Effective Prediction Modes Design for Adaptive Compression With Application in Video Coding

Bharath Vishwanath, *Student Member, IEEE*, Tejaswi Nanjundaswamy, *Member, IEEE*, and Kenneth Rose, *Life Fellow, IEEE*

***Abstract*— Adaptive prediction is an important tool for efficient compression of non-stationary signals. A common approach to achieve adaptivity is to switch between a set of prediction modes, designed to capture variations in signal statistics. The design poses several challenges including: *i)* catastrophic instability due to statistical mismatch driven by propagation through the prediction loop, and *ii)* severe non-convexity of the cost surface that is often riddled with poor local minima. Motivated by these challenges, this paper presents a near-optimal method for designing prediction modes for adaptive compression. The proposed method builds on a stable, open-loop platform, but with a subterfuge that ensures that the design is asymptotically optimized for closed-loop operation. The non-convexity is handled by deterministic annealing, a powerful optimization tool to avoid poor local minima. To demonstrate the impact of the proposed approach on practical applications, we consider adaptive, transform-domain predictor design for enhancing standard video coding. Experimental results validate the benefits of the proposed design in terms of significant performance gains for both predictive compression systems in general and video coding in particular.**

***Index Terms*— Predictor design, deterministic annealing, asymptotic closed-loop, video coding, transform domain temporal prediction.**

I. Introduction

LINEAR prediction is an integral part of most modern compression systems [1]–[3], tasked with removing temporal or spatial redundancies in signals. Often, the design of prediction filters assumes that the signal is stationary. However, most real-world signals are non-stationary and naturally call for adaptive compression systems. A common paradigm to achieve adaptivity involves block-based encoding, wherein the source signal is partitioned into blocks and the prediction filters can be adapted per block. However, sending per-block prediction filter specification would incur considerable overhead. Instead, a common, cost-effective approach is to design a 'codebook' of predictors, which is also available to the decoder, and have the encoder convey the index of the predictor (mode) used to predict a given block. The performance gains of such an adaptive system critically depend on efficient design of all prediction modes.

The design of a codebook of prediction modes poses several challenges. The design can be viewed as 'quantization' of the prediction filter parameter space. The cost-function depends on both the codebook and the encoder decisions (assigning codebook entries to individual signal blocks). Note that the cost is piece-wise constant with respect to the encoder decisions (the encoder does not modify a decision until the block content changes sufficiently) which implies that the corresponding derivatives vanish almost everywhere. This makes it impossible to employ standard gradient-based algorithms. A common remedy is to design predictors in a "K-means" clustering fashion [4], wherein the design iterates between choosing the best prediction modes for the blocks (i.e., nearest neighbor rule for the encoding decisions) and then updating the prediction modes (centroid rule). It is well known that the performance of greedy approaches depends on initialization, and there is substantial risk of getting trapped in poor local optima. The prevalence of local optima, coupled with the piecewise constant property of the cost function, make the design of a codebook of prediction filters a highly challenging, non-convex optimization problem.

The problem is further exacerbated by stability issues that arise due to the coder's prediction loop. Specifically, note that the optimal set of prediction filters depends on the reconstructed signal from which predictions are made. But the reconstructed signal itself depends on the prediction filters in use. Clearly, we have a "chicken and egg" problem here, and this complex interplay between predictors and reconstructions makes codebook design a challenging problem. The dependency between predictors and reconstructions calls for an iterative design technique, wherein optimal predictors are designed for the given reconstruction statistics, and then the reconstructions are updated with the designed predictors. In the standard closed-loop technique (see for e.g., [5] for

Manuscript received January 31, 2021; revised June 13, 2021 and August 25, 2021; accepted November 16, 2021. Date of publication December 16, 2021; date of current version December 22, 2021. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Marc Antonini. *(Corresponding author: Bharath Vishwanath.)*

Bharath Vishwanath was with the Department of Electrical and Computer Engineering, University of California at Santa Barbara, Santa Barbara, CA 93106 USA. He is now with the Bytedance, San Diego, CA 92122 USA (e-mail: bharathv@ece.ucsb.edu).

Tejaswi Nanjundaswamy was with the Department of Electrical and Computer Engineering, University of California at Santa Barbara, Santa Barbara, CA 93106 USA. He is now with Apple Inc., Cupertino, CA 95014 USA (e-mail: tejaswi@ece.ucsb.edu).

Kenneth Rose is with the Department of Electrical and Computer Engineering, University of California at Santa Barbara, Santa Barbara, CA 93106 USA (e-mail: rose@ece.ucsb.edu).

Digital Object Identifier 10.1109/TIP.2021.3134454

quantizer design and [6] for a stochastic gradient version), the predictors designed in a given design iteration, i.e., given a training set of reconstructed blocks, are then plugged into the encoder and applied to a newly reconstructed signal in the next iteration, which will likely exhibit different statistics. This statistical mismatch can (and often does) grow as the encoder proceeds down the sequence, due to propagation through the prediction loop, causing severe design instability. As an effective remedy, the asymptotic closed-loop (ACL) design paradigm was proposed in [7]. ACL operates in an open-loop fashion by predicting from the (now fixed) reconstructed samples in the previous iteration. Thus, the predictors are applied to the same reconstruction statistics they were designed for, thereby eliminating statistical mismatch and ensuring better reconstructions over iterations. Nevertheless, as will be explained in Section II, on convergence, the design effectively operates in closed-loop fashion, and optimizes the predictors for closed-loop operation.

ACL provides a stable design platform. However, the design is still plagued by many poor local minima of the cost function. To address this, we propose a deterministic annealing (DA) approach to design prediction modes. DA [8] is a powerful non-convex optimization tool, inspired by principles of statistical physics and information theory. The probabilistic nature of DA yields an effective cost function via expectation, which is differentiable with respect to the prediction modes. At high temperature (maximum randomness), at the early stage of the algorithm, all the prediction modes are shown to coincide (at convergence all modes are identical), regardless of initialization, and they will only separate (through a sequence of phase transitions in the physical analogy) as the temperature is lowered. In other words, DA is independent of the initialization. Its annealing schedule gradually reduces the randomness of the solution so as to avoid poor local minima. The overall method proposed herein embeds ACL within the DA framework. The benefits of DA are complemented by the stable design platform of ACL, effectively addressing the central design challenges enumerated above.

The design of prediction modes has many practical applications involving adaptive prediction. In this paper, we consider an important application, namely, predictor design in video coding. Modern video coders exploit temporal correlations by employing motion compensated prediction [3], [9]. Simple pixel copying of the best (motion-compensated and possibly interpolated) block from the reference frame is used to obtain the prediction signal. The resulting prediction error is then decorrelated by a transform, typically the discrete cosine transform (DCT), and the transform coefficients are quantized and sent to the decoder. Such pixel-to-pixel temporal prediction is suboptimal in that it ignores significant spatial correlations in the video signal. Several approaches that account for spatial correlations include multi-tap filtering [10], [11] and three-dimensional subband coding [12], [13], which incur high encoder complexity. An earlier work from our lab [14] proposed an effective way to account for complex spatio-temporal correlations by first applying the transform to spatially decorrelate a block, and subsequently performing temporal prediction of the resulting uncorrelated transform coefficients. The temporal evolution of each transform coefficient in a block, along its motion trajectory, is modelled as a first order auto-regressive process. Thus we have a set of uncorrelated temporal processes, each representing the temporal evolution of a given coefficient (or "frequency") in the block. Moreover, transform domain temporal prediction (TDTP) perfectly captures and exploits the variations in temporal correlations across frequencies, which are otherwise masked in the pixel domain.

Modern video coders employ sub-pixel motion compensation for improved prediction, by interpolating the reference blocks to fractional pixel accuracy. Interpolation filters also use information from outside the block boundary, a fact that must be accounted for when optimizing prediction modes. Moreover, sub-pixel interpolation filters, when considered in the transform domain, interfere with the operation of TDTP filters. Thus, to completely disentangle the effect of interpolation filters, first handling interpolation within the block [15], and later to account for boundary information through the construct of extended block TDTP (EB-TDTP) [16] were proposed. With EB-TDTP, an extended reference block is first spatially decorrelated via DCT. Temporal prediction filters are then applied for the extended transform blocks. This is followed by inverse-DCT and interpolation to obtain the prediction signal. The optimal EB-TDTP filters were shown to be least square estimates [16], which enhances the performance beyond that offered by the standard correlation coefficient formulation.

