the procedure of recovering the uncoded input to a channel with as high a fidelity as possible, by only observing the output of the channel. Universal discrete denoising [1] deals with discrete input/output channels without knowledge of the input distribution. When the observations up to the previous time instant are available for recovering the input symbol at the current time instant, the problem is called universal prediction in a noisy environment [4], [5]. When the observations up to the current time instant are available for recovering the input symbol at the current time instant, the problem is called (causal) universal discrete filtering [6]. In contrast to the above two cases, which can be categorized as *causal* universal discrete denoising, the case when all observations are available to recover the

input symbol at any time instant is called *noncausal* universal discrete denoising, which is the main application of this paper. Another relevant application of the algorithms presented here is in universal channel decoding of uncompressed sources [7].

B. Unidirectional Modeling

Unidirectional models estimate the unidirectional conditional distributions $\mathbf{P}_{Z_j|b_1^{j-1}}$ from a realization z^n (either sequentially available or not) for a stationary process $Z = (Z_t)_{t \in \mathbb{Z}}$ on an alphabet \mathcal{B} of size $M' < \infty$, where the a th component of $\mathbf{P}_{Z_j|b_1^{j-1}} \in [0, 1]^{M'}$ is

$$\mathbb{P}\{Z_j = a | Z_1^{j-1} = b_1^{j-1}\}, \quad \forall a \in \mathcal{B}, b_1^{j-1} \in \mathcal{B}^{j-1}$$

for $\forall j = 1, 2, \dots, n$. Because of its importance in data compression, unidirectional modeling has been subject to extensive research, cf. [8]–[12] and references therein. Various context tree methods are widely used in those models.

A quite different approach to modeling the source is the celebrated context-tree weighting (CTW) algorithm [2], [3]. Although it is still based on constructing a context tree, its main principle is to use a *weighted mixture* of all submodels, in which Krichevsky–Trofimov (KT) estimators [13] for the empirical counts are directly used. Here submodels refer to either models with orders not exceeding the given memory length constraint D in the basic CTW [2], or equivalence classes of models with all possible orders in the extended CTW [3]. The equivalence of two models means they give the same distributions. Thus, by the uniform bound [2, Sec. V, eq. (11)] derived for the KT estimator [13], though the CTW mixture is strictly inferior compared to the best submodel, it is quite close to the best submodel asymptotically. CTW essentially specifies a unidirectional model estimating

$$\mathbb{P}\{Z_1^j = z_1^j\}$$

on the basis of $z_1^j, j = 1, 2, \dots, n$, for the given realization z_1^n of Z_1^n . With some modifications, this unidirectional model can be turned into a partial bidirectional model, which will be discussed in Section II-C. Recently, the consistency result [14] and its efficient implementation as a generalization of the context-tree maximizing (CTM) algorithm [15] (which is quite related to CTW) have shown great potential of the context-based unidirectional modeling approach for stationary ergodic processes via various information criteria.

C. DUDE

A stationary process $X = (X_t)_{t \in \mathbb{Z}}$ with finite alphabet $\mathcal{A} = \{1, 2, \dots, M\}$ goes through a discrete memoryless channel (DMC) with a channel transition probability matrix

Manuscript received May 4, 2005; revised June 21, 2006. This work was partially supported by the National Science Foundation under Grant CCR-0312839. The material in this paper was presented in part at the Conference on Information Sciences and Systems, Baltimore, MD, March 2005.

The authors are with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: jiminyu@princeton.edu; verdu@princeton.edu).

Communicated by P. L. Bartlett, Associate Editor for Pattern Recognition, Statistical Learning and Inference.

Digital Object Identifier 10.1109/TIT.2006.883626

$\mathbf{\Pi}$ and a noisy stationary output process $Z = (Z_t)_{t \in \mathbb{Z}}$ with alphabet $\mathcal{B} = \{1, 2, \dots, M'\}$ is observed, where

$$\mathbf{\Pi}(a, b) \triangleq \mathbb{P}\{Z_t = b | X_t = a\}, \quad \forall t \in \mathbb{Z}, a \in \mathcal{A}, b \in \mathcal{B}.$$

A fixed loss function $\Lambda : \mathcal{A} \times \hat{\mathcal{A}} \rightarrow [0, \infty)$ such that $\Lambda(a, \hat{a})$ is the loss of estimating $a \in \mathcal{A}$ by $\hat{a} \in \hat{\mathcal{A}} \triangleq \{1, 2, \dots, \hat{M}\}$, is represented by an $M \times \hat{M}$ matrix $\mathbf{\Lambda} = \{\Lambda(a, b)\}_{M \times \hat{M}}$. $\mathbf{\Lambda}$ induces an average cumulative loss $\frac{1}{n} \sum_{j=1}^n \Lambda(X_j, \hat{X}_j)$ by giving \hat{X}^n as the output of the denoiser, for finite-time input X^n and finite-time observations Z^n . The denoiser outputs an estimate \hat{X}^n of X^n with as small a loss as possible, by observing Z^n and knowing the channel matrix $\mathbf{\Pi}$ and the loss matrix $\mathbf{\Lambda}$.

Under the assumption that the channel matrix has full row rank, the input distribution is uniquely determined by the output distribution and the channel matrix. For any realization (x^n, z^n) , this leads to the following optimal Bayesian denoiser [1]:

$$\hat{X}_{\text{opt}}^n(z^n)[j] = \arg \min_{\hat{x} \in \hat{\mathcal{A}}} \mathbf{P}_{Z_j | z_1^{j-1}, z_{j+1}^n}^T \mathbf{\Pi}^T (\mathbf{\Pi} \mathbf{\Pi}^T)^{-1} [\boldsymbol{\lambda}_{\hat{x}} \odot \boldsymbol{\pi}_{z_j}] \quad (1)$$

where $\hat{X}_{\text{opt}}^n : \mathcal{B}^n \rightarrow \hat{\mathcal{A}}^n$ is the denoiser in vector form, i.e., $\hat{X}_{\text{opt}}^n(z^n)[j] = \hat{x}_j$ is its j th component representing the estimate of x_j based on the observations z^n , $\mathbf{P}_{Z_j | z_1^{j-1}, z_{j+1}^n} \in [0, 1]^{M'}$ with b th component specified as

$$\mathbb{P}\{Z_j = b | Z_1^{j-1} = z_1^{j-1}, Z_{j+1}^n = z_{j+1}^n\}, \quad \forall b \in \mathcal{B}$$

and $\boldsymbol{\lambda}_{\hat{x}}$ is the \hat{x} th column of $\mathbf{\Lambda}$, $\boldsymbol{\pi}_{z_j}$ is the z_j th column of $\mathbf{\Pi}$, and for any strictly positive integer d

$$u \odot v \triangleq (u_1 v_1, u_2 v_2, \dots, u_d v_d)^T, \quad \forall u, v \in \mathbb{R}^d.$$

When the input distribution is known, the output distribution is completely determined from the input distribution and the channel matrix $\mathbf{\Pi}$, and thus, $\mathbf{P}_{Z_j | z_1^{j-1}, z_{j+1}^n}$ can be obtained. In the universal case where input distribution is unknown, the only unknown quantities in (1) are the bidirectional conditional distributions

$$\mathbf{P}_{Z_j | z_1^{j-1}, z_{j+1}^n}, \quad j = 1, 2, \dots, n. \quad (2)$$

Thus, through (1), universal discrete denoising (at least under the full row rank condition) reduces to bidirectional modeling (cf. (2)) of the output process based on the observation of a realization. Discrete Universal DEnoiser (DUDE) [1] uses static symmetric bidirectional contexts to estimate $\mathbf{P}_{Z_j | z_1^{j-1}, z_{j+1}^n}$, namely, the approximation

$$\mathbf{P}_{Z_j | z_1^{j-1}, z_{j+1}^n} \approx \mathbf{P}_{Z_j | z_{j-m}^{j-1}, z_{j+1}^{j+m}} \quad (3)$$

for some fixed integer m , where $\mathbf{P}_{Z_j | z_{j-m}^{j-1}, z_{j+1}^{j+m}} \in [0, 1]^{M'}$ with the b th component specified as

$$\mathbb{P}\{Z_j = b | Z_{j-m}^{j-1} = z_{j-m}^{j-1}, Z_{j+1}^{j+m} = z_{j+1}^{j+m}\}, \quad \forall b \in \mathcal{B}.$$

Furthermore, DUDE [1] replaces all bidirectional conditional distributions $\mathbf{P}_{Z_j | z_{j-m}^{j-1}, z_{j+1}^{j+m}}$ by their corresponding empirical distributions.

Fundamental to the problem of denoising (with other applications such as [16]) is the estimation of the marginal conditional distribution of the input X_j given the output being $Z_1^n = z_1^n$ [1]

$$\mathbf{P}_{X_j | z_1^n} = \frac{\pi_{z_j} \odot [(\mathbf{\Pi} \mathbf{\Pi}^T)^{-1} \mathbf{\Pi} \mathbf{P}_{Z_j | z_1^{j-1}, z_{j+1}^n}]}{\mathbf{P}_{Z_j | z_1^{j-1}, z_{j+1}^n}[z_j]} \quad (4)$$

where the right-hand side of (4) can be estimated via estimates of $\mathbf{P}_{Z_j | z_1^{j-1}, z_{j+1}^n}$ from a bidirectional model for process Z . Note that (1) is derived from (4) by Bayesian estimation [1]. While the known channel case is solved in [1], the case of a binary symmetric channel with unknown crossover probability is studied in [17], [18] under a minimax criterion. But whether the channel is known or not, our algorithms always apply to the bidirectional modeling of the noisy channel output. When the input process is a Markov chain, an alternative to compute the marginal conditional input distribution via (4) is backward-forward dynamic programming. Hidden Markov modeling (HMM) tools [19], [20] are popular in order to estimate both the channel and the input Markov chain transition probability matrix when they are unknown. More recently, HMM has been used to universally filter noisy outputs of a known DMC [21].

D. Bidirectional Modeling Beyond DUDE

If the noisy observations were from an m th-order Markov chain, the approximation in (3) would be exact (assuming $m < j \leq n - m$). But in general, this static symmetric bidirectional context-based approximation is not necessarily the best we can do. After all, Z_j may depend on different number of symbols for different j in an asymmetric way, i.e.,

$$\mathbf{P}_{Z_j | z_1^{j-1}, z_{j+1}^n} \approx \mathbf{P}_{Z_j | z_{j-m_1}^{j-1}, z_{j+1}^{j+m_2}}$$

for some m_1, m_2 depending on both z_1^n and j , and m_1 is not necessarily equal to m_2 , where $\mathbf{P}_{Z_j | z_{j-m_1}^{j-1}, z_{j+1}^{j+m_2}} \in [0, 1]^{M'}$ with the b th component specified as

$$\mathbb{P}\{Z_j = b | Z_{j-m_1}^{j-1} = z_{j-m_1}^{j-1}, Z_{j+1}^{j+m_2} = z_{j+1}^{j+m_2}\}, \forall b \in \mathcal{B}.$$

Adaptive bidirectional models that incorporate this dynamic asymmetric bidirectional context-based approximation are considered in this paper. Note that the static symmetric bidirectional context approach taken in DUDE leads to performance which is optimized by a finite (and unknown) context length. Not only are short lengths suboptimal because they do not fully exploit the source memory but because of the finiteness of the reservoir of data also long context lengths give rise to unreliable empirical bidirectional conditional distributions for a given data length n .

Independently of the work reported here (and in conference version in [22]), Ordentlich *et al.* [23] (see also [24]) have solved the related problem of finding the bidirectional context tree that minimizes a given cost criterion. By estimating the unknown true denoising performance, bidirectional context trees are constructed in [23] so as to minimize the estimated losses incurred by the denoisers that are specified by those bidirectional context trees. There are many conceptual similarities between [23] and the BCT scheme in this paper (cf. Section II-D), e.g., both try

to model the source by identifying bidirectional contexts with tree structures. There are also fundamental differences in the methodologies and the tree structures between [23] and BCT. In terms of methodology, [23] assumes a given cost criterion depending on bidirectional contexts, according to which they use dynamic programming to optimize their tree; BCT directly aims at the probabilistic bidirectional modeling of stationary sources, and uses a divergence criterion to optimize its own tree. In terms of tree structure, [23] allows the bidirectional context corresponding to a child node to be exactly unit length longer than that corresponding to its father node in one direction; whereas as we discuss later, our BCT tree structure allows the bidirectional context corresponding to a child node to be unit length(s) longer than that corresponding to its father node in either direction or in both directions.

E. Organization

This paper is organized as follows. Section II introduces six adaptive bidirectional models that estimate bidirectional conditional distributions. Section III gives efficient algorithms to construct the six adaptive bidirectional models. Section IV gives comparative experimental results of various universal discrete denoisers based on different models.

II. ADAPTIVE BIDIRECTIONAL MODELS FOR DISCRETE STATIONARY SOURCES

In this paper, $z_1^n \in \mathcal{B}^n$ is fixed to be a given finite realization of a stationary process Z based on which we estimate the unidirectional or bidirectional models.

A. Unidirectional Modeling: Variable-Length Markov Chains (VLMCs)

A powerful unidirectional model for discrete stationary sources, the so-called Variable-Length Markov Chain (VLMC), has been developed in [8]–[12]. This model can be constructed both sequentially [8], [9] and nonsequentially [10]–[12], the former of which can be applied directly to lossless data compression based on arithmetic coding. The latter approach well suits our purpose of *noncausal* universal discrete denoising, and it is useful in other applications such as two-pass (or nonsequential) lossless data compression algorithms [25]. The main idea is to model the discrete stationary source $Z = (Z_t)_{t \in \mathbb{Z}}$ as a Markov process with order that depends on the history of the realization

$$\mathbb{P}\{Z_j = a | Z_1^{j-1} = b_1^{j-1}\} = \mathbb{P}\{Z_j = a | Z_{j-k_n}^{j-1} = b_{j-k_n}^{j-1}\} \quad (5)$$

for any $a \in \mathcal{B}$, $b_1^{j-1} \in \mathcal{B}^{j-1}$ and some $k_n = k_n(b_1^{j-1}) \leq j-1$ which can be estimated from a realization z^n , $j = 1, 2, \dots, n$. Here $b_{j-k_n}^{j-1}$ is called the unidirectional context of Z_j . The estimation is accomplished by a context tree model [8]–[12], whose construction requires the normalized divergence of unidirectional conditional distribution estimates from all leaf nodes with respect to that of their father nodes be larger than some preselected parameter. This algorithm has been shown to consistently estimate the unidirectional context tree and corresponding unidirectional conditional distributions from the nodes in this tree [8]–[12]. The examples of binary sources with memory which

give rise to first-order Markov processes when corrupted by binary symmetric channels (BSCs) [26], [27] further justify our application of VLMCs to denoising.

B. Four Adaptive Bidirectional Models Based on VLMCs

In general, unidirectional models are not directly applicable to bidirectional modeling. But there are some ways that bidirectional dependencies can be reflected by unidirectional models. In this subsection, we describe four methods for constructing bidirectional models by utilizing unidirectional models (e.g., unidirectional models running forwards and backwards).

1) *Backward-Forward Product (BFP)*: An extremely simple way to estimate $\mathbb{P}\{Z_j = a | Z_1^{j-1} = b_1^{j-1}, Z_{j+1}^n = b_{j+1}^n\}$ from the unidirectional conditional probabilities is¹

$$\begin{aligned} & \tilde{\mathbb{P}}\{Z_j = a | Z_1^{j-1} = b_1^{j-1}, Z_{j+1}^n = b_{j+1}^n\} \\ & \propto \tilde{\mathbb{P}}\{Z_j = a | Z_1^{j-1} = b_1^{j-1}\} \times \tilde{\mathbb{P}}\{Z_j = a | Z_{j+1}^n = b_{j+1}^n\} \quad (6) \end{aligned}$$

for any $a \in \mathcal{B}$, $b_1^{j-1} \in \mathcal{B}^{j-1}$, $b_{j+1}^n \in \mathcal{B}^{n-j}$, $j = 1, 2, \dots, n$, where \propto means “proportional to,” and $\tilde{\mathbb{P}}$ stands for the estimated probability law. Once equipped with any unidirectional model (e.g., the one mentioned in Section II-A), the first term in the right-hand side of (6) can be estimated from data z^n , and the second term can be estimated from time-reversed data $z_*^n \triangleq (z_{n-i+1})_{i=1}^n$. Then (6) can be used in specifying a corresponding bidirectional model. This *ad hoc* approximation obviously has the merit that it is straightforward to implement once we have a computationally efficient unidirectional model. A first-order Markov chain with equiprobable one-dimensional marginal distribution on \mathcal{B} satisfies (6), by the more general formula (16) in Section II-B2 below, which is exact under the $(2m+1)$ th-order Markov assumption. In this case, BFP gives the exact bidirectional conditional distribution up to a normalizing constant.

2) *Generalized Markov (GM) Scheme*: We make the assumption that there exists an integer $m < \frac{1}{2}n$ such that given any Z_{j-m}^{j+m} , the past and the future are conditionally independent, that is, Z_1^{j-m-1} and Z_{j+m+1}^n are conditionally independent given Z_{j-m}^{j+m} , $\forall j = m+1, m+2, \dots, n-m$. Under that assumption which is equivalent to the assumption that Z is $(2m+1)$ th-order Markovian, we first see that

$$\mathbb{P}\left\{Z_{j-m}^{j+m} = b_{j-m}^{j-1} a b_{j+1}^{j+m} \mid Z_1^{j-m-1} = b_1^{j-m-1}, Z_{j+m+1}^n = b_{j+m+1}^n\right\} \quad (7)$$

is the product of

$$\frac{\mathbb{P}\{Z_{j-m}^{j+m} = b_{j-m}^{j-1} a b_{j+1}^{j+m}\}}{\mathbb{P}\{Z_1^{j-m-1} = b_1^{j-m-1}, Z_{j+m+1}^n = b_{j+m+1}^n\}} \quad (8)$$

and

$$\mathbb{P}\left\{Z_1^{j-m-1} = b_1^{j-m-1}, Z_{j+m+1}^n = b_{j+m+1}^n \mid Z_{j-m}^{j+m} = b_{j-m}^{j-1} a b_{j+1}^{j+m}\right\}. \quad (9)$$

¹In many applications (e.g., universal discrete denoising) unnormalized conditional marginals are sufficient.

That is, (7)=(8) × (9). Now by our assumption, (9) can be decomposed as

$$\mathbb{P}\{Z_1^{j-m-1} = b_1^{j-m-1} | Z_{j-m}^{j+m} = b_{j-m}^{j-1} a b_{j+1}^{j+m}\} \\ \times \mathbb{P}\{Z_{j+m+1}^n = b_{j+m+1}^n | Z_{j-m}^{j+m} = b_{j-m}^{j-1} a b_{j+1}^{j+m}\} \quad (10)$$

which can be written as

$$\frac{\mathbb{P}\{Z_{j-m}^{j+m} = b_{j-m}^{j-1} a b_{j+1}^{j+m} | Z_1^{j-m-1} = b_1^{j-m-1}\}}{\mathbb{P}\{Z_{j-m}^{j+m} = b_{j-m}^{j-1} a b_{j+1}^{j+m}\}} \\ \times \frac{\mathbb{P}\{Z_{j-m}^{j+m} = b_{j-m}^{j-1} a b_{j+1}^{j+m} | Z_{j+m+1}^n = b_{j+m+1}^n\}}{\mathbb{P}\{Z_{j-m}^{j+m} = b_{j-m}^{j-1} a b_{j+1}^{j+m}\}} \\ \times \mathbb{P}\{Z_1^{j-m-1} = b_1^{j-m-1}\} \mathbb{P}\{Z_{j+m+1}^n = b_{j+m+1}^n\}. \quad (11)$$

So (7)=(8) × (9)=(8) × (10)=(8) × (11) becomes

$$(7) = \frac{\mathbb{P}\{Z_{j-m}^{j+m} = b_{j-m}^{j-1} a b_{j+1}^{j+m} | Z_1^{j-m-1} = b_1^{j-m-1}\}}{\mathbb{P}\{Z_1^{j-m-1} = b_1^{j-m-1}, Z_{j+m+1}^n = b_{j+m+1}^n\}} \\ \times \frac{\mathbb{P}\{Z_{j-m}^{j+m} = b_{j-m}^{j-1} a b_{j+1}^{j+m} | Z_{j+m+1}^n = b_{j+m+1}^n\}}{\mathbb{P}\{Z_{j-m}^{j+m} = b_{j-m}^{j-1} a b_{j+1}^{j+m}\}} \\ \times \mathbb{P}\{Z_{j+m+1}^n = b_{j+m+1}^n\} \mathbb{P}\{Z_1^{j-m-1} = b_1^{j-m-1}\}. \quad (12)$$

Let

$$f^*(b_1^{j-1}, b_{j+1}^n) \\ \triangleq \mathbb{P}\left\{ Z_{j-m}^{j-1} = b_{j-m}^{j-1}, Z_{j+1}^{j+m} = b_{j+1}^{j+m} \right. \\ \left. \left| Z_1^{j-m-1} = b_1^{j-m-1}, Z_{j+m+1}^n = b_{j+m+1}^n \right. \right\} \\ = \sum_{c \in \mathcal{B}} \left[\frac{\mathbb{P}\{Z_{j-m}^{j+m} = b_{j-m}^{j-1} c b_{j+1}^{j+m} | Z_1^{j-m-1} = b_1^{j-m-1}\}}{\mathbb{P}\{Z_1^{j-m-1} = b_1^{j-m-1}, Z_{j+m+1}^n = b_{j+m+1}^n\}} \right. \\ \times \frac{\mathbb{P}\{Z_{j-m}^{j+m} = b_{j-m}^{j-1} c b_{j+1}^{j+m} | Z_{j+m+1}^n = b_{j+m+1}^n\}}{\mathbb{P}\{Z_{j-m}^{j+m} = b_{j-m}^{j-1} c b_{j+1}^{j+m}\}} \\ \left. \times \mathbb{P}\{Z_{j+m+1}^n = b_{j+m+1}^n\} \mathbb{P}\{Z_1^{j-m-1} = b_1^{j-m-1}\} \right] \quad (13)$$

which is the sum of (7)=(12) over all $a \in \mathcal{B}$. By noticing the common factors in (12) and $f^*(b_1^{j-1}, b_{j+1}^n)$, we have

$$\mathbb{P}\{Z_j = a | Z_1^{j-1} = b_1^{j-1}, Z_{j+1}^n = b_{j+1}^n\} = \\ \frac{\mathbb{P}\{Z_{j-m}^{j+m} = b_{j-m}^{j-1} a b_{j+1}^{j+m} | Z_1^{j-m-1} = b_1^{j-m-1}\}}{f(b_1^{j-1}, b_{j+1}^n) \mathbb{P}\{Z_{j-m}^{j+m} = b_{j-m}^{j-1} a b_{j+1}^{j+m}\}} \\ \times \mathbb{P}\{Z_{j-m}^{j+m} = b_{j-m}^{j-1} a b_{j+1}^{j+m} | Z_{j+m+1}^n = b_{j+m+1}^n\} \quad (14)$$

where $f(b_1^{j-1}, b_{j+1}^n)$ is defined as

$$\sum_{c \in \mathcal{B}} \left[\mathbb{P}\{Z_{j-m}^{j+m} = b_{j-m}^{j-1} c b_{j+1}^{j+m} | Z_{j+m+1}^n = b_{j+m+1}^n\} \right. \\ \left. \times \frac{\mathbb{P}\{Z_{j-m}^{j+m} = b_{j-m}^{j-1} c b_{j+1}^{j+m} | Z_1^{j-m-1} = b_1^{j-m-1}\}}{\mathbb{P}\{Z_{j-m}^{j+m} = b_{j-m}^{j-1} c b_{j+1}^{j+m}\}} \right] \quad (15)$$

which is a normalizing constant that does not depend on a . So we directly use

$$\mathbb{P}\{Z_j = a | Z_1^{j-1} = b_1^{j-1}, Z_{j+1}^n = b_{j+1}^n\} \propto \\ \frac{\mathbb{P}\{Z_{j-m}^{j+m} = b_{j-m}^{j-1} a b_{j+1}^{j+m} | Z_1^{j-m-1} = b_1^{j-m-1}\}}{\mathbb{P}\{Z_{j-m}^{j+m} = b_{j-m}^{j-1} a b_{j+1}^{j+m}\}} \\ \times \mathbb{P}\{Z_{j-m}^{j+m} = b_{j-m}^{j-1} a b_{j+1}^{j+m} | Z_{j+m+1}^n = b_{j+m+1}^n\} \quad (16)$$

in many applications by (14).

From a realization z^n , DUDE [1] gives estimates for (i.e., empirical distributions for)

$$\mathbb{P}\{Z_{j-m}^{j+m} = b_{j-m}^{j-1} a b_{j+1}^{j+m}\}, \quad m+1 \leq j \leq n-m. \quad (17)$$

And furthermore we only use (14) or (16) when $b_{j-m}^{j-1} a b_{j+1}^{j+m}$ actually has appeared in the sequence z^n , otherwise, we let the estimate for $\mathbb{P}\{Z_j = a | Z_1^{j-1} = b_1^{j-1}, Z_{j+1}^n = b_{j+1}^n\}$ be 0, since the true value is indeed 0 if $\mathbb{P}\{Z_{j-m}^{j+m} = b_{j-m}^{j-1} a b_{j+1}^{j+m}\} = 0$.

Now, by the definition of conditional probabilities, we have

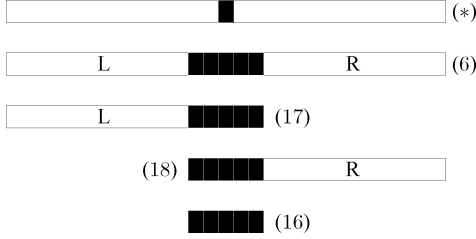
$$\mathbb{P}\{Z_{j-m}^{j+m} = b_{j-m}^{j-1} a b_{j+1}^{j+m} | Z_1^{j-m-1} = b_1^{j-m-1}\} \\ = \mathbb{P}\{Z_j = a | Z_1^{j-1} = b_1^{j-1}\} \mathbb{P}\{Z_{j+1} = b_{j+1} | Z_1^j = b_1^{j-1} a\} \\ \times \prod_{t=j+2}^{j+m} \mathbb{P}\{Z_t = b_t | Z_1^{t-1} = b_1^{t-1} a b_{j+1}^{j+1}\} \\ \times \prod_{t=j-m}^{j-1} \mathbb{P}\{Z_t = b_t | Z_1^{t-1} = b_1^{t-1}\} \quad (18)$$

and

$$\mathbb{P}\{Z_{j-m}^{j+m} = b_{j-m}^{j-1} a b_{j+1}^{j+m} | Z_{j+m+1}^n = b_{j+m+1}^n\} \\ = \mathbb{P}\{Z_{j-1} = b_{j-1} | Z_j^n = a b_{j+1}^n\} \mathbb{P}\{Z_j = a | Z_{j+1}^n = b_{j+1}^n\} \\ \times \prod_{t=j-m}^{j-2} \mathbb{P}\{Z_t = b_t | Z_{t+1}^n = b_{t+1}^n a b_{j+1}^{j+1}\} \\ \times \prod_{t=j+1}^{j+m} \mathbb{P}\{Z_t = b_t | Z_{t+1}^n = b_{t+1}^n\}. \quad (19)$$

Note that terms in (18) can be estimated by a unidirectional model (e.g., the one mentioned in Section II-A) for data z^n , and terms in (19) can be estimated by a unidirectional model for time-reversed data $z_*^n \triangleq (z_{n-j+1})_{j=1}^n$. Plugging estimates for (17), (18), and (19) into (14) or (16) for $j = m+1, m+2, \dots, n-m$, and assigning arbitrary values to the conditional probabilities for other values of j : $j = 1, 2, \dots, m, n-m+1, n-m+2, \dots, n$, we can get a bidirectional model under the above assumption.

The idea of GM can be illustrated by Fig. 1. The bars of this figure represent conditional distributions of the quantities in black given the quantities in white. The left-hand side of (16) is represented by (*) which should be interpreted as the conditional distribution of a single symbol given the contexts on two sides. First, expand the single symbol in (*) to encompass m of its neighbors on both left and right as the second bar (corresponding to (7)) indicates, and denote the remaining conditioning blocks by “L” and “R.” Then we decompose this expanded version into two bars; the third bar representing the conditional distribution of the black block given the “L” block,

Fig. 1. $(*) \propto (17) \times (18)/(16)$.

corresponding to (18); the fourth bar representing the conditional distribution of the black block given the “R” block, corresponding to (19). By multiplying (18) and (19), we have included the black block (17) twice. Thus, (16) simply means $(*) \propto (18) \times (19)/(17)$.

Our approach here is different from directly estimating transition probabilities for the $(2m+1)$ th-order Markov approximation of the original process Z , which obviously is quite inaccurate either when m is big (data length n is then relatively small) or when m is small (cannot capture long memory), and is computationally intensive because of the large state space and the n -term factorization of the joint distribution of Z^n that is proportional to the desired bidirectional conditional distribution. However, we use the unidirectional model VLMC to estimate the “forward” and “backward” unidirectional conditional distributions (18), (19) in a parsimonious way, capitalizing on the constant number of terms on the right-hand side in (14) or (16). This justifies the name “generalized Markov.”

3) *Forward and Backward One-Sided Generalized Markov (f-OGM and b-OGM) Schemes*: The following is straightforward ($1 \leq j \leq n-2$):

$$\begin{aligned} & \mathbb{P}\{Z_j = a | Z_1^{j-1} = b_1^{j-1}, Z_{j+1}^n = b_{j+1}^n\} \\ &= \frac{\mathbb{P}\{Z_j = a, Z_{j+1}^n = b_{j+1}^n | Z_1^{j-1} = b_1^{j-1}\}}{\mathbb{P}\{Z_{j+1}^n = b_{j+1}^n | Z_1^{j-1} = b_1^{j-1}\}} \\ &= \frac{\mathbb{P}\{Z_j = a | Z_1^{j-1} = b_1^{j-1}\}}{\mathbb{P}\{Z_{j+1}^n = b_{j+1}^n | Z_1^{j-1} = b_1^{j-1}\}} \\ & \quad \times \mathbb{P}\{Z_{j+1} = b_{j+1} | Z_1^j = b_1^{j-1} a\} \\ & \quad \times \prod_{t=j+2}^n \mathbb{P}\{Z_t = b_t | Z_1^{t-1} = b_1^{j-1} a b_{j+1}^{t-1}\}. \end{aligned}$$

That is, for $1 \leq j \leq n-2$

$$\begin{aligned} & \mathbb{P}\{Z_j = a | Z_1^{j-1} = b_1^{j-1}, Z_{j+1}^n = b_{j+1}^n\} \\ & \propto \mathbb{P}\{Z_j = a | Z_1^{j-1} = b_1^{j-1}\} \mathbb{P}\{Z_{j+1} = b_{j+1} | Z_1^j = b_1^{j-1} a\} \\ & \quad \times \prod_{t=j+2}^n \mathbb{P}\{Z_t = b_t | Z_1^{t-1} = b_1^{j-1} a b_{j+1}^{t-1}\} \end{aligned} \quad (20)$$

with the normalizing constant (not depending on a) being the reciprocal of $\mathbb{P}\{Z_{j+1}^n = b_{j+1}^n | Z_1^{j-1} = b_1^{j-1}\}$. Similarly, from another direction, we can get ($3 \leq j \leq n$)

$$\begin{aligned} & \mathbb{P}\{Z_j = a | Z_1^{j-1} = b_1^{j-1}, Z_{j+1}^n = b_{j+1}^n\} \\ & \propto \mathbb{P}\{Z_j = a | Z_{j+1}^n = b_{j+1}^n\} \mathbb{P}\{Z_{j-1} = b_{j-1} | Z_j^n = a b_{j+1}^n\} \\ & \quad \times \prod_{t=1}^{j-2} \mathbb{P}\{Z_t = b_t | Z_{t+1}^n = b_{t+1}^{j-1} a b_{j+1}^n\} \end{aligned} \quad (21)$$

with the normalizing constant (not depending on a) being the reciprocal of $\mathbb{P}\{Z_1^{j-1} = b_1^{j-1} | Z_{j+1}^n = b_{j+1}^n\}$.