Video signals exhibit significant variations in local statistics. This requires the coder to adapt to local statistics, and an effective approach involves a set of trained prediction modes for the encoder to choose from. The EB-TDTP filter is a high-dimensional vector and the problem at hand effectively corresponds to vector quantizer design, a notorious non-convex optimization problem. Here too, standard closed-loop design suffers from significant instability issues. We thus propose a DA-ACL framework to learn prediction filters to address these challenges. Note that an initial framework for predictor design for simple 1D predictive coders, and design of "plain" TDTP filters restricted to fixed block size, appeared in our preliminary conference publications [17] and [18], respectively. This paper subsumes the early work and complements it with the following contributions: i) It extends the DA-ACL design to account for EB-TDTP filters ii) It allows for variable block size, an important feature in HEVC that was disabled in our previous works and iii) Extended derivation of DA-ACL in conjunction with least-squares estimation which opens the door to general applicability of the method to a broader class of problems.

The rest of the paper is organized as follows. In Section II, we formulate the problem for a simple predictive compression system along with some relevant background. Section III introduces the DA-ACL paradigm for predictor design. Applications to video coding with EB-TDTP filter design is discussed in Section IV. Experimental results are provided in Section V followed by conclusions in Section VI.

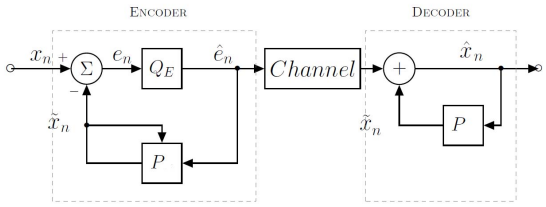


Fig. 1. Predictive compression system.

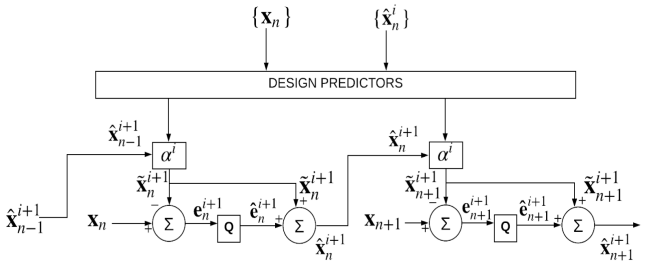


Fig. 2. Closed-loop design.

II. Background

A. Linear Prediction

Fig. 1 shows a predictive compression system. Let x_n, $0 \leq n \leq N$ be the input samples. The signal is modelled as a first-order auto-regressive process. The current sample x_n is predicted from the previous reconstructed sample $\hat{x}_{n-1}$ as,

$$\tilde{x}_n = \alpha \hat{x}_{n-1} \tag{1}$$

The resulting prediction error, $x_n - \tilde{x}_n$, is quantized and sent to the decoder. The predictor is designed to minimize the sum of squared prediction errors given by

$$E = \sum_{n=1}^{N} (x_n - \alpha \hat{x}_{n-1})^2 \tag{2}$$

The optimal predictor, obtained by basic linear estimation derivation, is

$$\alpha = \frac{\sum_n x_n \hat{x}_{n-1}}{\sum_n \hat{x}_{n-1}^2} \tag{3}$$

In order to adapt the predictor to variations in signal statistics, let the input be partitioned into groups or blocks of samples $\{g\}$. Let N_g be the set of samples belonging to a particular block g. The encoder is given a choice of K prediction filters $\{\alpha_k, k = 1, 2, \ldots, K\}$. The encoder chooses the best prediction mode for each block of samples. Let the best prediction mode for a given block g be $\hat{\alpha}_g \in \{\alpha_k\}$. The problem at hand is to design the prediction filters $\{\alpha_k\}$ such that the overall sum of squared prediction error

$$E' = \sum_g \sum_{n \in N_g} (x_n - \hat{\alpha}_g \hat{x}_{n-1})^2, \tag{4}$$

is minimized.

The piecewise constant nature of the cost function, with respect to the encoder's mode decisions, renders standard convex optimization algorithms inapplicable to the current scenario. A common, suboptimal remedy is the "*K-modes*" predictor design which we discuss next.

B. Iterative K-Mode Predictor Design

Let us assume for the moment that we have a set of reconstructed samples $\hat{x}_n$ at the encoder. Given these reconstructions, we can design prediction modes in a way similar to "K-means" clustering. With an initialization of the prediction modes, the following steps are performed iteratively:

- Mode assignment: For a given block g, assign the best mode from the set of prediction modes which minimizes the squared prediction error for the block.
- Prediction modes update: Let N_k be the union of samples from blocks that share the same prediction mode. Similar to (3), the optimal prediction mode α_k for this cluster is given by,

$$\alpha_k = \frac{\sum_{n \in N_k} x_n \hat{x}_{n-1}}{\sum_n \hat{x}_{n-1}^2} \tag{5}$$

Such *"K-modes"* predictor design optimizes the predictors for a given fixed set of reconstructions. However, in practice, these reconstructions will themselves depend on the predictors in use. This necessitates a two-fold optimization strategy, wherein, reconstructions and predictors are optimized iteratively. Given an updated set of prediction modes, there are several optional ways to update the reconstructions, leading to the following design paradigms.

C. Open-Loop, Closed-Loop, and Asymptotic Closed-Loop Design

Various techniques have been proposed in the context of joint design of predictors and quantizers. Since in most of modern video codecs the quantizer is fixed (up to scaling), our focus here is on predictor design given fixed quantizers, while noting that the same principles are also applicable to other predictive coder modules such as quantizers. In open-loop predictor design (see e.g., [5]), the predictor is designed using original samples, which do not depend on the predictors and the design is inherently stable. However, since the predictor must ultimately be applied to reconstructed samples, to avoid decoder drift, it will in fact operate on statistics mismatched with the design phase. In closed-loop design, predictors are designed iteratively. Let $\hat{\alpha}_g^i$ be the predictor for block g in iteration i. The reconstructed samples for the corresponding block in iteration $i+1$ is updated as,

$$\hat{x}_n^{i+1} = \hat{\alpha}_g^i \hat{x}_{n-1}^{i+1} + \hat{e}_n^{i+1} \tag{6}$$

where $\hat{e}_n^{i+1}$ is the quantized prediction error $e_n = x_n - \hat{\alpha}_g^i \hat{x}_{n-1}^{i+1}$. Predictor $\hat{\alpha}_g^i$ was designed for reconstruction in iteration i: $\{\hat{x}_n^i\}$. However, it is applied to the reconstructed samples of iteration $i+1$: $\{\hat{x}_n^{i+1}\}$. This mismatch results in design instability, which is exacerbated due to feedback through the prediction loop, and often proves catastrophic at low rates. To tackle this issue, ACL was proposed in [7]. ACL enjoys

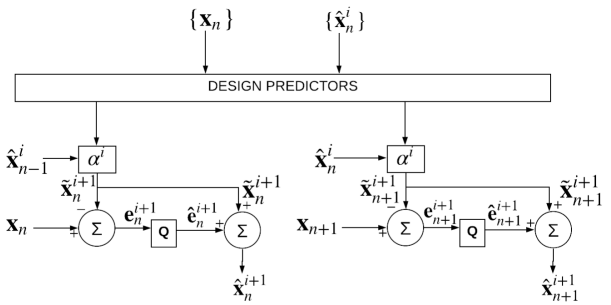


Fig. 3. Asymptotic closed-loop design.

the best of both worlds. At each iteration, the samples are predicted and reconstructed in open loop fashion as,

$$\hat{x}_n^{i+1} = \hat{\alpha}_g^i \hat{x}_{n-1}^i + \hat{e}_n^{i+1} \tag{7}$$

where $\hat{e}_n^{i+1}$ is the quantized prediction error $e_n^{i+1} = x_n - \hat{\alpha}_g^i \hat{x}_{n-1}^i$. The predictor $\hat{\alpha}_g^i$ is used with reconstructed samples $\hat{x}_n^i$, the same set of samples that it was designed for, thereby eliminating statistical mismatch and the resulting design instability. The new set of reconstructed samples are then used to design prediction modes α_k^{i+1}. Upon convergence, the reconstructed samples remain the same over iterations. Thus, predicting from $\hat{x}_n^i$ is same as predicting from $\hat{x}_n^{i+1}$, which is essentially closed-loop operation. The predictors designed are thus optimal for closed-loop operation. Fig. 2 depicts closed-loop design and Fig. 3 illustrates ACL design. Note that the prediction loop of CL is open in ACL which disallows propagation through the loop and hence avoids change in statistics.