From (20) and (21), we can use any unidirectional model (e.g., the one mentioned in Section II-A) to estimate the desired bidirectional conditional distributions. But the problem is that the number of multiplications used in those two estimators is quite large. Even assuming that estimating all the unidirectional conditional distributions involved in either (20) or (21) has overall constant time complexity (which underestimates the time complexity, see Section IV-A for details), the total time complexity incurred in an application like denoising would be $O(n^2)$ because of the $(n-j+1)$ -term or the j -term product for each j in (20) or (21). To circumvent this, it is necessary to reduce the number of terms. Assume the process Z is a k th-order Markov process, we first show the time-reversed sequence of random variables $Z_*^n \triangleq (Z_{n-t+1})_{t=1}^n$ is also k th-order Markov: for any $1 \leq t \leq n-k$,

$$\begin{aligned} & \mathbb{P}\{Z_t = b_t | Z_{t+1}^n = b_{t+1}^n\} \\ &= \frac{\mathbb{P}\{Z_t = b_t\} \mathbb{P}\{Z_{t+1}^n = b_{t+1}^n\}}{\mathbb{P}\{Z_{t+1}^n = b_{t+1}^n\}} \\ &= \frac{\mathbb{P}\{Z_t = b_t\} \prod_{i=t+1}^{t+k} \mathbb{P}\{Z_i = b_i | Z_t^{i-1} = b_t^{i-1}\}}{\mathbb{P}\{Z_{t+1} = b_{t+1}\} \prod_{i=t+2}^{t+k} \mathbb{P}\{Z_i = b_i | Z_{t+1}^{i-1} = b_{t+1}^{i-1}\}} \\ & \quad \times \frac{\prod_{j=t+k+1}^n \mathbb{P}\{Z_j = b_j | Z_{j-k}^{j-1} = b_{j-k}^{j-1}\}}{\prod_{j=t+k+1}^n \mathbb{P}\{Z_j = b_j | Z_{j-k}^{j-1} = b_{j-k}^{j-1}\}} \\ &= \frac{\mathbb{P}\{Z_t = b_t\} \prod_{i=t+1}^{t+k} \mathbb{P}\{Z_i = b_i | Z_t^{i-1} = b_t^{i-1}\}}{\mathbb{P}\{Z_{t+1} = b_{t+1}\} \prod_{i=t+2}^{t+k} \mathbb{P}\{Z_i = b_i | Z_{t+1}^{i-1} = b_{t+1}^{i-1}\}} \\ &= \frac{\mathbb{P}\{Z_t^{t+k} = b_t^{t+k}\}}{\mathbb{P}\{Z_{t+1}^{t+k} = b_{t+1}^{t+k}\}} = \mathbb{P}\{Z_t = b_t | Z_{t+1}^{t+k} = b_{t+1}^{t+k}\}. \end{aligned} \quad (22)$$

Now we have for any $k+1 \leq j \leq n-k$:

$$\begin{aligned} & \mathbb{P}\{Z_j = a | Z_1^{j-1} = b_1^{j-1}, Z_{j+1}^n = b_{j+1}^n\} \\ &= \frac{\mathbb{P}\{Z_j = a, Z_{j+1}^n = b_{j+1}^n | Z_1^{j-1} = b_1^{j-1}\}}{\mathbb{P}\{Z_{j+1}^n = b_{j+1}^n | Z_1^{j-1} = b_1^{j-1}\}} \\ & \stackrel{(a)}{=} \frac{\mathbb{P}\{Z_j = a, Z_{j+1}^n = b_{j+1}^n | Z_{j-k}^{j-1} = b_{j-k}^{j-1}\}}{\mathbb{P}\{Z_{j+1}^n = b_{j+1}^n | Z_{j-k}^{j-1} = b_{j-k}^{j-1}\}} \\ &= \mathbb{P}\{Z_j = a | Z_{j-k}^{j-1} = b_{j-k}^{j-1}, Z_{j+1}^n = b_{j+1}^n\} \\ &= \frac{\mathbb{P}\{Z_j = a, Z_{j-k}^{j-1} = b_{j-k}^{j-1}, Z_{j+1}^n = b_{j+1}^n\}}{\mathbb{P}\{Z_{j-k}^{j-1} = b_{j-k}^{j-1} | Z_{j+1}^n = b_{j+1}^n\}} \\ & \stackrel{(b)}{=} \frac{\mathbb{P}\{Z_j = a, Z_{j-k}^{j-1} = b_{j-k}^{j-1} | Z_{j+1}^n = b_{j+1}^n\}}{\mathbb{P}\{Z_{j-k}^{j-1} = b_{j-k}^{j-1} | Z_{j+1}^n = b_{j+1}^n\}} \\ &= \mathbb{P}\{Z_j = a | Z_{j-k}^{j-1} = b_{j-k}^{j-1}, Z_{j+1}^n = b_{j+1}^n\} \end{aligned} \quad (23)$$

where (a) follows from the k th-order Markov assumption for Z_1^n , (b) follows by (22) from the k th-order Markov property of the time-reversed sequence Z_n, Z_{n-1}, \dots, Z_1 . What (23) says is that a Markov process indexed by \mathbb{Z} is also a Markov random field indexed by \mathbb{Z} . In fact, the converse is true if assuming all finite realizations have strictly positive probabilities and stationarity, that is, a stationary Markov process indexed by \mathbb{Z} is equivalent to a stationary Markov random field indexed by \mathbb{Z} if this

positivity condition holds [28]–[32]. From this point of view, it is easy to understand intuitively why the time-reversed version of a Markov process indexed by Z is still Markovian.

Based on (23), with the same method used in deriving (20), we have for any $k + 1 \leq j \leq n - k$,

$$\begin{aligned} & \mathbb{P}\{Z_j = a | Z_1^{j-1} = b_1^{j-1}, Z_{j+1}^n = b_{j+1}^n\} \\ & \propto \mathbb{P}\{Z_j = a | Z_{j-k}^{j-1} = b_{j-k}^{j-1}\} \\ & \quad \times \mathbb{P}\{Z_{j+1} = b_{j+1} | Z_{j-k+1}^j = b_{j-k+1}^{j-1} a\} \\ & \quad \times \prod_{t=j+2}^{j+k} \mathbb{P}\{Z_t = b_t | Z_{t-k}^{t-1} = b_{t-k}^{t-1} a b_{j+1}^{t-1}\} \\ & = \mathbb{P}\{Z_j = a | Z_1^{j-1} = b_1^{j-1}\} \\ & \quad \times \mathbb{P}\{Z_{j+1} = b_{j+1} | Z_1^j = b_1^{j-1} a\} \\ & \quad \times \prod_{t=j+2}^{j+k} \mathbb{P}\{Z_t = b_t | Z_1^{t-1} = b_1^{j-1} a b_{j+1}^{t-1}\} \quad (24) \end{aligned}$$

because of the k th-order Markov assumption. And the normalizing constant is the reciprocal of

$$\mathbb{P}\{Z_{j+1}^{j+k} = b_{j+1}^{j+k} | Z_{j-k}^{j-1} = b_{j-k}^{j-1}\}.$$

And from another direction with the same method used in deriving (21), based on (23), for any $k + 1 \leq j \leq n - k$:

$$\begin{aligned} & \mathbb{P}\{Z_j = a | Z_1^{j-1} = b_1^{j-1}, Z_{j+1}^n = b_{j+1}^n\} \\ & \propto \mathbb{P}\{Z_j = a | Z_{j+1}^{j+k} = b_{j+1}^{j+k}\} \\ & \quad \times \mathbb{P}\{Z_{j-1} = b_{j-1} | Z_j^{j+k-1} = a b_{j+1}^{j+k-1}\} \\ & \quad \times \prod_{t=j-k}^{j-2} \mathbb{P}\{Z_t = b_t | Z_{t+1}^{t+k} = b_{t+1}^{t+k} a b_{j+1}^{t+k}\} \\ & = \mathbb{P}\{Z_j = a | Z_{j+1}^n = b_{j+1}^n\} \\ & \quad \times \mathbb{P}\{Z_{j-1} = b_{j-1} | Z_j^n = a b_{j+1}^n\} \\ & \quad \times \prod_{t=j-k}^{j-2} \mathbb{P}\{Z_t = b_t | Z_{t+1}^n = b_{t+1}^{j-1} a b_{j+1}^n\} \quad (25) \end{aligned}$$

because the time-reversed sequence Z_n, Z_{n-1}, \dots, Z_1 is still k th-order Markov. And the normalizing constant is the reciprocal of

$$\mathbb{P}\{Z_{j-k}^{j-1} = b_{j-k}^{j-1} | Z_{j+1}^{j+k} = b_{j+1}^{j+k}\}.$$

We refer to the model specified by (24) as forward OGM or f-OGM; and we refer to the model specified by (25) as backward OGM or b-OGM. It is interesting to compare (20) with (24), and to compare (21) with (25). The idea of the memory length k has its direct intuitive meaning here. It is also interesting to compare GM with f-OGM or b-OGM, that is, (14) (or (16)) versus (24) or (25). When the process Z is k th-order Markov with $k = 2m + 1$, those three expressions coincide theoretically up to a normalizing constant that does not depend on a , though they may have differences in practical estimation. And GM combines the unidirectional conditional distributions from two sides; whereas f-OGM or b-OGM utilizes only unidirectional conditional distributions from one side. In GM, f-OGM and b-OGM, if $k = 2m + 1$, f-OGM and b-OGM have less time complexity in computing the bidirectional conditional distributions.

In practice, we use any unidirectional model (e.g., the one mentioned in Section II-A) to estimate those terms in the f-OGM and b-OGM schemes (either (24) or (25) or both), instead of directly estimating the unidirectional conditional distributions of the k th-order Markov approximation of the process Z with state space size $|\mathcal{B}|^k = (M')^k$, in the same spirit of GM, where $|\cdot|$ represents the length of a string or the cardinality of a set.

C. The Adaptive Bidirectional Model Based on CTW

CTW is a widely used method in lossless data compression [2], [3], and traditionally it is viewed as a *sequentially available* prediction scheme that is crucial in model-based data compression algorithms such as arithmetic coding. But by some modifications of the CTW algorithm, we can get an adaptive (partial) bidirectional model based on CTW.

Notice that CTW estimates block probabilities (i.e., joint probabilities) instead of conditional probabilities. The observation that

$$\begin{aligned} & \mathbb{P}\{Z_j = a | Z_1^{j-1} = b_1^{j-1}, Z_{j+1}^n = b_{j+1}^n\} \\ & \propto \mathbb{P}\{Z_1^n = b_1^{j-1} a b_{j+1}^n\} = P_{Z_1^n}(b_1^{j-1} a b_{j+1}^n) \end{aligned}$$

with the normalizing constant being independent of $a \in \mathcal{B}$, leads to the conclusion that the problem of bidirectional modeling of Z can be solved by estimating all block probabilities. Only if $b_1^{j-1} a b_{j+1}^n$ is a realization of Z_1^n can we directly use CTW to estimate $P_{Z_1^n}(b_1^{j-1} a b_{j+1}^n)$. This is because the CTW estimate for an arbitrary c_1^n is meaningless unless it is intimately related to the given realization z_1^n (say $c_1^n = z_1^n$). Therefore, it is more useful to discuss the CTW approach in applications such as statistical inference (cf. (4) and (1)) in which only $P_{Z_1^n}(z_1^{j-1} a z_{j+1}^n)$ is needed for any $a \in \mathcal{B}, j = 1, 2, \dots, n$, where z_1^n is the given channel output realization. In this case, *pretending* $z_1^{j-1} a z_{j+1}^n$ to be a realization of Z_1^n only leads to negligible differences in all the empirical distributions involved which converge to the true probability law for the process Z . We specify our CTW-based partial bidirectional model by the direct basic CTW estimates for $z_1^{j-1} a z_{j+1}^n, \forall a \in \mathcal{B}, j = D + 1, D + 2, \dots, n - D$ where D is any given maximum memory length to be considered. The proofs of the results here are similar to those in [33], but with the difference that ours is a one-pass approach. The algorithm in Section III-B discusses an efficient implementation of the scheme mentioned here.

We fix a specific $1 \leq j^*(n) \leq n$ and $a^* \in \mathcal{B}$, and let $Y(n) = (Y_t(n))_{t \in \mathbb{Z}}$ be the process such that

$$Y_{j^*(n)}(n) = a^*, Y_t(n) = Z_t, \quad \forall t \in \mathbb{Z} \setminus \{j^*(n)\},$$

and thus, $Y_1^n(n) = Z_1^{j^*(n)-1} a^* Z_{j^*(n)+1}^n$ for any finite segment Z_1^n with $1 \leq j^*(n) \leq n$. Letting $j^*(n)$ be generally dependent on n enables us to include both relatively small and relatively large time indices $j^*(n)$ that can be close to n . We refer to [2] for the original CTW algorithm and terminology of binary sources, and we use its obvious modification to the general finite-alphabet case. The following point of view is taken here: the basic CTW tree can be constructed indefinitely for the

process itself as we increase the time horizon i , thus we can speak of statistics or the basic CTW tree for processes at any time $i \in \mathbb{N}^*$ without paying attention to the actual total number of available observations n unless needed. Let $N_s^i(b)$ denote the number of occurrences of symbol $b \in \mathcal{B}$ in the unidirectional context corresponding to s in the original process Z after updating the basic CTW tree for the i th symbol $Z_i, i \geq D + 1$, and the corresponding root node of the basic CTW tree is denoted by \mathbf{e} . Let $\tilde{N}_s^{i,n}(b), \mathbf{e}$ denote the corresponding quantities for $Y(n)$ for $i \geq D + 1$. Let $\tilde{P}_e^s(i, n), \tilde{P}_w^s(i, n)$ be the estimated and weighted probabilities at node s after updating the basic CTW tree of $Y(n)$ for the i th symbol $Y_i(n), i \geq D + 1$. Notice here that $N_s^i(b)$ only depends on i, Z_1^i and does not depend on n at all. The dependence of $Y(n), \tilde{N}_s^{i,n}(b), \tilde{P}_e^s(i, n), \tilde{P}_w^s(i, n)$ on n is only through its $j^*(n)$ th observation $Y_{j^*(n)}(n) = a^*$, and $\tilde{N}_s^{i,n}(b), \tilde{P}_e^s(i, n), \tilde{P}_w^s(i, n)$ only depend on $i, Y_1^i(n)$. Both the basic CTW trees for Z and $Y(n)$ are considered to be the same complete, balanced $|\mathcal{B}|$ -ary tree with common depth D , but with different empirical counts $N_s^i(b), \tilde{N}_s^{i,n}(b)$ associated to each node s . Thus, their root nodes are both denoted by the same notation \mathbf{e} which is identified with the empty string. Define for any node $s \in \mathcal{B}^m$ with some $m \leq D$ (we identify a node and its corresponding unidirectional context):

$$\tilde{\beta}_{i,n}^s \triangleq \frac{\tilde{P}_e^s(i, n)}{\prod_{s' \in C(s)} \tilde{P}_w^{s'}(i, n)}$$

where $C(s)$ denotes the set of child nodes for node s . Let $\{p \rightarrow q\}$ denote the set of nodes that belong to the path from node p back to node q , including both nodes p, q . The updating path for the i th symbol $Y_i(n)$ is defined as the set of nodes $\{Y_{i-D}^{i-1}(n) \rightarrow \mathbf{e}\}$. Furthermore, we notice that for any node s and any $n \in \mathbb{N}^*$:

$$|\tilde{N}_s^{i,n}(b) - N_s^i(b)| \leq D + 1, \quad \forall i \geq D + 1, b \in \mathcal{B}. \quad (26)$$

Lemma 1: Let $\{v \rightarrow \mathbf{e}\}$ be the updating path for the i th symbol $Y_i(n)$, i.e., $v = Y_{i-D}^{i-1}(n)$. Let

$$\gamma_{i-1,n} \triangleq \prod_{q=Y_{i-d}^{i-1}(n):d=0,1,\dots,D-1} \frac{1}{\tilde{\beta}_{i-1,n}^q + 1}$$

and for the node $s = Y_{i-d}^{i-1}(n)$ at depth d , let

$$\alpha_{i-1,n}^s \triangleq \tilde{\beta}_{i-1,n}^s \prod_{q=Y_{i-k}^{i-1}(n):k=0,1,\dots,d} \frac{1}{\tilde{\beta}_{i-1,n}^q + 1}.$$

For $\forall i \geq D + 2, a \in \mathcal{B}$, let

$$\begin{aligned} \hat{P}_i(Y_1^{i-1}(n))[a] &\triangleq \gamma_{i-1,n} \frac{\tilde{N}_v^{i-1,n}(a) + \frac{1}{2}}{\sum_{b \in \mathcal{B}} \tilde{N}_v^{i-1,n}(b) + \frac{1}{2}|\mathcal{B}|} \\ &+ \sum_{s=Y_{i-d}^{i-1}(n):d=0,1,\dots,D-1} \alpha_{i-1,n}^s \frac{\tilde{N}_s^{i-1,n}(a) + \frac{1}{2}}{\sum_{b \in \mathcal{B}} \tilde{N}_s^{i-1,n}(b) + \frac{1}{2}|\mathcal{B}|} \end{aligned} \quad (27)$$

be the estimated unidirectional conditional distribution of $Y_i(n)$ given $Y_1^{i-1}(n)$ at time $i - 1$ (i.e., after updating the basic CTW tree according to $Y_{i-1}(n)$) which is not dependent on $Y_i(n)$.

Then, the unidirectional conditional distributions in (27) satisfy

$$\tilde{P}_w^{\mathbf{e}}(D + 1, n) \prod_{i=D+2}^n \hat{P}_i(Y_1^{i-1}(n))[Y_i(n)] = \tilde{P}_w^{\mathbf{e}}(n, n) \quad (28)$$

which is the weighted probability at root node \mathbf{e} and thus is the estimated block (joint) probability of $Y_1^n(n)$ by the basic CTW algorithm.

Proof: See Appendix I. \square

Proposition 1: Let $\hat{P}_i(Y_1^{i-1}(n))$ be as in Lemma 1. Let Z be stationary ergodic, then for any $n \in \mathbb{N}^*$ almost surely

$$\lim_{i \rightarrow \infty} \left\{ \left| \hat{P}_i(Y_1^{i-1}(n))[Y_i(n)] - P_{Z_0|Z_{-D}^{-1}}(Y_i(n)|Y_{i-D}^{i-1}(n)) \right| \times 1_{P_{Z_{-D}^{-1}}(Y_{i-D}^{i-1}(n)) > 0} \right\} = 0.$$

A fortiori, as $n \rightarrow \infty, i \rightarrow \infty$ with $n \geq i$, almost surely

$$\left\{ \left| \hat{P}_i(Y_1^{i-1}(n))[Y_i(n)] - P_{Z_0|Z_{-D}^{-1}}(Y_i(n)|Y_{i-D}^{i-1}(n)) \right| \times 1_{P_{Z_{-D}^{-1}}(Y_{i-D}^{i-1}(n)) > 0} \right\} \rightarrow 0.$$

Proof: See Appendix II. \square

To summarize Lemma 1 and Proposition 1, the basic CTW-based algorithm at least gives an asymptotically good D th-order Markov approximation for the block (joint) probability $P_{Z_1^n}(Y_1^n(n))$ up to a normalizing constant for any

$$Y_1^n(n) = Z_1^{j^*(n)-1} a^* Z_{j^*(n)+1}^n, 1 \leq j^*(n) \leq n, \quad a^* \in \mathcal{B}$$

in the sense of (28) in Lemma 1 and Proposition 1. Recall that (28) is the final CTW estimate for

$$P_{Z_1^n}(Y_1^n(n)) = P_{Z_1^n}(Z_1^{j^*(n)-1} a^* Z_{j^*(n)+1}^n).$$

The problematic matter is the complexity of specifying this partial bidirectional model for all needed sequences $z_1^{j-1} a z_{j+1}^n, j = 1, 2, \dots, n, a \in \mathcal{B}$. The basic CTW tree is constructed to estimate the block probability of only a single realization $z_1^{j-1} a z_{j+1}^n$ in linear time and space. Thus, for example, using the basic CTW to get estimates for the $O(n|\mathcal{B}|)$ realizations $\{z_1^{j-1} a z_{j+1}^n : j = D + 1, D + 2, \dots, n, a \in \mathcal{B}\}$ which are needed in statistical inference applications like non-causal denoising (cf. (4) and (1)) takes $O(n^2)$ time complexity. By specifying a maximal memory length D , the effect of a symbol to the whole basic CTW tree is limited to a range of constant length. Bearing in mind the statistical inference applications such as in (4), we get a solution for building a partial bidirectional model based on basic CTW as follows. To estimate

$$\mathbb{P}\{Z_j = a | Z_1^{j-1} = b_1^{j-1}, Z_{j+1}^n = b_{j+1}^n\} \propto \mathbb{P}\{Z_1^n = z_1^{j-1} a z_{j+1}^n\}$$

for any $a \in \mathcal{B}$ and any $j = 1, 2, \dots, n$, we first build the basic CTW tree for z_1^n (note z_1^n is our given realization to do all estimation) with the given memory length D . Then we need to modify this tree to incorporate any symbol $a \in \mathcal{B}$ at any position j without changing other symbols at positions other than the j th position. Notice the symbol at position j only affects the counts of nodes on the path from root to the leaves

$$z_{j-D}^{j-1}, z_{j-D+1}^j, z_{j-D+2}^{j+1}, \dots, z_j^{j+D-1}$$

in the basic CTW tree. Then we just change z_j to symbol a and make necessary *local* modifications to the nodes in those paths so that we can estimate $\mathbb{P}\{Z_1^n = z_1^{j-1} a z_{j+1}^n\}$ as the weighted probability in the root node. We need to recover the original basic CTW tree for z_1^n by reversing the above operations, in order to get estimates for positions/time indices later than j . The estimations of all the probabilities of the $O(n|\mathcal{B}|)$ realizations is achieved within linear time and space complexity. This problem now becomes mainly an algorithmic one, which is discussed in detail in Section III-B specifically about the basic CTW-based bidirectional modeling algorithm.

There are some other possible variants for bidirectional modeling based on CTW ideas. Instead of using unidirectional contexts, we can construct CTW-like trees with symmetric bidirectional contexts, from which we can estimate the desired bidirectional conditional distributions by a two-pass approach (similar to that of [33]) instead of block probabilities (joint probabilities). There are similar equations like (27) for the two-pass-based approach for the bidirectional versions of both the basic and extended CTWs which directly give estimates of bidirectional conditional distributions. Experimental results show that their performance is inferior to the other models we propose here, and thus they are not included in this paper.

D. The Adaptive BCT

Motivated by (5), we now construct a bidirectional model directly instead of only constructing bidirectional models from unidirectional models. In addition to increased computational complexity, when we consider bidirectional contexts we have to cope with their nonuniqueness.

1) *Bidirectional Contexts*: Let $Z = (Z_t)_{t \in \mathbb{Z}}$ be a stationary process with finite alphabet \mathcal{B} . For a segment Z^n , a pair of strings $(b_{j-s}^{j-1}, b_{j+1}^{j+t})$ for some integers $s, t: 0 \leq s \leq j-1, 0 \leq t \leq n-j$ is called a bidirectional context of $(b_1^{j-1}, b_{j+1}^n) \in \mathcal{B}^{j-1} \times \mathcal{B}^{n-j}$ if

$$\begin{aligned} \forall a \in \mathcal{B}, \quad & \mathbb{P}\{Z_j = a | Z_{j-s}^{j-1} = b_{j-s}^{j-1}, Z_{j+1}^{j+t} = b_{j+1}^{j+t}\} \\ & = \mathbb{P}\{Z_j = a | Z_1^{j-1} = b_1^{j-1}, Z_{j+1}^n = b_{j+1}^n\} \quad (29) \end{aligned}$$

with the *minimality* property that for any $s' \leq s, t' \leq t, s'+t' < s+t$,

$$\begin{aligned} \exists a \in \mathcal{B}, \quad & \mathbb{P}\{Z_j = a | Z_{j-s'}^{j-1} = b_{j-s'}^{j-1}, Z_{j+1}^{j+t'} = b_{j+1}^{j+t'}\} \\ & \neq \mathbb{P}\{Z_j = a | Z_1^{j-1} = b_1^{j-1}, Z_{j+1}^n = b_{j+1}^n\}. \quad (30) \end{aligned}$$

If $s = j-1, t = n-j$, or equivalently, $s+t = n-1$, the bidirectional context is said to be trivial. Let $S_n(b_1^{j-1}, b_{j+1}^n)$ be

the bidirectional context set for (b_1^{j-1}, b_{j+1}^n) , namely, the set of all bidirectional contexts for (b_1^{j-1}, b_{j+1}^n) . In fact

$$|S_n(b_1^{j-1}, b_{j+1}^n)| \leq \min\{j-1, n-j\}.$$

Note that $|S_n(b_1^{j-1}, b_{j+1}^n)|$ can be larger than 1, in contrast with the unidirectional model [8]–[12], where a unique (minimal) unidirectional context is identified for the same historical data b_1^{j-1} .

2) *Bidirectional Context Trees*: For a segment Z^n of a stationary process Z , let $\mathcal{B}^0 \triangleq \{\mathbf{e}\}$ be the set consisting of only the empty string, we define the set of bidirectional substrings of bidirectional contexts in $S_n(b_1^{j-1}, b_{j+1}^n)$ which consists of the most recent past and future data with respect to time instant j , namely

$$\begin{aligned} T_n(b_1^{j-1}, b_{j+1}^n) \triangleq & \left\{ (u, v) : u \in \bigcup_{t=0}^{j-1} \mathcal{B}^t, v \in \bigcup_{t=0}^{n-j} \mathcal{B}^t \right. \\ & \text{such that } \exists (x, y) \in S_n(b_1^{j-1}, b_{j+1}^n), \\ & \left. x_{|x|-|u|+1}^{|x|} = u, y_1^{|v|} = v \right\}. \quad (31) \end{aligned}$$

The bidirectional context tree is the tree with all bidirectional contexts identified as incomplete nodes² in this tree, namely it is defined as the set

$$\tau_n = \bigcup_{(b_1^{j-1}, b_{j+1}^n) \in \mathcal{B}^{j-1} \times \mathcal{B}^{n-j}} T_n(b_1^{j-1}, b_{j+1}^n) \quad (32)$$

with a tree structure, in which a node corresponds to a pair of strings (u, v) and *vice versa*. This correspondence is as follows: the root corresponds to (\mathbf{e}, \mathbf{e}) , and all branches originating from a node are labeled by a pair of symbols in $\mathcal{B} \cup \mathcal{B}^0$, but not both in \mathcal{B}^0 . If a node is obtained by traveling through a path from root (\mathbf{e}, \mathbf{e}) consisting of branches labeled by $(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$ (in that order) for some $x_i, y_i \in \mathcal{B} \cup \mathcal{B}^0, i = 1, 2, \dots, m$, then this node corresponds to the pair of strings $(x_m x_{m-1} \dots x_2 x_1, y_1 y_2 \dots y_m)$ by juxtapositions from inside to outside. Conversely, for any pair of strings (u, v) , if $|u| = |v|$, then it corresponds to the node obtained by traveling through the path $(u_{|u|}, v_1), (u_{|u|-1}, v_2), \dots, (u_1, v_{|u|})$ from the root; if $|u| < |v|$, then the path is

$$(u_{|u|}, v_1), (u_{|u|-1}, v_2), \dots, (u_1, v_{|u|}), (\mathbf{e}, v_{|u|+1}), (\mathbf{e}, v_{|u|+2}), \dots, (\mathbf{e}, v_{|v|});$$

similarly, for the case $|u| > |v|$. Because of this correspondence, a node in the bidirectional context tree and its corresponding pair of strings (possibly a bidirectional context) are used interchangeably.

Bidirectional context sets completely characterize bidirectional conditional distributions for a finite time segment Z^n , while bidirectional context trees are more suitable for the purpose of estimating bidirectional context sets and hence those

²Incomplete nodes are those which actually do not have all the possible child nodes, i.e., they only have some of the possible child nodes. They can be internal nodes, and leaf nodes are necessarily incomplete.

bidirectional conditional distributions. In contrast to the unidirectional case in [8]–[12], the data structure of a bidirectional context tree is more complicated and needs further elaboration.

3) *Estimation of the Bidirectional Context Sets From z^n* : Our definitions of bidirectional models and their bidirectional contexts are suitable when we want to build a source model for a finite data segment Z^n . We want (29) to hold approximately for general discrete stationary process Z^n with estimates of the bidirectional context sets $S_n(b_1^{j-1}, b_{j+1}^n)$ for all (b_1^{j-1}, b_{j+1}^n) . We use $\hat{S}_{z^n}(b_1^{j-1}, b_{j+1}^n)$ to denote the estimated bidirectional context set from z^n , which can be obtained by an efficient algorithm (see Section III-C for details). The estimated bidirectional context tree corresponding to those estimated bidirectional context sets is denoted by $\hat{\tau}_n(z^n)$.

4) *Two Constraints for an Estimated Bidirectional Context Tree*: The first constraint is the consecutive constraint, which is implicit in the definition of bidirectional contexts, and it is better made explicit. In addition, this consecutive constraint applies to estimated bidirectional context trees as well as bidirectional context trees. For a bidirectional context (w, v) we have $w = b_{j-s}^{j-1}, v = b_{j+1}^{j+t}$ for some b_1^{j-1}, b_{j+1}^n, s and t . By referring to w as the historical (left) side of this bidirectional context, and v as the future (right) side of this bidirectional context, we see that only symbols in consecutive time instants are allowed to exist in either side of a bidirectional context, i.e., $b_{j-s}^{j-1}, b_{j+1}^{j+t}$. In the construction of such a tree, this implicit constraint induces the following alternative situations for node (w, v) (recall that \mathbf{e} is the empty string with length $|\mathbf{e}| = 0$).

- $|w| = |v|$: Three possible kinds of child nodes with the forms (aw, vb) , $(aw, v\mathbf{e})$, or $(\mathbf{e}w, vb)$ for some $a, b \in \mathcal{B}$.
- $|w| < |v|$: One possible kind of child node with the form $(\mathbf{e}w, vb)$ for some $b \in \mathcal{B}$.
- $|w| > |v|$: One possible kind of child node with the form $(aw, v\mathbf{e})$ for some $a \in \mathcal{B}$.

The above constraint comes from simultaneous consideration of contexts on two sides, allowing possible unit length increments on two sides and preserving the consecutiveness. The philosophy is that the nearest symbols in both the past and the future are always the most relevant to the present data.

To achieve a good performance/complexity tradeoff we place a constraint N on the maximal tree depth (or the maximal single-sided context length) of an estimated bidirectional context tree. In the worst case, we have $O((M')^{2N})$ nodes in an estimated bidirectional context tree, which is a constant not depending on the data length n , and we thus have linear complexity in both time and space. In practice, this assumption works pretty well, and a larger maximal tree depth does not necessarily lead to better estimation accuracy (in terms of denoising performance, cf. Section IV) compared to a smaller one.

5) *Empirical Bidirectional Conditional Distributions From z^n* : We define $\mathcal{B}^* \triangleq \bigcup_{n \in \mathbb{N}^*} \mathcal{B}^n$ and

$$N_u(w) \triangleq \sum_{t=1}^{|w|-|w|+1} 1_{u_t^{t+|w|-1}=w}, \forall u \in \mathcal{B}^*, w \in \bigcup_{t=1}^{|w|} \mathcal{B}^t,$$

$$\hat{P}_u(a|w, v) \triangleq \frac{N_u(wav)}{\sum_{b \in \mathcal{B}} N_u(wbv)} \quad (33)$$

for all $u \in \mathcal{B}^*, a \in \mathcal{B}, w, v \in \bigcup_{t=1}^{|u|-1} \mathcal{B}^t \cup \{\mathbf{e}\}, |w|+|v|+1 \leq |u|$. Note that (33) is an estimate for

$$\mathbb{P}\{Z_j = a | Z_{j-|w|}^{j-1} = w, Z_{j+1}^{j+|v|} = v\}$$

for any $|w| + 1 \leq j \leq n - |v|$ if $u = z^n$, since Z is stationary.