With this background, we next introduce the proposed paradigm for predictor modes design.

III. Deterministic Annealing-Based Predictor Design

The hard prediction mode assignment to every signal block makes it difficult to optimize the system with respect to the prediction modes, as the derivatives with respect to mode decisions vanish almost everywhere. Hence an iterative K-mode design, a variant of "K-means" clustering was proposed in [19]. However, this only ensures convergence to a local minimum and renders the system highly sensitive to initialization. A related problem is encountered in quantizer design, where the piecewise constant nature of the quantizer makes it a challenging optimization problem. In order to jointly overcome the fundamental challenges of non-convexity and design instability, we propose to embed the ACL based minimization of the overall prediction error within the DA framework. The proposed approach is inspired by, and builds on the deterministic annealing (DA) framework of [8]. DA is motivated by the intuition gained from annealing process in physical chemistry, where certain systems are driven to their low energy states by gradual cooling of the system. Analogously, we introduce controlled randomness in the prediction mode assignment for the blocks, but deterministically minimize the overall prediction error, thereby avoiding many poor local minima. The inherent probabilistic nature of DA allows us to deterministically optimize the effective cost function, an appropriate expectation function that efficiently accounts for and replaces the stochastic wandering on the cost surface of the classical method of simulated annealing [20]. The amount of randomness is measured by the Shannon entropy and is essentially controlled by the "temperature" of the system. The prediction mode assignment is no longer piecewise constant, and is differentiable everywhere, thus paving the way to effective optimization of prediction modes.

A. Prediction Mode Derivation

We consider a random setting wherein for each block, a prediction mode is chosen *in probability*. Thus, the mean squared prediction error to minimize in ACL iteration i is taken as the expectation,

$$J = \sum_g \sum_k \sum_{n \in N_g} P_g P_{k|g}^i (x_n - \alpha_k^i \hat{x}_{n-1}^i)^2 \tag{8}$$

where P_g is the probability assigned to input data block g which is assumed to be uniform over all signal blocks. Association probability $P_{k|g}^i$ is the probability that prediction mode α_k is selected for input block g. The degree of randomness in the system is naturally measured by the Shannon entropy:

$$H = -\sum_g \sum_k P_{gk}^i \log(P_{gk}^i), \tag{9}$$

where $P_{gk}^i = P_g P_{k|g}^i$ is the joint probability distribution over prediction modes and training data blocks. The optimization problem is naturally restated as the minimization of the Lagrangian cost function, directly analogous to the Helmholtz free energy of statistical physics:

$$\mathcal{L} = J - TH, \tag{10}$$

where Lagrange parameter T controls the randomness of the solution. As an aside, there is an alternative (equivalent) way of formulating the problem, which is to maximize the Shannon entropy under a constraint of a given level of expected distortion, i.e, to maximize the Lagrangian given by,

$$\mathcal{L}' = H - \beta J \tag{11}$$

The motivation to maximize entropy stems from Jaynes's celebrated maximum entropy principle [21] which states that of all the probability distributions that satisfy a given set of constraints, it is beneficial to choose the one that maximizes the entropy, thereby avoiding the implicit imposition of any restrictive assumptions. It is obvious that the solution that minimizes the Lagrangian in (10) also maximizes the Lagrangian in (11).

Returning to the formulation of (10), note that the degree of randomness is controlled by Lagrangian parameter T, which corresponds to temperature in the physical analogy. As we lower T, we trade entropy for prediction error. At the limit of zero randomness, we in fact directly minimize the overall prediction error.

A notable benefit of randomization is that the expected distortion is now differentiable with respect to the mode decisions (now association probabilities rather than binary decisions). Minimizing the Lagrangian cost with respect to the association probabilities $P^i_{k|g}$, while additionally imposing the obvious constraint $\sum_k P^i_{k|g} = 1$ (legitimate probabilities), yields the Gibbs distribution:

$$P^i_{k|g} = \frac{e^{\frac{-\sum_{n\in N_g}(x_n-\alpha^i_k\hat{x}^i_{n-1})^2}{T}}}{\sum_j e^{\frac{-\sum_{n\in N_g}(x_n-\alpha^i_j\hat{x}^i_{n-1})^2}{T}}} \tag{12}$$

Note that at high temperatures, we in fact maximize the system entropy and the association probabilities are indeed uniform.

The optimal prediction modes satisfy,

$$\frac{\partial J}{\partial \alpha^i_k} = \sum_g \sum_{n\in N_g} 2P_g P^i_{k|g}(x_n - \alpha^i_k\hat{x}^i_{n-1})(-\hat{x}^i_{n-1}) = 0 \tag{13}$$

Thus, the optimal prediction modes are given by,

$$\alpha^i_k = \frac{\sum_g \sum_{n\in N_g} P^i_{k|g} x_n \hat{x}^i_{n-1}}{\sum_g \sum_{n\in N_g} P^i_{k|g}(\hat{x}^i_{n-1})^2} \tag{14}$$

At this point, it is instructive to pause and compare the solution from DA (14) to the prediction modes in standard K-modes design (5). As we see in (5), the cross correlations and auto-correlations are taken as expectations over samples classified to a particular mode. However, in (14), the expectations are taken, with respect to the association probabilities, over the entire training set. Thus, in standard K-modes design in (5), the samples have highly localized influence, as they only impact the "nearest mode", thus blinding the system to possible better solutions further away. In other words, it is easy to get trapped in poor local optima. In contrast, in a DA based solution, each sample influences all the prediction modes through their association probabilities and the degree of influence varies with temperature. Specifically, at high temperature, all the association probabilities are uniform. Thus, the optimal prediction modes converge to and coincide at the correlation coefficient of the entire training set, the globally optimal single prediction mode. As the temperature is lowered, the degree of influence decreases and at the limit of zero temperature, the design is similar to the standard K-modes design. From this viewpoint, the standard K-modes design is a hard, zero-temperature design.

B. Overall Design

The design starts with a closed-loop initialization of the reconstructions and at a high temperature T_0. As observed earlier, at high temperatures, given the uniform association probabilities, all the prediction modes coincide at the optimal single prediction mode of (3), *regardless of initialization*. Thus, DA is effectively independent of initialization. As the temperature is lowered, the association probabilities become more "discriminating" and the solution less random. As the system is cooled it reaches certain temperatures called "critical temperatures", where the existing solution with its set of prediction modes is no longer stable. Thus, with slight perturbations, the number of distinct modes increases as new prediction modes emerge through cluster splits. This phenomenon corresponds to "phase transitions" in the physical analogy.

Algorithm 1 Proposed DA-ACL Predictor Design

```
initialize: closed-loop reconstructions, T=T_0;
while T < T_min do
    while ACL_iter < max_ACL_iter do
        (a) Predictor design:
        do
            (i) Optimize association probabilities;
            (ii) Optimize prediction modes;
        while association probabilities converge
        (b) ACL update of reconstructions;
        break on convergence;
    end
    Cool system: T = bT
end
```

At each temperature, the design iterates between predictor design and reconstruction update. For a given set of reconstruction statistics, the predictor design iterates between:

- a) Computing association probabilities for the prediction modes (12)
- b) Updating prediction modes (14)

These monotonically non-increasing steps minimize the Lagrangian $\mathcal{L}$. Upon convergence, the reconstructed samples $\hat{x}^{i+1}_n$ in a block g are updated in ACL fashion as,

$$\hat{x}^{i+1}_n = \sum_k P^i_{k|g}(\alpha^i_k\hat{x}^i_{n-1} + \hat{e}^{i+1}_{n,k}) \tag{15}$$

where, $\hat{e}^{i+1}_{n,k}$ is the quantized prediction error. Open-loop update ensures better reconstructions. Thus, ACL iterations are also monotonically non-increasing, ensuring the convergence of reconstructions. Upon convergence in reconstructions, the system is cooled and the process is repeated. Once the cooling is complete, the system gives prediction modes that minimize the overall prediction error (4) and that are optimal for closed-loop operation. The overall design procedure is summarized in Algorithm 1.

Having introduced a general framework for predictor design, we next consider an important application in the context of video coding.