6) *Estimators for Bidirectional Conditional Distributions z^n* : For any (b_1^{j-1}, b_{j+1}^n) , equipped with the bidirectional context sets $\hat{S}_{z^n}(b_1^{j-1}, b_{j+1}^n)$ estimated from a realization z^n , let (w, v) be any element in $\hat{S}_{z^n}(b_1^{j-1}, b_{j+1}^n)$. Then by definition of $S_n(b_1^{j-1}, b_{j+1}^n)$ and (29), we know (33) becomes the estimate

$$\hat{P}_{z^n}(a|w, v) \quad (34)$$

of

$$\mathbb{P}\{Z_j = a | Z_1^{j-1} = b_1^{j-1}, Z_{j+1}^n = b_{j+1}^n\}$$

for all $(w, v) \in \hat{S}_{z^n}(b_1^{j-1}, b_{j+1}^n)$. Note that the multiplicity of valid choices for (w, v) in (34) does not arise in the unidirectional model [8]–[12]. To resolve the ambiguity in finding a bidirectional context for the given bidirectional data (b_1^{j-1}, b_{j+1}^n) , an exponential weighting scheme is used. To motivate this scheme, we notice the similarity of our problem with that of estimating an *unknown* distribution P from a collection of distributions $(P_i)_{i=1}^k$ on a finite alphabet \mathcal{B} . A natural method is to minimize the divergence between the mixture of $(P_i)_{i=1}^k$ and P . An optimization problem can be formulated as

$$\begin{aligned} & \text{minimize} && D\left(\sum_{i=1}^k \alpha_i P_i || P\right) \\ & \text{subject to} && \sum_{i=1}^k \alpha_i = 1, \alpha_i \geq 0, i = 1, 2, \dots, k. \end{aligned}$$

Here we refer to [34] for terminology and corresponding results. The convexity of the objective function in $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_k)^T \in [0, 1]^k$ follows directly from the convexity of divergence in its two arguments. And constraints are all affine, thus this problem is a convex optimization problem. Now we proceed as if P were known to get some idea of how to choose the weighting factors $(\alpha_i)_{i=1}^k$. Since Slater's condition holds, we know strong duality holds, and Karush–Kuhn–Tucker conditions are sufficient and necessary for α to solve the above optimization problem [34]. The Lagrangian is defined as

$$\begin{aligned} L(\alpha, \nu, \mu) &= D\left(\sum_{i=1}^k \alpha_i P_i || P\right) - \sum_{i=1}^k \nu_i \alpha_i + \mu \left(\sum_{i=1}^k \alpha_i - 1\right) \\ &= \sum_{c \in \mathcal{B}} \sum_{i=1}^k \alpha_i P_i(c) \log \frac{\sum_{l=1}^k \alpha_l P_l(c)}{P(c)} - \sum_{i=1}^k \nu_i \alpha_i \\ &\quad + \mu \left(\sum_{i=1}^k \alpha_i - 1\right) \end{aligned}$$

for dual variables $\nu = (\nu_1, \nu_2, \dots, \nu_k)^T \in \mathbb{R}_+^k, \mu \in \mathbb{R}$. One of the Karush–Kuhn–Tucker conditions is the gradient of Lagrangian $L(\alpha, \nu, \mu)$ with respect to α vanishes

$$\nabla_{\alpha} L(\alpha, \nu, \mu) = 0 \quad (35)$$

as noncausal denoising. The consistency results in Section II-C provide a partial guarantee for the performance of our bidirectional model (which is applied to noncausal discrete denoising) based on the basic CTW. Instead of omitting the original basic CTW construction algorithm [2], we next give a relatively complete description of our modification of that algorithm. The first step is the same as in [2], but later steps need to be modified to the bidirectional modeling setting.

We first state some updating procedures that are repeatedly used in the algorithm. Let s be any node in the basic CTW tree.

Module 1 (Increasing Procedure for Symbol b at s): Update the estimated probability for node s as

$$P_e^s \leftarrow \frac{N_s(b) + \frac{1}{2}}{\sum_{a \in \mathcal{B}} N_s(a) + \frac{1}{2}|\mathcal{B}|} P_e^s.$$

Update the count for the symbol b in this context s as

$$N_s(b) \leftarrow N_s(b) + 1$$

and do nothing to other counts $N_s(a)$, $a \in \mathcal{B} \setminus \{b\}$.

Update the weighted probability for node s as

$$P_w^s = \begin{cases} \frac{1}{2}P_e^s + \frac{1}{2} \prod_{s' \in C(s)} P_w^{s'}, & \text{if } 0 \leq l(s) < D \\ P_e^s, & \text{if } l(s) = D \end{cases}$$

where $l(s)$ is the depth of node s , $C(s)$ is the set of child nodes of s , and then let

$$s \leftarrow \text{father node of } s$$

if the father node of s exists. After that, do the same updating for the “new” s until we have finally updated the root node. Notice that only the root node does not have a father node.

Module 2 (Decreasing Procedure for Symbol b at s): Update the estimated probability for node s as

$$P_e^s \leftarrow \frac{\sum_{a \in \mathcal{B}} N_s(a) + \frac{1}{2}|\mathcal{B}| - 1}{N_s(b) - \frac{1}{2}} P_e^s.$$

Update the count for the symbol b in this context s as

$$N_s(b) \leftarrow N_s(b) - 1$$

and do nothing to other counts $N_s(a)$, $a \in \mathcal{B} \setminus \{b\}$.

Update the weighted probability for node s as

$$P_w^s = \begin{cases} \frac{1}{2}P_e^s + \frac{1}{2} \prod_{s' \in C(s)} P_w^{s'}, & \text{if } 0 \leq l(s) < D \\ P_e^s, & \text{if } l(s) = D \end{cases}$$

and then let

$$s \leftarrow \text{father node of } s$$

if the father node of s exists. After that, do the same updating for the “new” s until we have finally updated the root node. Notice that only the root node does not have a father node.

Now we describe the basic CTW based algorithm for building a partial bidirectional model.

1) *Step 1:* Construct the basic CTW tree for the given realization z_1^n as follows. Initially we assume a complete, balanced $|\mathcal{B}|$ -ary tree with depth D , and all counts $N_s(a)$ of the number of occurrences of symbol a in any unidirectional context s in the given realization up to now are set to be 0.

- Phase 1: Set $i = D + 1$.
- Phase 2: Start from the root node, go to branches labelled by

$$z_{i-1}, z_{i-2}, \dots, z_{i-D}$$

to arrive at the leaf node $s = z_{i-D}^{i-1}$.

- Phase 3: Start at node s . We update the whole path from s back to the root node by the Increasing Procedure Module 1 for symbol z_i at s until we have finally updated the root node.
- Phase 4: Set $i \leftarrow i + 1$, if $i \leq n$ then go to Phase 2, otherwise, the construction of basic CTW tree for z_1^n is completed and we go to the next step.

In fact, after this step, we have already obtained the estimate for $\mathbb{P}\{Z_1^n = z_1^n\}$, thus, in the following steps we do not need to consider the symbol z_i at time $i = D + 1, D + 2, \dots, n - D$.

2) *Step 2:* Set $i = D + 1$.

From *Step 3* to *Step 5*, we will get an estimate for $\mathbb{P}\{Z_1^n = z_1^{i-1} a z_{i+1}^n\}$ that is proportional to

$$\mathbb{P}\{Z_i = a | Z_1^{i-1} = z_1^{i-1}, Z_{i+1}^n = z_{i+1}^n\}, \\ \forall a \in \mathcal{B} \setminus \{z_i\}, D + 1 \leq i \leq n - D.$$

3) *Step 3:* Let a be any symbol in \mathcal{B} other than z_i and let $m = i$. We do the following until $m > i + D$.

- Start at the root node, go through branches labeled by $z_{m-1}, z_{m-2}, \dots, z_{m-D}$ to reach node s . We update the whole path from s back to the root node by the Decreasing Procedure Module 2 for symbol z_m at s until we have finally updated the root node.
- Let $m \leftarrow m + 1$.

4) *Step 4:* Let $m = i$, until $m > i + D$, do:

- If $m = i$, start at the root node, go through branches labeled by $z_{m-1}, z_{m-2}, \dots, z_{m-D}$ to reach node s . We update the whole path from s back to the root node by the Increasing Procedure Module 1 for symbol a at s until we have finally updated the root node.
- If $m > i$, start at the root node, go through branches labeled by

$$z_{m-1}, z_{m-2}, \dots, z_{i+2}, z_{i+1}, a, z_{i-1}, z_{i-2}, \dots, z_{m-D}$$

to reach node s . We update the whole path from s back to the root node by the Increasing Procedure Module 1 for symbol z_m at s until we have finally updated the root node.

- Let $m \leftarrow m + 1$.

5) *Step 5:* Output the weighted probability at the root node as an estimate of $\mathbb{P}\{Z_1^n = z_1^{i-1} a z_{i+1}^n\}$

The following *Step 6* and *Step 7* are aiming at recovering the original basic CTW tree for future use. The tree structure

changes a little, namely, we in fact add several nodes to the original tree using *Step 4*. But we *will* recover all the counts for z_1^n in the original basic CTW tree in later steps. A more efficient implementation deletes the subtree rooted at any node s as we perform the decreasing procedure if its total count $\sum_{a \in \mathcal{B}} N_s(a) = 0$, which gives $P_w^s = P_e^s = 1$. We then have fewer terms in the updating equation in Decreasing Procedure Module 2, due to the terms of value 1. This is in fact the version we use for experiments later.

6) *Step 6*: Let $m = i$, until $m > i + D$, do:

- If $m = i$, start at the root node, go through branches labeled by $z_{m-1}, z_{m-2}, \dots, z_{m-D}$ to reach node s . We update the whole path from s back to the root node by the Decreasing Procedure Module 2 for symbol a at s until we have finally updated the root node.
- If $m > i$, start at the root node, go through branches labeled by

$$z_{m-1}, z_{m-2}, \dots, z_{i+2}, z_{i+1}, a, z_{i-1}, z_{i-2}, \dots, z_{m-D}$$

to reach node s . We update the whole path from s back to the root node by the Decreasing Procedure Module 2 for symbol z_m at s until we have finally updated the root node.

- Let $m \leftarrow m + 1$.

7) *Step 7*: Let $m = i$, until $m > i + D$, do:

- Start at the root node, go through branches labeled by $z_{m-1}, z_{m-2}, \dots, z_{m-D}$ to reach node s . We update the whole path from s back to the root node by the Increasing Procedure Module 1 for symbol z_m at s until we have finally updated the root node.
- Let $m \leftarrow m + 1$.

8) *Step 8*: Go to *Step 3* for other symbols $a \in \mathcal{B} \setminus \{z_i\}$, until all such symbols have been considered and the corresponding estimate for $\mathbb{P}\{Z_1^n = z_1^{i-1} a z_{i+1}^n\}$ has been obtained by above steps.

9) *Step 9*: Set $i \leftarrow i + 1$. If $i \leq n - D$, go to *Step 3*. Otherwise the algorithm ends here.

If the alphabet size $|\mathcal{B}|$ is large, CTW generally does not achieve good performance. To solve this problem, a binary symbol decomposition method is used [35], [36], namely, each symbol in the alphabet \mathcal{B} is decomposed into $\lceil \log_2 |\mathcal{B}| \rceil$ bits. Each bit in a symbol has its own basic CTW tree, where the basic CTW tree is constructed by *only* considering unidirectional contexts consisting of complete symbols, regardless of other bits in the binary decomposition of the symbol. For example, in the sequence sb where s is a string on \mathcal{B} , let $b \in \mathcal{B}$ be decomposed into $L \triangleq \lceil \log_2 |\mathcal{B}| \rceil$ bits b_1^L , then each bit b_i has exactly the same unidirectional contexts, e.g., s is one of their common unidirectional contexts. The root node of the basic CTW tree for the j th bit in one symbol of \mathcal{B} has 2^{j-1} child nodes, $j = 1, 2, \dots, L$, corresponding to each of the previous $(j - 1)$ bits. The subtrees rooted at those first level child nodes are constructed as described above, by considering only unidirectional contexts consisting of complete symbols. Each internal node in those subtrees has exactly $|\mathcal{B}|$ child nodes. The

updating procedure for estimated and weighted probabilities will then be done until the root node for the basic CTW tree of each bit is updated.

C. BCT

The central issue for constructing the adaptive bidirectional model BCT (cf. Section II-D) is to estimate the bidirectional context sets from a realization z^n of a stationary process $Z = (Z_t)_{t \in \mathbb{Z}}$. The bidirectional context tree is first estimated and then bidirectional context sets are specified from the knowledge of the estimated bidirectional context tree. As in [8], the main idea is to keep certain level of dissimilarity (measured by divergence) between bidirectional conditional distribution estimates from leaf nodes and that from their father nodes. This adaptive bidirectional model gives out estimated bidirectional context sets and estimates for

$$\mathbb{P}\{Z_j = a | Z_1^{j-1} = b_1^{j-1}, Z_{j+1}^n = b_{j+1}^n\}, \\ \forall a \in \mathcal{B}, j = 1, 2, \dots, n, b_1^{j-1} \in \mathcal{B}^{j-1}, b_{j+1}^n \in \mathcal{B}^{n-j}$$

which suffices (and actually gives much more) to solve many noncausal statistical inference problems with *no a priori* information about the probability law of the underlying process to be inferred, such as universal discrete denoising with known channel, cf. (4). For notational convenience, we suppress the dependence on z^n below in this section.

1) *Step 1*: Given \mathcal{B} -valued data z^n , construct a tree from z^n by including all possible bidirectional contexts with single-sided context lengths not exceeding N . We store counts $N(wav), \forall a \in \mathcal{B}$ and their total count $\sum_{a \in \mathcal{B}} N(wav)$ in each node (w, v) . Let this tree be $\mathcal{T}(0)$.

2) *Step 2*: Examine every leaf node in $\mathcal{T}(0)$ as follows. Select any leaf node (w, v) in $\mathcal{T}(0)$, where the order of selection here is irrelevant, prune this node if

$$\Delta_{(w,v)} \leq K_n \quad (40)$$

for some pruning parameter $K_n \geq 0$, where

$$\Delta_{(w,v)} \triangleq D(\hat{P}(\cdot | w, v) \| \hat{P}(\cdot | w', v')) \times \sum_{a \in \mathcal{B}} N(wav) \quad (41)$$

is the product of the divergence between the bidirectional conditional distribution estimated from (w, v) and that from its father node (w', v') , and the number of occurrences of the bidirectional context (w, v) in z^n . We recall the definitions of $\hat{P}(\cdot | w, v)$ and $\hat{P}(\cdot | w', v')$ in (33). Note the father node (w', v') of (w, v) has at most three child nodes by construction. After examining each leaf node in $\mathcal{T}(0)$ by the above method, we should get another bidirectional context tree $\mathcal{T}(1)$, which is a subtree of $\mathcal{T}(0)$.

An intuitive explanation for using (40) is to avoid overfitting the data z^n by choosing an appropriate K_n , since if (40) indeed holds, then either we have too similar estimates for the bidirectional conditional distributions estimated from this node (w, v) and its father node (w', v') in the sense of divergence, or we have too sparse data for this node (w, v) .

3) *Step 3*: Repeat *Step 2*, starting from $\mathcal{T}(i)$ to $\mathcal{T}(i+1)$, $i \geq 1$ in the same way as *Step 2* does, until no more pruning is pos-

sible. This tree is our $\hat{\tau}_n$, the estimated bidirectional context tree from data z^n .

4) *Step 4*: Equipped with the estimated bidirectional context tree $\hat{\tau}_n$, we are ready to specify the estimated bidirectional context sets $\hat{S}(b_1^{j-1}, b_{j+1}^n)$. Let $(x_i, y_i)_{i=1}^k$ be the $k = |\hat{\tau}_n|$ nodes in $\hat{\tau}_n$. For any $j = 1, 2, \dots, n$ and any $b_1^{j-1} \in \mathcal{B}^{j-1}, b_{j+1}^n \in \mathcal{B}^{n-j}$, let $\hat{S}(b_1^{j-1}, b_{j+1}^n)$ be those $(x_i, y_i), i \in \{1, 2, \dots, k\}$ such that

$$j \geq |x_i| + 1, \quad j \leq n - |y_i| \quad (42)$$

$$b_{j-|x_i|}^{j-1} = x_i, \quad b_{j+1}^{j+|y_i|} = y_i \quad (43)$$

and no other node in the subtree rooted at (x_i, y_i) satisfies those conditions (42) and (43). Notice the above specification of the bidirectional context sets uses exactly the same method as explained in the example of Section II-D7. If no such (x_i, y_i) exists for (b_1^{j-1}, b_{j+1}^n) , we let $\hat{S}(b_1^{j-1}, b_{j+1}^n) = \{(b_1^{j-1}, b_{j+1}^n)\}$, which only consists of the trivial bidirectional context.

The definition of $\hat{S}(b_1^{j-1}, b_{j+1}^n)$ chooses the deepest nodes that fit the current bidirectional data (b_1^{j-1}, b_{j+1}^n) to be its elements, the estimated bidirectional contexts for (b_1^{j-1}, b_{j+1}^n) . Intuitively speaking, this avoids underfitting the data z^n . Any pair of strings (w, v) corresponding to a node on the path from the root of $\hat{\tau}_n$ to any node (excluded) corresponding to an element (x, y) of $\hat{S}(b_1^{j-1}, b_{j+1}^n)$, is not in $\hat{S}(b_1^{j-1}, b_{j+1}^n)$. Since otherwise, in the subtree rooted at this very node (w, v) , there exists a node that corresponds to a bidirectional context (x, y) in $\hat{S}(b_1^{j-1}, b_{j+1}^n)$ and therefore contradicts our definition of $\hat{S}(b_1^{j-1}, b_{j+1}^n)$. This ensures the minimality property (30) to hold at least for those nodes along the paths to elements in $\hat{S}(b_1^{j-1}, b_{j+1}^n)$.

If $\hat{S}(b_1^{j-1}, b_{j+1}^n)$ contains only the trivial bidirectional context (b_1^{j-1}, b_{j+1}^n) , then we do not have any useful estimation at all due to the dearth of data (or even no data at all for most choices of (b_1^{j-1}, b_{j+1}^n)). In this case, (b_1^{j-1}, b_{j+1}^n) appears at most once in z^n as a bidirectional context, which happens when and only when $z_1^{j-1} = b_1^{j-1}, z_{j+1}^n = b_{j+1}^n$ with realization z^n available for estimation.

5) *Step 5*: For any $a \in \mathcal{B}, j = 1, 2, \dots, n, b_1^{j-1} \in \mathcal{B}^{j-1}, b_{j+1}^n \in \mathcal{B}^{n-j}$, estimate

$$\mathbb{P}\{Z_j = a | Z_1^{j-1} = b_1^{j-1}, Z_{j+1}^n = b_{j+1}^n\}$$

by (38) with bidirectional context sets specified in *Step 4* and weights specified in (39).

Though BCT is similar to the bidirectional counterpart of minimum description length (MDL) [37]–[39] via pseudolikelihood (see, e.g., [40]), its penalty term is dealt with differently. The penalty term corresponds to a parameter that thresholds the minimal normalized divergence between the bidirectional conditional distribution estimates from leaves and those from their father nodes. Using the compression heuristic (cf. Section IV-B) to adaptively change the threshold gives better (if not much better) performance in applications such as denoising than directly minimizing description length or code length, where the

code length interpretation can be found in [9]. The unidirectional model VLMC discussed in Section II-A has a similar MDL interpretation. In fact, [14] provides an elegant solution to the problem of unidirectional modeling via Bayesian information criterion (BIC) [41] and MDL. This has been extended to multidirectional modeling with *symmetric* multidirectional contexts in [40] via modified BIC. In [14], [40] the asymptotic results do not change if the penalty term is multiplied by any strictly positive constant, though for finite data lengths the constant clearly matters. Note also that, in [23], there may exist more than one valid optimal bidirectional context sets for the same sequence, but within each valid bidirectional context set the bidirectional context for a given pair of strings (x, y) is unique; whereas in BCT, the bidirectional context sets are defined probabilistically for each given pair of strings (x, y) . Any (x, y) may possess more than one bidirectional context in this probabilistic sense. That is, the nonuniqueness of bidirectional contexts is dealt with by [23] and our paper in different ways.

IV. COMPARISON OF THE ADAPTIVE BIDIRECTIONAL MODELS: EXPERIMENTS ON DENOISING

We now turn our attention to a concrete application of adaptive bidirectional modeling, namely, noncausal universal discrete denoising. In the experiments, we only consider data indexed by time. DUDE, CTW and BCT can be extended in a natural way (cf. [1], [23]) to images or other multidimensional data. However, computational issues are not trivial due to the growth of the number of contexts.

We compare the seven bidirectional models:

- DUDE (static symmetric contexts as in [1]);
- BFP (cf. Section II-B1);
- GM (cf. Section II-B2);
- f-OGM (cf. (24) in Section II-B3);
- b-OGM (cf. (25) in Section II-B3);
- CTW (cf. Section II-C); and
- BCT (cf. Section II-D);

by using them in the Bayesian optimal noncausal denoiser (1). We let $\Lambda(a, b) = 1_{a \neq b}, \forall a, b \in \mathcal{A} = \mathcal{B} = \hat{\mathcal{A}} = \{1, 2, \dots, M\}, M = M' = \hat{M}$. We compare the performances of universal discrete denoisers induced from those bidirectional models, with *no* knowledge of the clean data x_1^n (channel input).

Various parameters need to be adjusted accordingly to adapt to real data. DUDE needs to choose m , the context length [1]; BFP, GM, f-OGM, and b-OGM have a pruning parameter to construct the unidirectional context tree [11]; In addition, GM has a parameter m that determines the size of the “present,” cf. (14) or (16); f-OGM and b-OGM has a parameter k that is the assumed memory length, cf. (24) and (25); CTW has a memory length parameter D [2]; BCT has a pruning parameter K_n (cf. (40)), an exponent constant β (cf. (39)), and a maximal tree depth N (cf. Section II-D4). The maximal tree depth N can be chosen rather arbitrarily, and longer depths do not necessarily lead to better performances.

A. Complexity Issues

The bidirectional models DUDE, CTW, and BCT have $O(n)$ time and space complexities because the trees involved therein

TABLE I
FIRST-ORDER MARKOV SOURCE (p) AND BSC (δ)

$\delta = 0.01$								
p	DUDE	BFP	GM	f-OGM	b-OGM	BCT	CTW	BFDP
0.01	0.0007	0.0007	0.0007	0.0007	0.0007	0.0007	0.0007	0.0007
0.05	0.0046	0.0045	0.0043	0.0044	0.0044	0.0043	0.0046	0.0043
0.10	0.0100	0.0116	0.0100	0.0102	0.0102	0.0100	0.0100	0.0100
0.15	0.0100	0.0100	0.0100	0.0104	0.0104	0.0100	0.0100	0.0100
0.20	0.0100	0.0100	0.0100	0.0101	0.0101	0.0100	0.0100	0.0100

TABLE II
FIRST-ORDER MARKOV SOURCE (p) AND BSC (δ)

$\delta = 0.10$								
p	DUDE	BFP	GM	f-OGM	b-OGM	BCT	CTW	BFDP
0.01	0.0070	0.0063	0.0066	0.0069	0.0070	0.0065	0.0063	0.0058
0.05	0.0306	0.0303	0.0301	0.0305	0.0304	0.0302	0.0299	0.0298
0.10	0.0566	0.0562	0.0567	0.0570	0.0570	0.0561	0.0561	0.0559
0.15	0.0799	0.0758	0.0752	0.0772	0.0773	0.0750	0.0752	0.0750
0.20	0.0924	0.0934	0.0924	0.0964	0.0963	0.0924	0.0925	0.0924

have constant number of nodes. In the bidirectional models BFP, GM, f-OGM, and b-OGM that are based on the unidirectional VLMC model, the time complexity $O(n \log n)$ [12] for constructing VLMC can be made linear by utilizing suffix trees and finite-state machine (FSM) [42], [43]. After constructing the FSM for z_1^n (in f-OGM) or the FSM for z_n^1 (in b-OGM) or both (in BFP and GM) within linear time, we store the FSM states of z_1^{j-1} , $j = 1, 2, \dots, n$ (in f-OGM) or z_{j+1}^n , $j = n, n-1, \dots, 1$ (in b-OGM) or both (in BFP and GM). This is done by transiting from one state to another, which costs linear time since one state transition needs constant time. BFP, GM, f-OGM, and b-OGM (cf. (6), (16), (24), and (25), respectively) all involve products of constant number of unidirectional conditional probabilities for each j , which can be obtained in a constant number of state transitions that start at states in the above stored state sequences (z_1^{j-1} and z_{j+1}^n for BFP, z_1^{j-m-1} and z_{j+m+1}^n for GM, z_1^{j-1} for f-OGM, z_{j+1}^n for b-OGM). Each state gives its corresponding estimate of the unidirectional conditional distribution. If one state transition is not defined (because the symbol a is arbitrary in (6), (16), (24), and (25) for each j), we go directly to the state of the empty string ϵ , then the current and later state transitions can be continued. If the current transition from ϵ is not defined, we use the state ϵ as the next state and continue subsequent transitions. This does not affect the overall constant time complexity of state transitions needed for each j . In addition, GM needs the DUDE estimates (17) which can be done in linear time and space [1]. Thus, the overall time complexities of BFP, GM, f-OGM, and b-OGM are all $O(n)$. The space complexities for BFP, GM, f-OGM, and b-OGM are all linear [43] because the storage space for the FSM(s) and the saved states are $O(n)$. Another way to resolve the issue of undefined state transitions is to set the “closest” valid state to be the next state. For example, if the transition from z_1^{j-1} to $z_1^{j-1}a$ is undefined, then we search for the state(s) of $z_i^{j-1}a$, $i = 2, 3, \dots, j$ until one valid state is found, which is set to be the next state. The search is performed more efficiently on the optimal full tree than in the FSM, since within time $O(j)$ we can search from a back to z_i such that $z_i^{j-1}a$ is

the deepest node. This would give better performance because generally longer memory is exploited, but the overall time complexity would then be $O(n^2)$. The latter way of finding and approximating the next state requires quite reasonable time complexity in practice, thus, it is used in the experiments as well as the simple $O(n \log n)$ implementation of VLMC [12].

B. Optimal Performance of a Model By the Compression Heuristic

Since the underlying true input sequence to the channel is unknown, in practice it is impossible to choose parameters according to the true loss induced by a denoiser under given parameters. Instead, the compression heuristic [1] can be used to tune the parameters. The compression heuristic says that the optimal parameters are achieved when the denoiser outputs a sequence that is shortest after universal lossless compression. As all experiments indicate, the performances of denoisers whose parameters are given by the compression heuristic are very close to the performances of denoisers which use optimal parameters with the aid of a genie that can access the true performances of the denoisers. In the English text experiment, we demonstrate the performances of denoisers whose parameters are chosen by both the compression heuristic and the genie, which turns out to strongly support the use of the compression heuristic.

C. Binary Markov Source and BSC

The Bayesian optimal denoiser can be exactly implemented via backward–forward dynamic programming (BFDP). A binary Markov source with transition probability p inputs a binary sequence of length 10^6 into a BSC with crossover probability δ . Tables I–III display the bit-error rates after denoising for different source/channel parameters (p, δ) and different bidirectional modeling algorithms.

D. English Text Correction

In this individual sequence setting, an English text goes through a DMC called the keyboard channel, which mimics a

TABLE III
FIRST-ORDER MARKOV SOURCE (p) AND BSC (δ)

$\delta = 0.20$								
p	DUDE	BFP	GM	f-OGM	b-OGM	BCT	CTW	BFDP
0.01	0.0333	0.0219	0.0233	0.0238	0.0231	0.0245	0.0215	0.0160
0.05	0.0776	0.0739	0.0739	0.0739	0.0743	0.0741	0.0730	0.0713
0.10	0.1217	0.1203	0.1203	0.1189	0.1149	0.1203	0.1199	0.1192
0.15	0.1542	0.1538	0.1541	0.1552	0.1551	0.1539	0.1538	0.1535
0.20	0.1796	0.1766	0.1765	0.1799	0.1800	0.1763	0.1762	0.1762

TABLE IV
DENOISING FOR ENGLISH TEXTS AND KEYBOARD CHANNEL: COMPRESSION HEURISTIC

No.	DUDE	BFP	GM	f-OGM	b-OGM	CTW	BCT
1	0.022479	0.0137131	0.0112764	0.0110296	0.0110676	0.0101708	0.0170538
2	0.022492	0.0128292	0.0111544	0.0104394	0.010695	0.00953467	0.0171619
3	0.022555	0.0142284	0.0119915	0.0116298	0.012021	0.0108227	0.0194086
4	0.021422	0.0128954	0.0105986	0.0102949	0.0104274	0.0093896	0.0169460
5	0.024042	0.0159728	0.0129864	0.0127082	0.0131825	0.0119023	0.0208239
6	0.021544	0.0138857	0.0111848	0.0112337	0.0112436	0.0107524	0.0176427
7	0.022778	0.0134215	0.0107607	0.0105639	0.0108004	0.00952405	0.0177708
8	0.024341	0.0169152	0.0137873	0.0137048	0.0137267	0.0122522	0.0212502
9	0.024225	0.0161501	0.0133945	0.0132311	0.0131803	0.0126302	0.0214963
10	0.022794	0.0132631	0.0100639	0.0103681	0.0106855	0.00954954	0.0180168
11	0.021034	0.0127723	0.0100843	0.0100739	0.0101516	0.00941619	0.0169560
12	0.024074	0.0141297	0.0112091	0.0114807	0.0112891	0.010163	0.0193032
13	0.023044	0.0136409	0.0113392	0.0108458	0.0111684	0.00999445	0.0179127
14	0.022882	0.0134861	0.0109549	0.0110448	0.0110053	0.00994375	0.0191298
15	0.023871	0.0168967	0.0146290	0.0136877	0.0136748	0.0129726	0.0213615

typist's most likely errors. Here $M = 128$, and the channel transition probability matrix $\mathbf{\Pi}$ is identical to the one in [1], which corrupts letters with probability 0.05. Fifteen novels have been tried (using data from Project Gutenberg: <http://www.gutenberg.org/>):

1. *Don Quixote*
2. *David Copperfield*
3. *Great Expectations*
4. *Household Tales by Brothers Grimm*
5. *Jane Eyre*
6. *Les Miserables*
7. *Little Dorrit*
8. *Micah Clarke*
9. *Notre-Dame de Paris*
10. *The Arabian Nights*
11. *The Count of Monte Cristo*
12. *The Great Boer War*
13. *The Life and Adventures of Nicholas Nickleby*
14. *The Moonstone*
15. *The White Company*

The 15 novels are also of varying sizes, ranging from about 8×10^5 bytes to about 3.3×10^6 bytes. Table IV shows the error rates after denoising for compression heuristic-based denoisers under different models in the English text case. The usefulness of compression heuristic is demonstrated by Table V, which displays error rates after denoising for genie-aided denoisers under different models in the English text case. As we can see, Ta-

bles V and IV have very close entries for the same novels and in many cases they even have the same entries, in favor of compression heuristic. For novel No. 1 (*Don Quixote*), [23] cites an error rate about 0.01655 after denoising, which is comparable to our BCT, but worse than BFP, GM, f-OGM, b-OGM, and CTW.

E. Best Model by the Compression Heuristic

Not only can the compression heuristic be used in choosing near-optimal parameters for a specific model, but also it can be used in choosing a best model among several models. Note that in the compression heuristic-based experiment depicted in Table IV, CTW yields the lowest error rate in all cases. When the compression heuristic is used to choose among DUDE, BFP, GM, f-OGM, b-OGM, CTW, and BCT, it also chooses CTW in almost all cases.

F. Which Model?

In conclusion, we have introduced six bidirectional models BFP, GM, f-OGM, b-OGM, CTW, and BCT. While BFP, GM, f-OGM, b-OGM, and CTW are induced from unidirectional models, BCT is based on the construction of a bidirectional context tree. CTW is based on a quite different principle of mixing all submodels. Occasionally, BFP performs worse than DUDE for the binary Markov source and BSC experiment, but in general, all the proposed models show a slight improvement in that setting. Much bigger improvements over DUDE are observed in the case of corrupted English text. Although GM, f-OGM,

TABLE V
DENOISING FOR ENGLISH TEXTS AND KEYBOARD CHANNEL: GENIE-AIDED

No.	DUDE	BFP	GM	f-OGM	b-OGM	CTW	BCT
1	0.021236	0.0131625	0.0103657	0.010719	0.0106585	0.0101708	0.0170538
2	0.021624	0.0126865	0.0102137	0.0102647	0.0103818	0.00953467	0.0172049
3	0.022555	0.0140889	0.0112870	0.0113829	0.0116507	0.0108227	0.0193545
4	0.021331	0.0125563	0.0102396	0.0101969	0.0102469	0.0093896	0.0169294
5	0.024042	0.0155329	0.0123253	0.0124588	0.0126839	0.0119023	0.0208239
6	0.021544	0.0133998	0.0108693	0.010887	0.0107797	0.0107524	0.0176454
7	0.022466	0.0130119	0.0102166	0.0103073	0.0104794	0.00952405	0.0177708
8	0.024341	0.0163097	0.013149	0.0135745	0.0132196	0.0122522	0.0212084
9	0.024225	0.0156092	0.0127555	0.0129887	0.012878	0.0126302	0.0214754
10	0.022794	0.0128845	0.0100248	0.010119	0.0103589	0.00954954	0.0180168
11	0.021034	0.0124139	0.00967979	0.00978352	0.00982813	0.00941619	0.0169560
12	0.024074	0.0135821	0.0108765	0.0112024	0.0111526	0.010163	0.0192745
13	0.022751	0.013262	0.0105594	0.0105609	0.0106734	0.00999445	0.0179048
14	0.022882	0.0133548	0.0105595	0.0107914	0.0108606	0.00994375	0.0191298
15	0.023871	0.0162488	0.0132884	0.0135228	0.0134427	0.0129726	0.0213297

b-OGM, and CTW all show excellent behavior, from the experiments we can see that none of the six proposed methods is uniformly the best. Using a compression heuristic it is usually possible to choose the best modeler for each particular application.