IV. Predictor Design in Video Coding

Motion compensated prediction is a central component in modern video coders, tasked with removing temporal redundancies, which is critical to the overall compression efficiency of the coder. The best matching block from the reference frame is used as the prediction for the current block. Simple pixel copying, however, largely ignores the spatial correlations between pixels, and renders the prediction suboptimal. Moreover, pixel copying implicitly assumes that the temporal correlation coefficient is one at all frequencies. The invalidity of this implicit assumption is illustrated by the temporal

TABLE I
TEMPORAL CORRELATION COEFFICIENTS, ALONG MOTION TRAJECTORY, OF DCT COEFFICIENTS IN THE BLOCK

0.99	0.96	0.92	0.91	0.89	0.84	0.79	0.67
0.97	0.95	0.91	0.87	0.83	0.78	0.73	0.58
0.96	0.93	0.88	0.86	0.84	0.75	0.69	0.6
0.93	0.88	0.88	0.84	0.79	0.72	0.64	0.58
0.89	0.90	0.90	0.84	0.75	0.66	0.62	0.46
0.83	0.89	0.84	0.83	0.70	0.58	0.54	0.44
0.83	0.81	0.82	0.74	0.62	0.53	0.49	0.4
0.77	0.71	0.62	0.66	0.58	0.45	0.39	0.38

correlation coefficients evaluated for various DCT coefficients in Table I, over a sample sequence. Note how the correlation varies with frequency. Thus, to completely disentangle spatial and temporal correlations and to exploit the true frequency dependent nature of temporal correlations, transform domain temporal prediction (TDTP) was proposed in [14] which we briefly discuss next.

A. Transform Domain Temporal Prediction

TDTP models the temporal evolution of each transform coefficient as a first order AR process. In other words, we have parallel, uncorrelated AR processes, one per frequency (transform coefficient). Let x_n be a particular transform (say, DCT) coefficient in a given block in frame n, along a motion trajectory. The evolution of x_n is thus modeled as,

$$x_n = \alpha \hat{x}_{n-1} + e_n \tag{16}$$

where $\hat{x}_{n-1}$ is the corresponding DCT coefficient of the block in reconstructed frame $n-1$, along the motion trajectory, and e_n is the innovation sequence. The optimal prediction coefficient that minimizes the mean square prediction error is given by (3). By performing temporal prediction in the transform domain, TDTP effectively achieves both temporal and spatial decorrelation. Further, TDTP captures the variation of temporal correlation with spatial frequency, by optimizing the predictor for each transform coefficient.

We observe that if one were to use the entries of Table I as predictor coefficients, the effect would be coincidentally similar to that of a low-pass filter. Video coders use sub-pixel motion compensation which employs low-pass filters for interpolation. These interpolation filters interfere with TDTP, and may compromise its performance. Thus, to completely disentangle the effects of interpolation filters and TDTP filters, extended-block transform domain temporal prediction (EB-TDTP) was proposed in [16] which we briefly discuss next,

B. Transform Domain Temporal Prediction With Extended Blocks

Video coders employ sub-pixel motion compensation in which interpolated reference blocks are used as prediction signals. To obtain the interpolated signal, the coder makes use of boundary samples outside the reference block. Thus, applying TDTP on the interpolated signal is effectively considering spatial and temporal decorrelations in the subspace of the interpolated signal, in contrast with decorrelating in the actual space of the boundary-extended reference block. Moreover, as observed earlier, interpolation filters interfere with TDTP filters. EB-TDTP effectively addresses these challenges by first scaling the extended block pixels according to the temporal prediction coefficients in the transform domain, and then applying the interpolation filters. To formulate mathematically, let $Y_{n,b}$ be the block b of dimensions $B_1 \times B_1$ in frame n, which is to be predicted. Let $\hat{Y}^{mv}_{n-1,b}$ be the reference block in frame $n-1$, of dimensions $B_2 \times B_2 (B_2 > B_1)$, to which the video coder applies interpolation to obtain the prediction signal. Let the vertical and horizontal interpolation filters be denoted as I_v and I_h, which are matrices of sizes $B_1 \times B_2$ and $B_2 \times B_1$, respectively. Thus the interpolated prediction signal of standard coders is,

$$\tilde{Y} = I_v \hat{Y}^{mv}_{n-1,b} I_h \tag{17}$$

and the prediction signal from "plain" (i.e., without "extended block") TDTP is,

$$\tilde{Y}_{TDTP} = D'_{B_1}[\{D_{B_1}(I_v \hat{Y}^{mv}_{n-1,b} I_h) D'_{B_1}\} \circ F_{B_1}] D_{B_1} \tag{18}$$

where D_{B1} is the DCT matrix, F_{B1} is the TDTP filter and $\circ$ denotes component-wise matrix multiplication.

In EB-TDTP, we first spatially decorrelate the extended block $\hat{Y}^{mv}_{n-1,b}$ by a separable DCT. Then, the extended block TDTP filter F_{B_2} is applied to the transformed extended block. This is followed by inverse transform and interpolation to derive the prediction signal. Thus, EB-TDTP of $Y_{n,b}$, as illustrated in Fig. 4 can be formulated as,

$$\tilde{Y}_{EB-TDTP} = I_v D'_{B_2}\{\{D_{B_2} \hat{Y}^{mv}_{n-1,b} D'_{B_2}\} \circ F_{B_2}\} D_{B_2} I_h \tag{19}$$

To derive the predictor F_{B_2}, let $K_1 = I_v D'_{B_2}$, $K_2 = D_{B_2} I_h$, and $\hat{X}^{mv}_{n-1,b} = D_{B_2} \hat{Y}^{mv}_{n-1,b} D'_{B_2}$. The prediction error can thus be written as,

$$\begin{aligned} E_{EB-TDTP} &= \sum_n \sum_b \left\| Y_{n,b} - \tilde{Y}_{n,b} \right\|^2 \\ &= \sum_n \sum_b \left\| Y_{n,b} - K_1(\hat{X}^{mv}_{n-1,b} \circ F_{B_2}) K_2 \right\|^2 \\ &= \sum_n \sum_b [\sum_{r=1}^{B_1} \sum_{s=1}^{B_1} \{Y_{n,b}(r,s) - \sum_{i=1}^{B_2} \sum_{j=1}^{B_2} F_{B_2}(i,j) \\ &\quad \times \hat{X}^{mv}_{n-1,b}(i,j) K_1(r,i) K_2(j,s)\}^2] \end{aligned} \tag{20}$$

This is essentially a least-squares estimation problem of minimizing,

$$E_{EB-TDTP} = \sum_n \sum_b \left\| A_{n,b} \boldsymbol{f}_{B_2} - \boldsymbol{t}_{n,b} \right\|^2, \tag{21}$$

where $\boldsymbol{f}_{B_2}$ is a vector representation (containing all elements) of matrix F_{B_2}. $A_{n,b}$ and $\boldsymbol{t}_{n,b}$ are derived as,

$$A_{n,b}(u,v) = \hat{X}^{mv}_{n-1,b}(i,j) K_1(r,i) K_2(j,s) \tag{22}$$

$$\boldsymbol{t}_{n,b}(u) = Y_{n,b}(r,s) \tag{23}$$

where, $u = rB_1 + s$ and $v = iB_2 + j$. The optimal predictor is given by,

$$\boldsymbol{f}_{B_2} = (\sum_n \sum_b A^T_{n,b} A_{n,b})^{-1} (\sum_n \sum_b A^T_{n,b} \boldsymbol{t}_{n,b}) \tag{24}$$

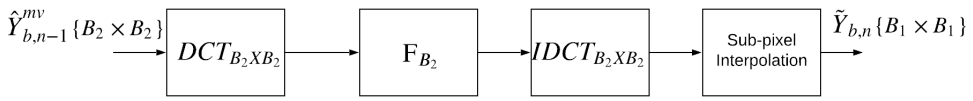


Fig. 4. Illustration of extended block transform domain temporal prediction.

As seen from (24), the optimal predictor computation is essentially posed as a classic least-squares problem. The discussion so far involved a single predictor. To realize the full potential of EB-TDTP, we need a set of EB-TDTP filters to achieve adaptivity, which implies the design of an efficient set of prediction modes. We note that multiple prediction modes introduce optimization in a higher dimensional parameter space, making the design more prone to be trapped in local poor minima. Thus, there is strong motivation to pursue a DA based solution.

V. Deterministic Annealing-Based EB-TDTP Mode Design

Video signals vary significantly in their statistics resulting in wide variations in the temporal correlations of the transform coefficients. This motivates for an adaptive EB-TDTP framework to realize its full potential. The design of EB-TDTP modes involves a large set of parameters to be optimized, causing severe statistical mismatch in closed-loop design, and often leading to catastrophic instabilities. Moreover, ACL based design was observed to be very sensitive to initialization, reflective of the severe non-convexity of the problem. To address these challenges, we propose a DA-ACL based solution to the problem of EB-TDTP modes design.