APPENDIX I
PROOF OF LEMMA 1

Proof: Define

$$\tilde{Q}_{i,e}^s(Y_1^i(n)) \triangleq \frac{\tilde{P}_e^s(i,n)}{\tilde{P}_e^s(i-1,n)}, \tilde{Q}_{i,w}^s(Y_1^i(n)) \triangleq \frac{\tilde{P}_w^s(i,n)}{\tilde{P}_w^s(i-1,n)}$$

for $\forall i \geq D+2$. Note that if s is on the updating path for the i th symbol $Y_i(n)$, namely, $s = Y_{i-d}^{i-1}(n)$ for some $0 \leq d \leq D$, then

$$\tilde{Q}_{i,e}^s(Y_1^i(n)) = \frac{\tilde{P}_e^s(i,n)}{\tilde{P}_e^s(i-1,n)} = \frac{\tilde{N}_s^{i-1,n}(Y_i(n)) + \frac{1}{2}}{\sum_{b \in \mathcal{B}} \tilde{N}_s^{i-1,n}(b) + \frac{1}{2}|\mathcal{B}|} \quad (44)$$

according to the CTW algorithm.

At any time $i \geq D+2$, consider the updating path $\{v \rightarrow \mathbf{e}\}$ for $Y_i(n)$. Let $s \in \{v \rightarrow \mathbf{e}\}$ be any internal node (i.e., any node with depth strictly less than D) along this path, let s^* be the unique child node of s that is on this path, i.e., the unique $s^* \in C(s) \cap \{v \rightarrow \mathbf{e}\}$. Then

$$\begin{aligned} \tilde{Q}_{i,w}^s(Y_1^i(n)) &= \frac{\tilde{P}_w^s(i,n)}{\tilde{P}_w^s(i-1,n)} \\ &= \frac{\frac{1}{2}\tilde{P}_e^s(i,n) + \frac{1}{2} \prod_{s' \in C(s)} \tilde{P}_w^{s'}(i,n)}{\tilde{P}_w^s(i-1,n)} \\ &\stackrel{(44)}{=} \frac{\tilde{P}_e^s(i-1,n)}{2\tilde{P}_w^s(i-1,n)} \frac{\tilde{N}_s^{i-1,n}(Y_i(n)) + \frac{1}{2}}{\sum_{b \in \mathcal{B}} \tilde{N}_s^{i-1,n}(b) + \frac{1}{2}|\mathcal{B}|} \\ &\quad + \frac{\frac{1}{2} \prod_{s' \in C(s) \setminus \{v \rightarrow \mathbf{e}\}} \tilde{P}_w^{s'}(i,n) \tilde{P}_w^{s^*}(i,n)}{\tilde{P}_w^s(i-1,n)} \end{aligned}$$

$$\begin{aligned} &= \frac{\tilde{\beta}_{i-1,n}^s}{\tilde{\beta}_{i-1,n}^s + 1} \tilde{Q}_{i,e}^s(Y_1^i(n)) + \\ &\quad \frac{\prod_{s' \in C(s)} \tilde{P}_w^{s'}(i-1,n)}{2\tilde{P}_w^s(i-1,n)} \frac{\tilde{P}_w^{s^*}(i,n)}{\tilde{P}_w^{s^*}(i-1,n)} \quad (45) \end{aligned}$$

$$\begin{aligned} &= \frac{\tilde{\beta}_{i-1,n}^s}{\tilde{\beta}_{i-1,n}^s + 1} \tilde{Q}_{i,e}^s(Y_1^i(n)) + \\ &\quad \frac{1}{\tilde{\beta}_{i-1,n}^s + 1} \frac{\tilde{P}_w^{s^*}(i,n)}{\tilde{P}_w^{s^*}(i-1,n)} \\ &= \frac{\tilde{\beta}_{i-1,n}^s}{\tilde{\beta}_{i-1,n}^s + 1} \tilde{Q}_{i,e}^s(Y_1^i(n)) + \\ &\quad \frac{1}{\tilde{\beta}_{i-1,n}^s + 1} \tilde{Q}_{i,w}^{s^*}(Y_1^i(n)) \quad (46) \end{aligned}$$

where (45) is from the fact that if $s' \in C(s)$ is not on the updating path for $Y_i(n)$, then

$$\tilde{P}_w^{s'}(i,n) = \tilde{P}_w^{s'}(i-1,n), \tilde{P}_e^{s'}(i,n) = \tilde{P}_e^{s'}(i-1,n).$$

For the leaf node $v = Y_{i-D}^{i-1}(n)$ on the updating path for $Y_i(n)$, we have

$$\begin{aligned} \tilde{Q}_{i,w}^v(Y_1^i(n)) &= \frac{\tilde{P}_w^v(i,n)}{\tilde{P}_w^v(i-1,n)} \\ &= \frac{\tilde{P}_e^v(i,n)}{\tilde{P}_e^v(i-1,n)} \\ &= \tilde{Q}_{i,e}^v(Y_1^i(n)) \\ &\stackrel{(44)}{=} \frac{\tilde{N}_v^{i-1,n}(Y_i(n)) + \frac{1}{2}}{\sum_{b \in \mathcal{B}} \tilde{N}_v^{i-1,n}(b) + \frac{1}{2}|\mathcal{B}|}. \quad (47) \end{aligned}$$

By (47) and recursively using (46), we can get

$$\tilde{Q}_{i,w}^e(Y_1^i(n)) = \hat{P}_i(Y_1^{i-1}(n))[Y_i(n)], i \geq D+2. \quad (48)$$

It is easy to see that $\hat{P}_i(Y_1^{i-1}(n)) \in [0, 1]^{|\mathcal{B}|}$ is indeed a probability distribution because the coefficients $(\alpha_{i-1,n}^s, \gamma_{i-1,n})$ in (27) sum up to 1 and

$$\frac{\tilde{N}_s^{i-1,n}(a) + \frac{1}{2}}{\sum_{b \in \mathcal{B}} \tilde{N}_s^{i-1,n}(b) + \frac{1}{2}|\mathcal{B}|}, \frac{\tilde{N}_v^{i-1,n}(a) + \frac{1}{2}}{\sum_{b \in \mathcal{B}} \tilde{N}_v^{i-1,n}(b) + \frac{1}{2}|\mathcal{B}|}, \quad \forall a \in \mathcal{B}$$

are probability distributions. By (48) and the definition of $\tilde{Q}_{i,w}^e(Y_1^i(n))$, the result follows. \square

APPENDIX II PROOF OF PROPOSITION 1

We first establish the following lemma.

Lemma 2: Let Z be stationary ergodic. Let $s = a_{-m}^{-1} \in \mathcal{B}^m$ with $m < D$ be such that there exists $a_{-D}^{-1} \in \mathcal{B}^D$ with $P_{Z_{-D}^{-1}}(a_{-D}^{-1}) > 0$ and a different unidirectional conditional distribution, i.e., $\exists a_0 \in \mathcal{B}, a_{-D}^{-m-1} \in \mathcal{B}^{D-m}$

$$P_{Z_0|Z_{-m}^{-1}}(a_0|a_{-m}^{-1}) \neq P_{Z_0|Z_{-D}^{-1}}(a_0|a_{-D}^{-1}).$$

(Note that $P_{Z_{-D}^{-1}}(a_{-D}^{-1}) > 0$ implies $P_{Z_{-m}^{-1}}(a_{-m}^{-1}) > 0$.)

Then for any $n \in \mathbb{N}^*$

$$\lim_{i \rightarrow \infty} \tilde{\beta}_{i,n}^s = 0 \quad \text{a.s.}$$

Proof:

$$\begin{aligned} \tilde{\beta}_{i,n}^s &= \frac{\tilde{P}_e^s(i, n)}{\prod_{s_1 \in C(s)} \tilde{P}_w^{s_1}(i, n)} \\ &= \frac{\tilde{P}_e^s(i, n)}{\prod_{s_1 \in C(s)} \left(\frac{1}{2} \tilde{P}_e^{s_1}(i, n) + \frac{1}{2} \prod_{s_2 \in C(s_1)} \tilde{P}_w^{s_2}(i, n) \right)} \\ &\leq \frac{\tilde{P}_e^s(i, n)}{\prod_{s_1 \in C(s)} \left(\frac{1}{2} \prod_{s_2 \in C(s_1)} \tilde{P}_w^{s_2}(i, n) \right)} \\ &= 2^{|\mathcal{B}|} \frac{\tilde{P}_e^s(i, n)}{\prod_{s_1 \in C(s)} \prod_{s_2 \in C(s_1)} \tilde{P}_w^{s_2}(i, n)}. \end{aligned}$$

And by recursively applying the definition of $\tilde{P}_w^q(i, n)$ for any node q as above until we have met the leaf nodes, with $l(q)$ denoting the depth of any node q and $L_D(s)$ denoting all the leaf nodes of the same depth D in the subtree rooted at node s , we finally get

$$\begin{aligned} \tilde{\beta}_i^s &\leq 2^{(D-l(s)-1)|\mathcal{B}|} \frac{\tilde{P}_e^s(i, n)}{\prod_{s' \in L_D(s)} \tilde{P}_w^{s'}(i, n)} \\ &= 2^{(D-l(s)-1)|\mathcal{B}|} \frac{\tilde{P}_e^s(i, n)}{\prod_{s' \in L_D(s)} \tilde{P}_e^{s'}(i, n)} \\ &= 2^{(D-l(s)-1)|\mathcal{B}|} \exp \left\{ -i \left[-\frac{1}{i} \log \tilde{P}_e^s(i, n) \right. \right. \\ &\quad \left. \left. + \frac{1}{i} \sum_{s' \in L_D(s)} \log \tilde{P}_e^{s'}(i, n) \right] \right\} \\ &= 2^{(D-l(s)-1)|\mathcal{B}|} \exp \left\{ -i \left[-\frac{1}{i} \log \tilde{P}_e^s(i, n) \right. \right. \end{aligned}$$

$$\begin{aligned} &\left. + \frac{1}{i} \sum_{s' \in L_D(s): P_{Z_{-D}^{-1}}(s') > 0} \log \tilde{P}_e^{s'}(i, n) \right. \\ &\left. + \frac{1}{i} \sum_{s' \in L_D^*(s): P_{Z_{-D}^{-1}}(s') = 0} \log \tilde{P}_e^{s'}(i, n) \right\} \end{aligned} \quad (49)$$

where

$$L_D^*(s) \triangleq \left\{ s' \in L_D(s) : \sum_{c \in \mathcal{B}} \tilde{N}_{s'}^{i,n}(c) \geq 1 \right\}$$

since $\tilde{P}_e^{s'}(i, n) = 1$ if $\sum_{c \in \mathcal{B}} \tilde{N}_{s'}^{i,n}(c) = 0$.

For any $q = b_{-k}^{-1} \in \mathcal{B}^k$:

$$\lim_{i \rightarrow \infty} \frac{1}{i} \sum_{c \in \mathcal{B}} \tilde{N}_q^{i,n}(c) \stackrel{(26)}{=} \lim_{i \rightarrow \infty} \frac{1}{i} \sum_{c \in \mathcal{B}} N_q^i(c) = P_{Z_{-k}^{-1}}(b_{-k}^{-1}),$$

thus, if $P_{Z_{-k}^{-1}}(b_{-k}^{-1}) > 0$, then for large enough i ,

$$\sum_{c \in \mathcal{B}} \tilde{N}_q^{i,n}(c) \geq 1 \quad \text{a.s.} \quad (50)$$

Notice for any node $q = b_{-k}^{-1} \in \mathcal{B}^k$ such that

$$\sum_{c \in \mathcal{B}} \tilde{N}_q^{i,n}(c) \geq 1$$

$\tilde{P}_e^q(i, n)$ is a KT estimator [13], we have [13, Sec. II, eq. (2.6)] (which is based on Stirling's formula for Gamma functions)

$$\left| -\log \tilde{P}_e^q(i, n) - \left(\sum_{c \in \mathcal{B}} \tilde{N}_q^{i,n}(c) \hat{H}^{i,n}(q) - \frac{|\mathcal{B}| - 1}{2} \log \left(\sum_{c \in \mathcal{B}} \tilde{N}_q^{i,n}(c) \right) \right) \right| \leq C \quad (51)$$

where C is a strictly positive constant that does not depend on i, n , and

$$\hat{H}^{i,n}(q) = \hat{H}^{i,n}(b_{-k}^{-1}) \triangleq \sum_{a \in \mathcal{B}} \frac{\tilde{N}_q^{i,n}(a)}{\sum_{c \in \mathcal{B}} \tilde{N}_q^{i,n}(c)} \log \frac{\sum_{c \in \mathcal{B}} \tilde{N}_q^{i,n}(c)}{\tilde{N}_q^{i,n}(a)}.$$

By the stationarity and ergodicity of Z :

$$\begin{aligned} \lim_{i \rightarrow \infty} \hat{H}^{i,n}(q) &= \lim_{i \rightarrow \infty} \hat{H}^{i,n}(b_{-k}^{-1}) \\ &= \lim_{i \rightarrow \infty} \sum_{a \in \mathcal{B}} \frac{\tilde{N}_q^{i,n}(a)}{\sum_{c \in \mathcal{B}} \tilde{N}_q^{i,n}(c)} \log \frac{\sum_{c \in \mathcal{B}} \tilde{N}_q^{i,n}(c)}{\tilde{N}_q^{i,n}(a)} \\ &\stackrel{(26)}{=} \sum_{a \in \mathcal{B}} \lim_{i \rightarrow \infty} \frac{N_q^i(a)}{\sum_{c \in \mathcal{A}} N_q^i(c)} \log \left(\lim_{i \rightarrow \infty} \frac{\sum_{c \in \mathcal{B}} N_q^i(c)}{N_q^i(a)} \right) \\ &= H(Z_0|Z_{-k}^{-1} = b_{-k}^{-1}) \quad \text{a.s.} \end{aligned} \quad (52)$$

Hence by (51), note

$$\sum_{c \in \mathcal{B}} \tilde{N}_q^{i,n}(c) \leq i, \quad \forall q = b_{-k}^{-1} \in \mathcal{B}^k,$$

with $\sum_{c \in \mathcal{B}} \tilde{N}_q^{i,n}(c) \geq 1$, almost surely we have

$$\begin{aligned} -\lim_{i \rightarrow \infty} \frac{1}{i} \log \tilde{P}_e^q(i, n) &= \lim_{i \rightarrow \infty} \frac{\sum_{c \in \mathcal{B}} \tilde{N}_q^{i,n}(c)}{i} \hat{H}^{i,n}(b_{-k}^{-1}) \\ &\stackrel{(26)}{=} \lim_{i \rightarrow \infty} \frac{\sum_{c \in \mathcal{B}} N_q^i(c)}{i} \lim_{i \rightarrow \infty} \hat{H}^{i,n}(b_{-k}^{-1}) \\ &= P_{Z_{-k}^{-1}}(b_{-k}^{-1}) H(Z_0 | Z_{-k}^{-1} = b_{-k}^{-1}) \end{aligned} \quad (53)$$

by the ergodicity of Z again. Note $\forall b_{-D}^0 \in \mathcal{B}^{D+1}$ with $P_{Z_{-m}^{-1}}(b_{-m}^{-1}) > 0$:

$$\begin{aligned} &P_{Z_0 | Z_{-m}^{-1}}(b_0 | b_{-m}^{-1}) \\ &= \sum_{b_{-D}^{-m-1} \in \mathcal{B}^{D-m}: P_{Z_{-D}^{-1}}(b_{-D}^{-1}) > 0} \left[P_{Z_{-D}^{-m-1} | Z_{-m}^{-1}}(b_{-D}^{-m-1} | b_{-m}^{-1}) \right. \\ &\quad \left. \times P_{Z_0 | Z_{-D}^{-1}}(b_0 | b_{-D}^{-1}) \right], \end{aligned}$$

whence, by the strict concavity of entropy, because $P_{Z_{-D}^{-1}}(a_{-D}^{-1}) > 0$ implies $P_{Z_{-m}^{-1}}(a_{-m}^{-1}) > 0$:

$$\begin{aligned} &H(Z_0 | Z_{-m}^{-1} = a_{-m}^{-1}) \geq \\ &\sum_{b_{-D}^{-m-1}: P_{Z_{-D}^{-1}}(b_{-D}^{-1} | a_{-m}^{-1}) > 0} \left[P_{Z_{-D}^{-m-1} | Z_{-m}^{-1}}(b_{-D}^{-m-1} | a_{-m}^{-1}) \right. \\ &\quad \left. \times H(Z_0 | Z_{-D}^{-1} = b_{-D}^{-m-1} a_{-m}^{-1}) \right] \end{aligned} \quad (54)$$

with equality if and only if for any $b_0 \in \mathcal{B}, b_{-D}^{-m-1} \in \mathcal{B}^{D-m}$ with $P_{Z_{-D}^{-1}}(b_{-D}^{-m-1} a_{-m}^{-1}) > 0$, we have

$$P_{Z_0 | Z_{-m}^{-1}}(b_0 | a_{-m}^{-1}) = P_{Z_0 | Z_{-D}^{-1}}(b_0 | b_{-D}^{-m-1} a_{-m}^{-1}).$$

But the equality cannot hold for any $b_0 \in \mathcal{B}, b_{-D}^{-m-1} \in \mathcal{B}^{D-m}$ with $P_{Z_{-D}^{-1}}(b_{-D}^{-m-1} a_{-m}^{-1}) > 0$, since it then contradicts our assumption. Thus, strict inequality in (54) holds.