A. Problem Formulation

Let us consider an input training set which is partitioned into segments, each of which is called a group of pictures (GOP). Let the set of frames in a GOP be denoted by N_g. To get a crisp understanding of the design, we first introduce the design in fixed block setting and then extend it to variable block size setting. At the GOP level, the encoder switches between EB-TDTP modes $\{F_k\}$, where each F_k is specified by a matrix of prediction coefficients, of size $B_2 \times B_2$. (Note that the mode subscript B_2 has now been discarded to simplify notation). Let the prediction mode chosen for a GOP g be $\hat{F}_g$. The cost function, which is the overall prediction error to be minimized, is given by

$$E'_{EB-TDTP} = \sum_g \sum_{n\in g} \sum_b \left\| Y_{g,n,b} - \tilde{Y}_{g,n,b} \right\|^2 = \sum_g \sum_{n\in g} \sum_b \left\| Y_{g,n,b} - K_1(\hat{X}^{mv}_{n-1,b} \circ \hat{F}_g)K_2 \right\|^2$$

The cost function can be equivalently written as

$$E'_{EB-TDTP} = \sum_g \sum_{n\in g} \sum_b \left\| A_{g,n,b}\hat{\boldsymbol{f}}_g - \boldsymbol{t}_{n,b} \right\|^2 \quad (25)$$

where $\hat{\boldsymbol{f}}_g$ is the vector representation of matrix F_k and the definitions of $A_{g,n,b}$ and $\boldsymbol{t}_{g,n,b}$ are straightforward extensions of (22) and (23).

B. EB-TDTP Mode Derivation

The design involves randomization of the prediction mode assignment to GOPs. At a given temperature T and ACL iteration i, let conditional probability $P^i_{k|g}$ denote the probability of assigning EB-TDTP mode F^i_k to GOP g. The prediction error to be minimized is given by the expectation:

$$J_{EB-TDTP} = \sum_g \sum_k \sum_{n\in g} \sum_b P_g P^i_{k|g} \left\| A^i_{g,n,b}\boldsymbol{f}^i_k - \boldsymbol{t}_{n,b} \right\|^2 \quad (26)$$

where P_g denotes the probability of the input GOPs (assumed uniform). The degree of randomness is measured by the Shannon entropy,

$$H_{EB-TDTP} = -\sum_g \sum_k P^i_{gk} \log(P^i_{gk}), \quad (27)$$

where $P^i_{gk} = P_g P^i_{k|g}$ is the joint distribution over EB-TDTP modes and input GOPs. The cost function to be minimized is the Lagrangian,

$$\mathcal{L}_{EB-TDTP} = J_{EB-TDTP} - T\, H_{EB-TDTP} \quad (28)$$

The problem at hand is posed as the minimization of the entropy-constrained Lagrangian $\mathcal{L}_{EB-TDTP}$. The association probabilities that minimize the Lagrangian cost subject to the standard normalization constraint (adding up to 1), are given by:

$$P^i_{k|g} = \frac{e^{-\frac{\sum_{n\in g}\sum_b \left\| A^i_{g,n,b}f^i_k - t_{n,b} \right\|^2}{T}}}{\sum_j e^{-\frac{\sum_{n\in g}\sum_b \left\| A^i_{g,n,b}f^i_j - t_{n,b} \right\|^2}{T}}} \quad (29)$$

Minimizing the expected distortion with respect to the prediction modes yields,

$$\frac{\partial J}{\partial \boldsymbol{f}^i_k} = \sum_g \sum_{n\in g} \sum_b P^i_{k|g}((A^i_{g,n,b})^T A^i_{g,n,b}\boldsymbol{f}_k - (A^i_{g,n,b})^T \boldsymbol{t}_{n,b}) = \boldsymbol{0} \quad (30)$$

Thus, the optimal prediction modes are given by,

$$\boldsymbol{f}^i_k = \{\sum_g \sum_{n\in g} \sum_b P^i_{k|g}((A^i_{g,n,b})^T A^i_{g,n,b})\}^{-1} \times \{\sum_g \sum_{n\in g} \sum_b P^i_{k|g}((A^i_{g,n,b})^T \boldsymbol{t}_{n,b})\} \quad (31)$$

C. Mode Design Extension to Variable Block Size Coding

Variable block size coding is an important feature of state-of-the-art codecs, which enables better adaptation to local

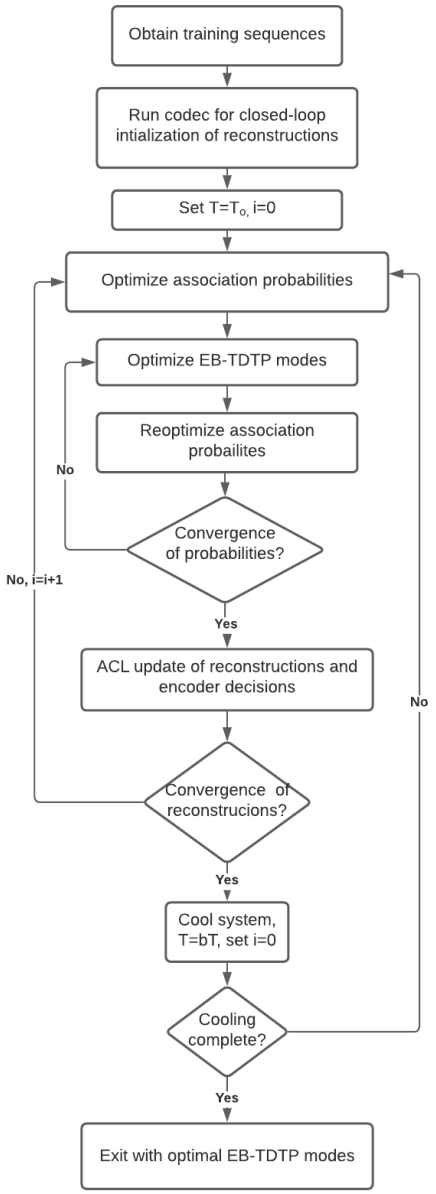


Fig. 5. Flow chart of the proposed DA-ACL EB-TDTP design algorithm.

signal statistics and provides considerable gains over fixed block size coding. So far, we covered prediction modes design for the simpler setting where the block size is fixed. This section extends the design to handle variable block sizes, which enables its incorporation within the latest codecs.

Let us index the available block sizes by s. To extend the framework to the variable block size setting, we define an EB-TDTP mode F_k as consisting of a set of EB-TDTP filters $\{F_{k,s}\}$, where $F_{k,s}$ is the EB-TDTP filter for block size s within mode F_k. Thus, a minimal side information specifying to the decoder the choice of EB-TDTP mode for a given GOP, completely specifies the prediction filters used for all block sizes. At design iteration i, using the convenient vector representation $\boldsymbol{f}^i_{k,s}$ for prediction filter $F_{k,s}$, the expected prediction error of (26) is now correspondingly expanded to a variable block setting (and denoted $EB-TDTP-VB$):

$$J_{EB-TDTP-VB} = \sum_g \sum_k \sum_{n\in g} \sum_s \sum_{b\in s} P_g \times P^i_{k|g} \left\| A^i_{g,n,b} \boldsymbol{f}^i_{k,s} - \boldsymbol{t}_{n,b} \right\|^2 \quad (32)$$

To obtain the Lagrangian we add the entropic constraint (27), and the free energy Lagrangian, $\mathcal{L} = J - TH$, becomes:

$$\mathcal{L}_{EB-TDTP-VB} = J_{EB-TDTP-VB} - T\, H_{EB-TDTP} \quad (33)$$

The association probabilities in this setting are given by

$$P^i_{k|g} = \frac{e^{-\frac{\sum_{n\in g}\sum_s \sum_{b\in s} \left\| A^i_{g,n,b} f^i_{k,s} - t_{n,b} \right\|^2}{T}}}{\sum_j e^{-\frac{\sum_{n\in g}\sum_b \left\| A^i_{g,n,b} f^i_{j,s} - t_{n,b} \right\|^2}{T}}} \quad (34)$$

and the optimal prediction filter for block size s in mode k is, in vector representation:

$$\boldsymbol{f}^i_{k,s} = \{\sum_g \sum_{n\in g} \sum_{b\in s} P^i_{k|g} ((A^i_{g,n,b})^T A^i_{g,n,b})\}^{-1} \times \{\sum_g \sum_{n\in g} \sum_{b\in s} P^i_{k|g} ((A^i_{g,n,b})^T \boldsymbol{t}_{n,b})\} \quad (35)$$

D. Overall Design

The overall design is illustrated in Fig. 5, where the design starts from a high temperature and is gradually cooled. At high temperatures, the association probabilities are uniform as is obvious from (34) and all the EB-TDTP modes given by (35) are coincidental. At a given temperature T, the design iterates between optimizing predictors and updating reconstructions in ACL way. Optimizing predictors for a given reconstruction set involves computing association probabilities as (34) and updating prediction modes according to (35). Upon convergence, the reconstructed samples in GOP g are updated in ACL fashion. With video codecs, we also need to account for various encoder decisions like motion vectors, block partitions etc., to ensure convergence. Thus, during reconstruction update, the encoder is allowed to update its decisions, ensuring optimal decisions for the new reconstructions. We use these decisions to generate prediction residual statistics for the next ACL iteration. Upon convergence in reconstructions, the system is cooled and the process is repeated. As the temperature is lowered, the system becomes more deterministic with the emergence of more EB-TDTP modes through a sequence of phase transitions. At the limit of zero temperature, the prediction modes directly minimize the squared distortion and with the convergence in the reconstructions, the prediction modes designed are optimal for the closed-loop operation. Note that our earlier work in [18] with 'plain' TDTP with fixed block size is a special case of the current approach with a single block size s and the extended block size $B_2 = B_1$. While the proposed solution was presented in conjunction with video coding, we note in passing that the method is generally applicable to a rich class of problems involving least-square estimation and non-stationary data (see, e.g, [22], [23]).

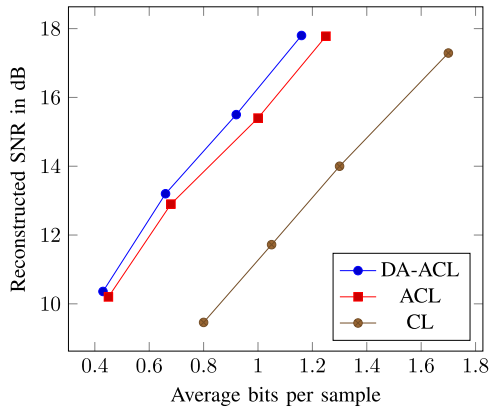


Fig. 6. First-order scalar predictive coding. Reconstructed SNR vs. average bits per sample for the test set of speech files.

VI. Experimental Results

A. A Simple First Order Predictive Encoder

The first experiment considers the simple setting of scalar, first order predictive coding. We chose speech signals as a real-world source data. A set of six speech files from the EBU SQAM database were chosen for simulations [24]. Half of the speech files were used as the training set for designing prediction modes and the remaining half as the test set. A set of six prediction modes were designed. A fixed dead-zone quantizer was employed for quantization. Different R-D points were obtained by varying Lagrange multiplier of entropy constrained quantization. The 3 competitors were: closed-loop (CL), "plain ACL", and the proposed method (DA-ACL). While DA-ACL is independent of initialization, CL and ACL designs were repeated with multiple initializations and the best results were selected. Fig. 6 shows the reconstructed SNR versus bit rate. It is evident from the results that the proposed DA-ACL method gives significant 0.4dB and 5dB gains over competitors ACL and CL, respectively.

B. Video Coding Results

The proposed method was implemented in HM 14.0. We chose low-delay P, or LDP profile (section 9.2.3 in [25]) for our experiments. In all experiments, the encoder only uses the previous frame as reference. Choosing uni-directional prediction and disabling multiple reference frames gives a "clean" comparison of design methods without being muddled by complexities in bi-directional prediction or multiple reference frames. We emphasize, that the approach is nonetheless applicable to less restricted settings. The anchor is the HEVC codec which performs conventional pixel domain prediction, i.e, performs simple pixel copying from a possibly interpolated block in the reference frame. The competing codecs use EB-TDTP prediction modes.

As mentioned in Section V-C, the EB-TDTP mode is a collection of EB-TDTP filters covering all block sizes. The EB-TDTP filters are employed as depicted in Fig. 4. Specifying an EB-TDTP mode for a GOP, at minimal side information cost, completely specifies the prediction filters for all block sizes. To minimize the encoding complexity,

TABLE II
The Training Set of Video Sequences

Sequence
Tennis (1080p)
Pedestrian (1080p)
Parkjoy (720p)
Vidyo3 (720p)
BQMall (720p)
Racehorses (240p)
Paris (cif)
Waterfall (cif)
City (cif)
Stefan (cif)
Highway (cif)
Mobile (cif)

TABLE III
Performance Over the Test Set: Bit-Rate Savings Over HEVC (in %) for the Y Component

Sequence	CL	ACL	DA-ACL
Kimono (1080p)	7.2	8.5	11.3
Tractor (1080p)	12.8	14.4	15.7
Parkscene (1080p)	5.9	6.0	6.5
BasketballDrive (1080p)	7.5	7.6	10.3
KristenAndSara (720p)	8.0	8.1	8.3
vidyo4 (720p)	14.3	16.1	16.5
Ducks (720p)	12.4	13.6	14.2
Mobisode2 (480p)	3.8	3.8	4.2
BQTerrace (480p)	0.4	1.8	2.8
Keiba (480p)	6.1	6.6	7.5
BasketballPass (240p)	-1.2	-0.1	0.2
Mobisode2 (240p)	-1.8	0.4	1.1
Coastguard (cif)	7.4	7.6	11.2
Bridge-far (cif)	3.9	5.3	6.3
Soccer (cif)	7.9	8.7	9.7
Salesman (qcif)	-3.7	-0.2	0.9
Average	**5.7**	**6.6**	**7.9**

EB-TDTP filters are used only during motion compensated prediction but not during motion estimation. In other words, conventional motion search is performed in the pixel domain to determine the motion vector, and only then we perform transform domain prediction with EB-TDTP filters. The simulations are performed at QP values of 22, 27, 32 and 37, as recommended in the HEVC Common Test Conditions (CTC). The implementation details for training EB-TDTP modes are discussed next.

1) Training EB-TDTP Modes: EB-TDTP filters depend on reconstruction statistics which vary with QP value. Thus, EB-TDTP modes are trained conditioned on the QP value. For each QP value, we design four EB-TDTP modes by each of the following design methods:

i) Standard closed-loop design, denoted CL: predictors are optimized by the 'K-modes' clustering method and the reconstruction is updated in the standard closed-loop fashion. This is the traditional approach to predictor design and suffers from both convergence to poor local minima due to the greedy 'K-modes' style design of predictors, and design instability due to closed-loop update of the reconstructed signal.

ii) K-mode design with "plain ACL", denoted ACL: predictors are still optimized by the 'K-modes' clustering method, but the reconstruction is updated using ACL. This design