Hence, by (53), almost surely the exponent on the right-hand side of (49) becomes (note (50) holds for $q = s$ and for any $q = s' \in L_D(s)$ with $P_{Z_{-1}^{-1}}(s') > 0$, when i is large enough)

$$\begin{aligned} &\lim_{i \rightarrow \infty} \left[-\frac{1}{i} \log \tilde{P}_e^s(i, n) + \frac{1}{i} \sum_{s' \in L_D(s)} \log \tilde{P}_e^{s'}(i, n) \right] \\ &= P_{Z_{-m}^{-1}}(a_{-m}^{-1}) H(Z_0 | Z_{-m}^{-1} = a_{-m}^{-1}) \\ &\quad - \sum_{b_{-D}^{-m-1}: P_{Z_{-D}^{-1}}(b_{-D}^{-m-1} a_{-m}^{-1}) > 0} \left[P_{Z_{-D}^{-1}}(b_{-D}^{-m-1} a_{-m}^{-1}) \right. \\ &\quad \left. \times H(Z_0 | Z_{-D}^{-1} = b_{-D}^{-m-1} a_{-m}^{-1}) \right] \\ &= P_{Z_{-m}^{-1}}(a_{-m}^{-1}) \left\{ H(Z_0 | Z_{-m}^{-1} = a_{-m}^{-1}) \right. \\ &\quad \left. - \sum_{b_{-D}^{-m-1}: P_{Z_{-D}^{-1}}(b_{-D}^{-m-1} a_{-m}^{-1}) > 0} \left[P_{Z_{-D}^{-m-1} | Z_{-m}^{-1}}(b_{-D}^{-m-1} | a_{-m}^{-1}) \right. \right. \\ &\quad \left. \left. \times H(Z_0 | Z_{-D}^{-1} = b_{-D}^{-m-1} a_{-m}^{-1}) \right] \right\} \\ &> 0 \end{aligned}$$

is a strictly positive constant, because our assumption implies $P_{Z_{-m}^{-1}}(a_{-m}^{-1}) > 0$ and because of the strict inequality in (54). The result then follows from (49). \square

Proof of Proposition 1: For any $a_{-D}^0 \in \mathcal{B}^{D+1}, i \geq D+2$, on the set $\{Y_{i-D}^i(n) = a_{-D}^0\}$ with $P_{Z_{-D}^{-1}}(a_{-D}^{-1}) > 0$, with the same notations as in Lemma 1, $\forall a \in \tilde{\mathcal{B}}$:

$$\begin{aligned} \hat{P}_i(Y_1^{i-1}(n)) [a_0] &= \gamma_{i-1,n} \frac{\tilde{N}_v^{i-1,n}(a_0) + \frac{1}{2}}{\sum_{b \in \mathcal{B}} \tilde{N}_v^{i-1,n}(b) + \frac{1}{2} |\mathcal{B}|} \\ &+ \sum_{s=a_{-d}^{-1}: d=0,1,\dots,D-1} \alpha_{i-1,n}^s \frac{\tilde{N}_s^{i-1,n}(a_0) + \frac{1}{2}}{\sum_{b \in \mathcal{B}} \tilde{N}_s^{i-1,n}(b) + \frac{1}{2} |\mathcal{B}|} \end{aligned} \quad (55)$$

where $v = a_{-D}^{-1}$, with

$$\begin{aligned} \gamma_{i-1,n} &= \prod_{q=a_{-d}^{-1}: d=0,1,\dots,D-1} \frac{1}{\tilde{\beta}_{i-1,n}^q + 1}, \\ \alpha_{i-1,n}^s &= \tilde{\beta}_{i-1,n}^s \prod_{q=a_{-k}^{-1}: k=0,1,\dots,d} \frac{1}{\tilde{\beta}_{i-1,n}^q + 1} \end{aligned}$$

for node $s = a_{-d}^{-1}$. Now let $0 \leq j \leq D$ be the smallest integer such that

$$P_{Z_0 | Z_{-j}^{-1}}(b_0 | a_{-j}^{-1}) = P_{Z_0 | Z_{-D}^{-1}}(b_0 | b_{-D}^{-j-1} a_{-j}^{-1})$$

for any $b_0 \in \mathcal{B}, b_{-D}^{-j-1} \in \mathcal{B}^{D-j}$. Obviously, such an integer always exists since D always satisfies above equation. Thus, it is obvious that

Case 1: for each node $q = a_{-m}^{-1}$ such that its depth $m \triangleq l(q) \geq j$, we have

$$\begin{aligned} P_{Z_0 | Z_{-m}^{-1}}(a_0 | a_{-m}^{-1}) &= P_{Z_0 | Z_{-j}^{-1}}(a_0 | a_{-j}^{-1}) \\ &= P_{Z_0 | Z_{-D}^{-1}}(a_0 | a_{-D}^{-1}), \end{aligned}$$

thus,

$$\begin{aligned} &\lim_{i \rightarrow \infty} \left[P_{Z_0 | Z_{-D}^{-1}}(a_0 | a_{-D}^{-1}) - \frac{\tilde{N}_q^{i-1,n}(a_0) + \frac{1}{2}}{\sum_{c \in \mathcal{B}} \tilde{N}_q^{i-1,n}(c) + \frac{1}{2} |\mathcal{B}|} \right] \\ &\stackrel{(26)}{=} P_{Z_0 | Z_{-m}^{-1}}(a_0 | a_{-m}^{-1}) - \lim_{i \rightarrow \infty} \frac{N_q^{i-1}(a_0)}{\sum_{c \in \mathcal{B}} N_q^{i-1}(c)} = 0 \text{ a.s.} \end{aligned}$$

by the stationarity and ergodicity of Z ;

Case 2: for each node $q = a_{-m}^{-1}$ such that its depth $m \triangleq l(q) < j$, the condition in Lemma 2 is satisfied by the definition of the integer j , thus,

$$\lim_{i \rightarrow \infty} \tilde{\beta}_{i-1,n}^q = 0 \text{ a.s.};$$

therefore, for any such node q

$$\lim_{i \rightarrow \infty} \alpha_{i-1,n}^q = 0 \text{ a.s.} \quad (56)$$

In view of (55), notice there are only $(D+1)$ terms, and coefficients ($\alpha_{i-1,n}^s, \gamma_{i-1,n}$) of those terms (estimated probabilities)

$$\frac{\tilde{N}_s^{i-1,n}(a_0) + \frac{1}{2}}{\sum_{c \in \mathcal{B}} \tilde{N}_s^{i-1,n}(c) + \frac{1}{2} |\mathcal{B}|}$$

sum up to 1, by the above *Case 1* and *Case 2*, we see immediately that almost surely as $i \rightarrow \infty$

$$\begin{aligned} & \left| \hat{P}_i(Y_1^{i-1}(n))[a_0] - P_{Z_0|Z_{-D}^{-1}}(a_0|a_{-D}^{-1}) \right| 1_{Y_{i-D}^i(n)=a_{-D}^0} \rightarrow 0 \\ \text{for any } & a_{-D}^0 \in \mathcal{B}^{D+1} \text{ such that } P_{Z_0|Z_{-D}^{-1}}(a_{-D}^{-1}) > 0. \text{ Thus,} \\ & \left\{ \left| \hat{P}_i(Y_1^{i-1}(n))[Y_i(n)] - P_{Z_0|Z_{-D}^{-1}}(Y_i(n)|Y_{i-D}^{i-1}(n)) \right| \right. \\ & \quad \left. \times 1_{P_{Z_{-D}^{-1}}(Y_{i-D}^{i-1}(n))>0} \right\} \\ & = \sum_{a_{-D}^0 \in \mathcal{B}^{D+1}} \left\{ \left| \hat{P}_i(Y_1^{i-1}(n))[a_0] - P_{Z_0|Z_{-D}^{-1}}(a_0|a_{-D}^{-1}) \right| \right. \\ & \quad \left. \times 1_{Y_{i-D}^i(n)=a_{-D}^0} 1_{P_{Z_{-D}^{-1}}(a_{-D}^{-1})>0} \right\} \\ & \rightarrow 0 \text{ a.s.} \end{aligned}$$

as $i \rightarrow \infty$. □

ACKNOWLEDGMENT

The paper has benefitted from comments by Tsachy Weissman on an earlier draft.

REFERENCES

[1] T. Weissman, E. Ordentlich, G. Seroussi, S. Verdú, and M. Weinberger, "Universal discrete denoising: Known channel," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 5–28, Jan. 2005.

[2] F. M. J. Willems, Y. M. Shtarkov, and T. J. Tjalkens, "The context-tree weighting method: Basic properties," *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 653–664, May 1995.

[3] F. M. J. Willems, "The context-tree weighting method: Extensions," *IEEE Trans. Inf. Theory*, vol. 44, no. 2, pp. 792–798, Mar. 1998.

[4] T. Weissman and N. Merhav, "Universal prediction of individual binary sequences in the presence of noise," *IEEE Trans. Inf. Theory*, vol. 47, no. 6, pp. 2151–2173, Sep. 2001.

[5] —, "Universal prediction of random binary sequences in a noisy environment," *Ann. Appl. Probab.*, vol. 14, no. 1, pp. 54–89, 2004.

[6] E. Ordentlich, T. Weissman, M. J. Weinberger, A. Somekh-Baruch, and N. Merhav, "Discrete universal filtering through incremental parsing," in *Proc. IEEE Data Compression Conf.*, Snowbird, UT, Mar. 2004, pp. 352–361.

[7] E. Ordentlich, G. Seroussi, S. Verdú, and K. Viswanathan, "Universal Algorithms for Channel Decoding of Uncompressed Sources", submitted for publication.

[8] J. Rissanen, "A universal data compression system," *IEEE Trans. Inf. Theory*, vol. IT-29, no. 5, pp. 656–664, Sep. 1983.

[9] M. J. Weinberger, J. J. Rissanen, and M. Feder, "A universal finite memory source," *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 643–652, May 1995.

[10] P. Bühlmann and A. J. Wyner, "Variable length Markov chains," *Ann. Statist.*, vol. 27, pp. 480–513, 1999.

[11] F. Ferrari and A. J. Wyner, "Estimation of general stationary processes by variable length Markov chains," *Scand. J. Statist.*, pp. 459–480, Sep. 2003.

[12] M. Mächler and P. Bühlmann, "Variable length Markov chains: Methodology, computing and software," *J. Comput. Graph. Statist.*, pp. 435–455, Jun. 2004.

[13] R. E. Krichevsky and V. K. Trofimov, "The performance of universal encoding," *IEEE Trans. Inf. Theory*, vol. IT-27, no. 2, pp. 199–207, Mar. 1981.

[14] I. Csiszár and Z. Talata, "Context tree estimation for not necessarily finite memory processes, via BIC and MDL," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 1007–1016, Mar. 2006.

[15] F. M. J. Willems, Y. M. Shtarkov, and T. J. Tjalkens, "Context-tree maximizing," in *Proc. 2000 Conf. Information Sciences and Systems*, Princeton, NJ, Mar. 2000.

[16] E. Ordentlich, G. Seroussi, S. Verdú, K. Viswanathan, M. Weinberger, and T. Weissman, "Channel decoding of systematically encoded unknown redundant sources," in *Proc. IEEE Int. Symp. Information Theory*, Chicago, IL, Jun./Jul. 2004, p. 165.

[17] G. M. Gemelos, S. Sigurjonsson, and T. Weissman, "Universal min-max discrete denoising under channel uncertainty," *IEEE Trans. Inf. Theory*, vol. 52, no. 8, pp. 3476–3497, Aug. 2006.

[18] —, "Algorithms for discrete denoising under channel uncertainty," *IEEE Trans. Signal Process.*, vol. 54, no. 6, pp. 2263–2276, Jun. 2006.

[19] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[20] Y. Ephraim and N. Merhav, "Hidden Markov processes," *IEEE Trans. Inf. Theory*, vol. 48, no. 6, pp. 1518–1569, Jun. 2002.

[21] T. Moon and T. Weissman, "Discrete universal filtering via hidden Markov modelling," in *Proc. IEEE Int. Symp. Information Theory*, Adelaide, Australia, Sep. 2005, pp. 1285–1289.

[22] J. Yu and S. Verdú, "Schemes for bidirectional modeling of discrete stationary sources," in *Proc. 39th Annu. Conf. Information Science and Systems*, Baltimore, MD, Mar. 2005.

[23] E. Ordentlich, M. J. Weinberger, and T. Weissman, "Multi-directional context sets with applications to universal denoising and compression," in *Proc. IEEE Int. Symp. Information Theory*, Adelaide, Australia, Sep. 2005, pp. 1270–1274.

[24] —, "Efficient pruning of bidirectional context trees with applications to universal denoising and compression," in *Proc. IEEE Information Theory Workshop*, San Antonio, TX, Oct. 2004, pp. 94–98.

[25] D. Baron and Y. Bresler, "An $O(N)$ semipredictive universal encoder via the BWT," *IEEE Trans. Inf. Theory*, vol. 50, no. 5, pp. 928–937, May 2004.

[26] R. M. Gray, "Information rates of autoregressive processes," *IEEE Trans. Inf. Theory*, vol. IT-16, no. 4, pp. 412–421, Jul. 1970.

[27] —, "Rate distortion functions for finite-state finite-alphabet Markov sources," *IEEE Trans. Inf. Theory*, vol. IT-17, no. 2, pp. 127–134, Mar. 1971.

[28] F. Spitzer, "Markov random fields and Gibbs ensembles," *Amer. Math. Monthly*, vol. 78, no. 2, pp. 142–154, 1971.

[29] —, *Random Fields and Interacting Particle Systems*. Willimstown, MA: Math. Assoc. America, 1971, Notes on lectures given at the 1971 MAA Summer Seminar, Williams College.

[30] J. Besag, "Spatial interaction and the statistical analysis of lattice systems," *J. Roy. Statist. Soc. Ser. B (Methodological)*, vol. 36, no. 2, pp. 192–236, 1974.

[31] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 721–741, Nov. 1984.

[32] H.-O. Georgii, *Gibbs Measures and Phase Transitions*. Berlin, Germany: Water de Gruyter, 1988, vol. 9, de Gruyter Studies in Mathematics.

[33] H. Cai, S. R. Kulkarni, and S. Verdú, "Universal divergence estimation for finite-alphabet sources," *IEEE Trans. Inf. Theory*, vol. 52, no. 8, pp. 3456–3475, Aug. 2006.

[34] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[35] F. M. J. Willems and T. J. Tjalkens, Complexity Reduction of the Context-Tree Weighting Algorithm: A Study for KPN Research, Tech. Univ. Eindhoven, Eindhoven, The Netherlands, 1997, EIDMA Rep. RS.97.01.

[36] P. Volf, "Weighting Techniques in Data Compression: Theory and Algorithm," Ph.D. dissertation, Tech. Univ. Eindhoven, Eindhoven, The Netherlands, 2002.

[37] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, pp. 465–471, 1978.

[38] —, "A universal prior for integers and estimation by minimum description length," *Ann. Statist.*, vol. 11, no. 2, pp. 416–431, 1983.

[39] —, "Universal coding, information, prediction, and estimation," *IEEE Trans. Inf. Theory*, vol. IT-30, no. 4, pp. 629–635, Jul. 1984.

[40] I. Csiszár and Z. Talata, "Consistent estimation of the basic neighborhood of Markov random fields," *Ann. Statist.*, vol. 34, pp. 123–145, Feb. 2006.

[41] G. Schwarz, "Estimating the dimension of a model," *Ann. Statist.*, vol. 6, pp. 461–464, Mar. 1978.

[42] R. Giegerich and S. Kurtz, "From Ukkonen to McCreight and Weiner: A unifying view of linear-time suffix tree construction," *Algorithmica*, vol. 19, pp. 331–353, Nov. 1997.

[43] A. Martín, G. Seroussi, and M. J. Weinberger, "Linear time universal coding and time reversal of tree sources via FSM closure," *IEEE Trans. Inf. Theory*, vol. 50, no. 7, pp. 1442–1468, Jul. 2004.