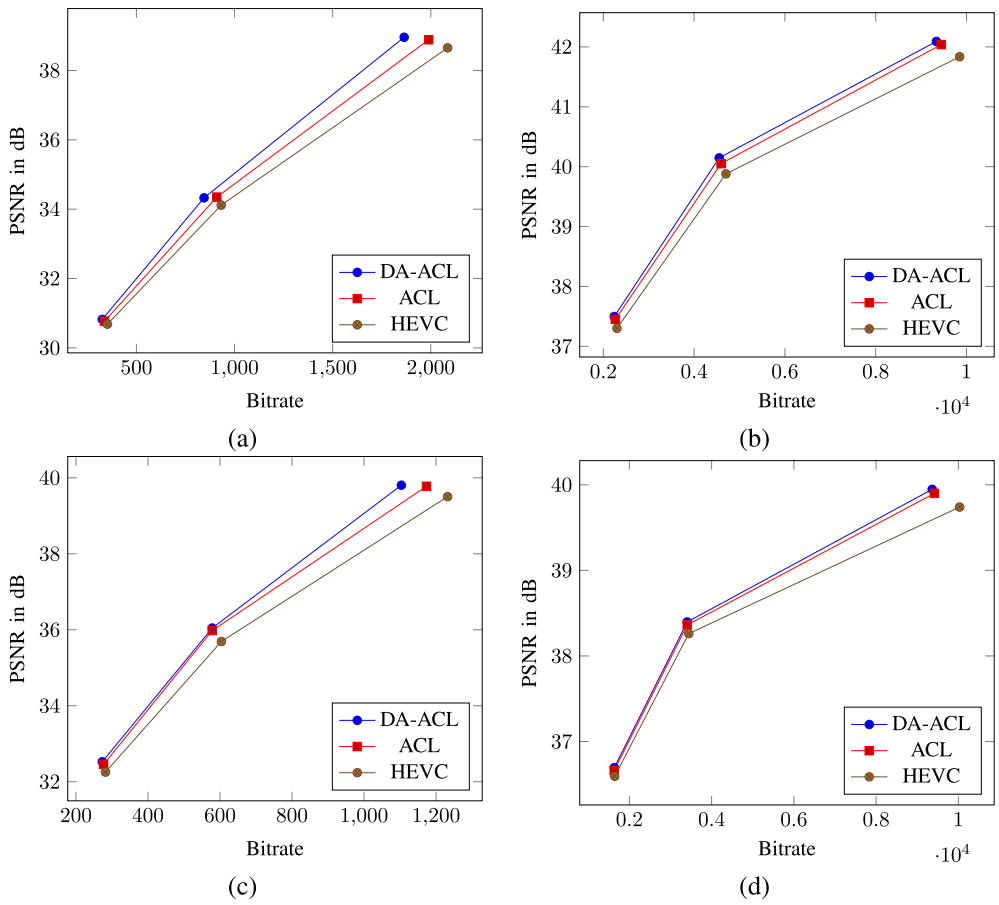


Fig. 7. RD curves for: (a) Coastguard, (b) Kimono, (c) Soccer, and (d) BasketballDrive sequences.

enjoys stability due to ACL but still suffers from poor local minima of the cost.

iii) The proposed method, denoted DA-ACL: predictors are optimized by DA and the reconstructions are updated in ACL fashion. This solves both the sensitivity to initialization and the design instability issues.

The aforementioned methods perform iterative optimization of predictors and reconstruction. Given a set of predictors, the reconstruction is updated by the HEVC codec. During reconstruction update, reconstruction statistics for different block sizes are collected for predictor optimization in the next design iteration. The predictor optimization is done in a separate module which is external to the codec. The training set sequences are listed in Table II. Note that, to avoid unintended bias, the data set was randomly partitioned into training and test sets.

2) Testing: The trained prediction modes are stored at both encoder and decoder. The encoder performs a brute-force search over all prediction modes for each GOP and selects the best mode. The average bit-rate reduction by using EB-TBTP modes over HEVC that performs conventional pixel domain prediction is calculated as per [26]. The bit-rate savings on the test set, due to the EB-TDTP modes designed by CL, ACL and DA-ACL, are tabulated in Table III. Rate-distortion (RD) curves for some example test sequences are shown in Fig. 7. It is important to note that the figures show only 3 RD points corresponding to QP values 27, 32 and 37. This is only to maintain a rate scale that allows for better visualization of the distinction between methods. The gains are in fact larger for the missing RD point (QP=22) that falls outside the figure. The significant bit-rate reduction over the test set provides clear evidence for the utility of proposed approach. Note that the gain increase with bit-rate may be explained by the fact that at lower rates, many transform coefficients are quantized to zero, thus reducing the scope for gains due to optimal predictors for these coefficients.

While the most practically relevant measure is the overall coding gains, which was therefore the central focus of the experiments, it is interesting to also measure how the improved prediction performed in terms of its own direct performance

TABLE IV

RESIDUAL ENERGY PER TRANSFORM COEFFICIENT WITH HEVC PREDICTION FOR *Coastguard* TEST SEQUENCE AT QP=32

591.6	397.0	226.8	146.0	105.7	57.9	28.1	9.8
608.2	363.7	228.7	126.1	93.7	65.6	27.3	9.4
629.8	367.6	243.1	159.9	99.2	55.5	31.7	10.1
666.5	408.0	239.1	142.9	91.5	56.3	26.3	9.8
561.6	392.9	200.7	147.4	81.7	47.1	20.2	6.8
402.5	295.8	151.0	94.9	52.9	29.1	10.9	4.0
211.1	120.0	59.4	33.3	18.7	9.5	4.1	1.7
81.4	31.4	14.0	8.5	5.2	2.9	1.1	0.8

TABLE V

ENERGY PER TRANSFORM COEFFICIENT WITH EB-TDTP PREDICTION FOR *Coastguard* TEST SEQUENCE AT QP=32

590.9	396.0	221.6	144.0	103.5	56.2	23.82	8.1
604.4	353.0	226.7	123.2	90.7	58.4	24.9	5.9
625.6	363.9	236.8	151.8	95.1	49.5	23.7	5.9
662.1	401.5	222.9	135.5	83.7	49.4	17.6	4.4
552.7	386.5	192.9	133.5	69.2	38.6	13.5	3.3
400.3	287.4	137.1	85.7	44.2	22.9	6.5	1.5
208.5	111.5	51.8	28.5	14.7	6.4	2.1	0.6
70.9	28.6	10.5	6.1	2.9	1.7	0.5	0.3

TABLE VI

PERCENTAGE REDUCTION IN VARIANCE OF TRANSFORM COEFFICIENTS FOR *Coastguard* TEST SEQUENCE AT QP=32

0.1	0.2	2.3	1.4	2.0	2.9	15.1	17.0
0.6	3.0	0.9	2.3	3.2	11.0	9.0	36.5
0.7	1.0	2.6	5.0	4.2	10.9	25.3	41.5
0.7	1.6	6.8	5.3	8.5	12.2	33.1	55.0
1.6	1.7	3.8	9.4	15.3	18.0	33.2	51.6
0.6	2.8	9.2	9.7	16.4	21.4	40.9	63.5
1.2	7.1	12.8	14.5	22.0	32.8	47.7	66.8
12.8	8.7	24.8	28.0	42.1	42.8	58.0	60.0

criterion, namely, in terms of the observed prediction error. (The mean squared prediction error is the energy of the residual prior to quantization in the codec). To illustrate this through a small experiment, we measured the prediction gain achieved on the *coastguard* sequence at QP=32, corresponding to the middle RD point in the Fig. 7(a), and observed a significant 0.84dB improvement over standard HEVC prediction. To further illustrate the direct impact of EB-TDTP prediction in minimizing residual energy in transform domain, we present the energy of the residual transform coefficients (for block size 8×8 in *coastguard* sequence at QP=32) with HEVC prediction and EB-TDTP prediction in Tables IV and V respectively. For ease of comparison, the percentage reduction in residual energy for each transform coefficient is presented in Table VI. As mentioned earlier, the proposed transform domain prediction captures the variations in temporal correlations across frequencies and provides enhanced prediction gains which grow with increase in frequency. For clear illustration, let us focus specifically on the prediction gains along the diagonal (i.e., for DCT coefficients at positions (i,i), $i = 0,\ldots,7$). The prediction gain, G_P, and its corresponding expression in dB, $\mathrm{SNR}_P[dB]$, are traditionally defined as [27]

$$G_P = \frac{\sigma^2}{\sigma_e^2}, \quad \mathrm{SNR}_P[dB] = 10\log G_P,$$

where σ^2 denotes the coefficient's variance, and σ_e^2 is its prediction error variance. The increase in dB of the prediction gain, denoted $\Delta\mathrm{SNR}_P$, provided by EB-TDTP over standard HEVC prediction, is depicted in Fig. 8 for coefficients (i,i) along the diagonal of 8×8 blocks in the *coastguard* sequence, at QP=32. Note how EB-TDTP offers enhanced prediction, wherein the additional prediction gains grow from 0 to 4 dB with increasing frequency.

As regards complexity, if one considers a direct implementation, then employing transform domain prediction with a particular EB-TDTP mode increases the encoder complexity by about a factor of two. Further, since the encoder does brute-force search over four modes, the overall encoding complexity is about 8x compared to the anchor. The decoder simply uses the mode specified by the encoder and thus incurs complexity increase of about 2x compared to the anchor decoder. It is clear that various approaches can be explored for fast selection of prediction modes and fast implementation of transform domain prediction. The current paper focuses on demonstrating the potential of the design method, and exploration of complexity reduction techniques is beyond its scope.

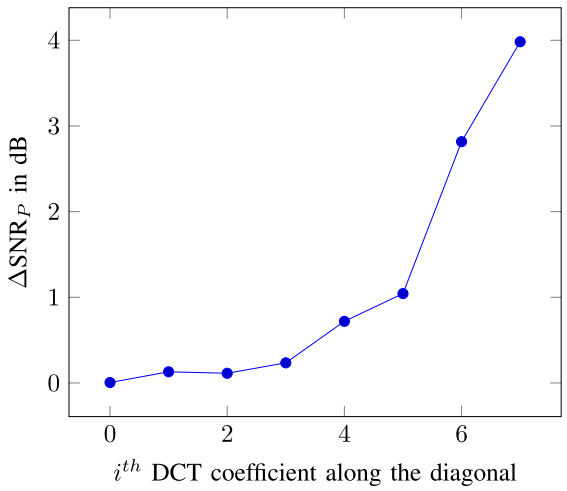


Fig. 8. Increase in prediction gain (in dB) achieved by EB-TDTP over standard HEVC prediction, measured over the diagonal (i,i) DCT coefficients for *coastguard* test sequence at QP=32.

VII. CONCLUSION

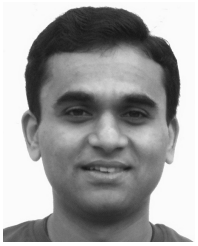
This paper presents a novel near-optimal procedure for designing prediction modes for adaptive compression systems. It effectively resolves significant shortcomings due to statistical mismatch and design instability of standard approaches. The deterministic annealing-based framework enables direct optimization of the overall cost with respect to prediction mode decisions, and avoids many poor local minima that trap its competitors. Substantial gains in the experiments demonstrate the efficacy of the proposed approach.

REFERENCES

[1] A. S. Spanias, "Speech coding: A tutorial review," *Proc. IEEE*, vol. 82, no. 10, pp. 1541–1582, Oct. 1994.
[2] T. Painter and A. Spanias, "Perceptual coding of digital audio," *Proc. IEEE*, vol. 88, no. 4, pp. 451–515, Apr. 2000.
[3] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.
[4] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *J. Roy. Stat. Soc. C, Appl. Statist.*, vol. 28, no. 1, pp. 100–108, 1979.
[5] V. Cuperman and A. Gersho, "Vector predictive coding of speech at 16 kbits/s," *IEEE Trans. Commun.*, vol. COM-33, no. 7, pp. 685–696, Jul. 1985.
[6] P. C. Chang and R. Gray, "Gradient algorithms for designing predictive vector quantizers," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-34, no. 4, pp. 679–690, Aug. 1986.

[7] H. Khalil, K. Rose, and S. L. Regunathan, "The asymptotic closed-loop approach to predictive vector quantizer design with application in video coding," *IEEE Trans. Image Process.*, vol. 10, no. 1, pp. 15–23, Jan. 2001.
[8] K. Rose, "Deterministic annealing for clustering, compression, classification, regression, and related optimization problems," *Proc. IEEE*, vol. 86, no. 11, pp. 2210–2239, Nov. 1998.
[9] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, 2007.
[10] J. Kim and J. W. Woods, "Spatio-temporal adaptive 3-D Kalman filter for video," *IEEE Trans. Image Process.*, vol. 6, no. 3, pp. 414–424, Mar. 1997.
[11] T. Wedi, "Adaptive interpolation filter for motion and aliasing compensated prediction," *Proc. SPIE*, vol. 4671, pp. 415–423, Jan. 2002.
[12] S.-J. Choi and J. W. Woods, "Motion-compensated 3-D subband coding of video," *IEEE Trans. Image Process.*, vol. 8, no. 2, pp. 155–167, Feb. 1999.
[13] J.-R. Ohm, "Three-dimensional subband coding with motion compensation," *IEEE Trans. Image Process.*, vol. 3, no. 5, pp. 559–571, Sep. 1994.
[14] J. Han, V. Melkote, and K. Rose, "Transform-domain temporal prediction in video coding: Exploiting correlation variation across coefficients," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 953–956.
[15] S. Li, T. Nanjundaswamy, Y. Chen, and K. Rose, "Asymptotic closed-loop design for transform domain temporal prediction," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 4907–4911.
[16] S. Li, T. Nanjundaswamy, and K. Rose, "Transform domain temporal prediction with extended blocks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 1476–1480.
[17] B. Vishwanath, T. Nanjundaswamy, and K. Rose, "A deterministic annealing approach to switched predictor design for adaptive compression systems," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 7150–7154.
[18] B. Vishwanath, T. Nanjundaswamy, and K. Rose, "Deterministic annealing based transform domain temporal predictor design for adaptive video coding," in *Proc. Data Compress. Conf. (DCC)*, Mar. 2019, pp. 192–200.
[19] S. Li, Y. Chen, J. Han, T. Nanjundaswamy, and K. Rose, "Rate-distortion optimization and adaptation of intra prediction filter parameters," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 3146–3150.
[20] P. J. M. Laarhoven and E. H. L. Aarts, "Simulated annealing," in *Simulated Annealing: Theory and Applications*. Dordrecht, The Netherlands: Springer, 1987, pp. 7–15.
[21] R. D. Rosenkrantz, *ET Jaynes: Papers on Probability, Statistics and Statistical Physics*, vol. 158. Boston, MA, USA: Springer, 2012.
[22] Y. W. Park, Y. Jiang, D. Klabjan, and L. Williams, "Algorithms for generalized clusterwise linear regression," *INFORMS J. Comput.*, vol. 29, no. 2, pp. 301–317, May 2017.
[23] W. S. DeSarbo, R. L. Oliver, and A. Rangaswamy, "A simulated annealing methodology for clusterwise linear regression," *Psychometrika*, vol. 54, no. 4, pp. 707–736, Sep. 1989.
[24] G. T. Waters, "Sound quality assessment material recordings for subjective tests," in *Users' Handbook for the EBQ-SQAM Compact Disc, European Broadcasting Union, Avenue Albert Lancaster*, vol. 32. Eur. Broadcasting Union Geneva, 1988, p. 1180.
[25] V. Sze, M. Budagavi, and G. J. Sullivan, "High efficiency video coding (HEVC)," in *Integrated Circuit and Systems, Algorithms and Architectures*, vol. 39. New York, NY, USA: Springer, 2014, p. 40.
[26] G. Bjontegaard, *Calculation of Average PSNR Differences Between RD-Curves*, document VCEG-M33 ITU-T Q6/16, Austin, TX, USA, Apr. 2001.
[27] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, vol. 159. New York, NY, USA: Springer, 2012.

Bharath Vishwanath (Student Member, IEEE) received the B.E. degree in electronics and communications engineering from the National Institute of Technology Karnataka, India, in 2014, and the M.Sc. degree in electrical and computer engineering from the University of California at Santa Barbara (UCSB) in 2016, where he is currently pursuing the Ph.D. degree with the Signal Compression Lab. His research interests include video coding, non-convex optimization, and information theory. He has interned with Interdigital Communications Inc., San Diego, from Summer 2016 to 2017, and with Dolby Laboratories in Summer 2019.



Tejaswi Nanjundaswamy (Member, IEEE) received the B.E. degree in electronics and communications engineering from the National Institute of Technology Karnataka, India, in 2004, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of California at Santa Barbara (UCSB), in 2009 and 2013, respectively. He is currently a Postdoctoral Researcher with the Signal Compression Lab, UCSB, where he focuses on audio/video compression, processing and related technologies. He worked at Ittiam Systems, Bengaluru, India, from 2004 to 2008, as a Senior Engineer on audio codecs and effects development. He also interned with the Multimedia Codecs Division, Texas Instruments (TI), India, in 2003. He is an Associate Member of the Audio Engineering Society (AES). He won the Student Technical Paper Award at AES 129th Convention.

Kenneth Rose (Life Fellow, IEEE) received the Ph.D. degree from the California Institute of Technology, Pasadena, in 1991. Then, he joined the Department of Electrical and Computer Engineering, University of California at Santa Barbara, where he is currently a Distinguished Professor. His main research activities are in the areas of information theory and signal processing, and include rate-distortion theory, source and source-channel coding, audio-video coding and networking, pattern recognition, and non-convex optimization. He has published over 350 peer-reviewed articles in these fields. A long-standing interest of his is in the relations between information theory, estimation theory, and statistical physics, and their potential impact on fundamental and practical problems in diverse disciplines. Recently, he was the senior co-author of a paper for which his students received the 2015 IEEE Signal Processing Society Young Author Best Paper Award. His earlier awards include the 1990 William R. Bennett Prize Paper Award of the IEEE Communications Society, as well as the 2004 and 2007 IEEE Signal Processing Society Best Paper Awards.