# A Full Single-Part Results

| # | Params | Rhythm Model | Notes Model | $\text{Loss}_t$ | $\text{Loss}_n$ |
|---|---|---|---|---|---|
| 1 | 112 | $\hat{\mathbf{r}}_{k,0} = \mathbf{bias}_0$ | $\hat{\mathbf{r}}_{k,1,n} = \mathbf{bias}_{1,n}$ | 2.92 | 7.15 |
| 2 | 21k | $\hat{\mathbf{r}}_{k,0} = \mathbf{lin}(\mathbf{r}_{(1)})$ | $\hat{\mathbf{r}}_{k,1,n} = \mathbf{lin}_n(\mathbf{r}_{(1)}, \mathbf{r}_+)$ | 2.00 | 6.05 |
| 3 | 9k | $\hat{\mathbf{r}}_{k,0} = \mathbf{lin}(\mathbf{r}_{(1)})$ | $\hat{\mathbf{r}}_{k,1,n} = \mathbf{lin}(\mathbf{r}_{(1)}, \mathbf{r}_+)$ | 2.00 | 4.29 |
| 4 | 11k | $\hat{\mathbf{r}}_{k,0} = \mathbf{lin}(\mathbf{r}_{(1)}, \ell)$ | $\hat{\mathbf{r}}_{k,1,n} = \mathbf{lin}(\mathbf{r}_{(1)}, \mathbf{r}_+, \mathbf{1}_n)$ | 1.83 | 4.29 |
| 5 | 149k | $\hat{\mathbf{r}}_{k,0} = \mathbf{lin} \circ \mathbf{fc}(\mathbf{r}_{(1)})$ | $\hat{\mathbf{r}}_{k,1,n} = \mathbf{lin}_n \circ \mathbf{fc}(\mathbf{r}_{(1)}, \mathbf{r}_+)$ | 1.99 | 3.93 |
| 6 | 135k | $\hat{\mathbf{r}}_{k,0} = \mathbf{lin} \circ \mathbf{fc}(\mathbf{r}_{(1)})$ | $\hat{\mathbf{r}}_{k,1,n} = \mathbf{lin} \circ \mathbf{fc}(\mathbf{r}_{(1)}, \mathbf{r}_+)$ | 1.99 | 4.07 |
| 7 | 172k | $\hat{\mathbf{r}}_{k,0} = \mathbf{lin} \circ \mathbf{fc}(\mathbf{r}_{(1)}, \ell)$ | $\hat{\mathbf{r}}_{k,1,n} = \mathbf{lin} \circ \mathbf{fc}(\mathbf{r}_{(1)}, \mathbf{r}_+, \mathbf{1}_n)$ | 1.80 | 3.90 |
| 8 | 72k | $\hat{\mathbf{r}}_{k,0} = \mathbf{lin}(\mathbf{r}_{(5)})$ | $\hat{\mathbf{r}}_{k,1,n} = \mathbf{lin}_n(\mathbf{r}_{(5)}, \mathbf{r}_+)$ | 1.86 | 6.05 |
| 9 | 36k | $\hat{\mathbf{r}}_{k,0} = \mathbf{lin}(\mathbf{r}_{(5)})$ | $\hat{\mathbf{r}}_{k,1,n} = \mathbf{lin}(\mathbf{r}_{(5)}, \mathbf{r}_+)$ | 1.86 | 3.91 |
| 10 | 38k | $\hat{\mathbf{r}}_{k,0} = \mathbf{lin}(\mathbf{r}_{(5)}, \ell)$ | $\hat{\mathbf{r}}_{k,1,n} = \mathbf{lin}(\mathbf{r}_{(5)}, \mathbf{r}_+, \mathbf{1}_n)$ | 1.73 | 3.91 |
| 11 | 418k | $\hat{\mathbf{r}}_{k,0} = \mathbf{lin} \circ \mathbf{fc}(\mathbf{r}_{(5)})$ | $\hat{\mathbf{r}}_{k,1,n} = \mathbf{lin}_n \circ \mathbf{fc}(\mathbf{r}_{(5)}, \mathbf{r}_+)$ | 1.64 | 3.26 |
| 12 | 497k | $\hat{\mathbf{r}}_{k,0} = \mathbf{lin} \circ \mathbf{fc}(\mathbf{r}_{(5)})$ | $\hat{\mathbf{r}}_{k,1,n} = \mathbf{lin} \circ \mathbf{fc}(\mathbf{r}_{(5)}, \mathbf{r}_+)$ | 1.64 | 3.16 |
| 13 | 535k | $\hat{\mathbf{r}}_{k,0} = \mathbf{lin} \circ \mathbf{fc}(\mathbf{r}_{(5)}, \ell)$ | $\hat{\mathbf{r}}_{k,1,n} = \mathbf{lin} \circ \mathbf{fc}(\mathbf{r}_{(5)}, \mathbf{r}_+, \mathbf{1}_n)$ | 1.59 | 3.10 |
| 14 | 228k | $\hat{\mathbf{r}}_{k,0} = \mathbf{lin} \circ \mathbf{fc}(\mathbf{f}(\mathbf{r}_{(5)}), \ell)$ | $\hat{\mathbf{r}}_{k,1,n} = \mathbf{lin} \circ \mathbf{fc}(\mathbf{c}(\mathbf{r}_{(5)}), \mathbf{r}_+, \mathbf{1}_n)$ | 1.58 | 3.05 |
| 15 | 134k | $\hat{\mathbf{r}}_{k,0} = \mathbf{lin}(\mathbf{r}_{(10)})$ | $\hat{\mathbf{r}}_{k,1,n} = \mathbf{lin}(\mathbf{r}_{(10)}, \mathbf{r}_+)$ | 1.83 | 6.05 |
| 16 | 71k | $\hat{\mathbf{r}}_{k,0} = \mathbf{lin}(\mathbf{r}_{(10)}, \ell)$ | $\hat{\mathbf{r}}_{k,1,n} = \mathbf{lin}(\mathbf{r}_{(10)}, \mathbf{r}_+, \mathbf{1}_n)$ | 1.71 | 3.83 |
| 17 | 372k | $\hat{\mathbf{r}}_{k,0} = \mathbf{lin} \circ \mathbf{fc}(\mathbf{f}(\mathbf{r}_{(10)}), \ell)$ | $\hat{\mathbf{r}}_{k,1,n} = \mathbf{lin} \circ \mathbf{fc}(\mathbf{c}(\mathbf{r}_{(10)}), \mathbf{r}_+, \mathbf{1}_n)$ | 1.55 | 3.00 |
| 18 | 250k | $\hat{\mathbf{r}}_{k,0} = \mathbf{lin} \circ \mathbf{conv}_5(\mathbf{f}(\mathbf{r}_{(10)}), \ell)$ | $\hat{\mathbf{r}}_{k,1,n} = \mathbf{lin} \circ \mathbf{conv}_5(\mathbf{c}(\mathbf{r}_{(10)}), \mathbf{r}_+, \mathbf{1}_n)$ | 1.55 | 3.01 |
| 19 | 769k | $\hat{\mathbf{r}}_{k,0} = \mathbf{lin} \circ \mathbf{conv}_3 \circ \mathbf{conv}_5(\mathbf{f}(\mathbf{r}_{(10)}), \ell)$ | $\hat{\mathbf{r}}_{k,1,n} = \mathbf{lin} \circ \mathbf{conv}_3 \circ \ \mathbf{conv}_5(\mathbf{c}(\mathbf{r}_{(10)}), \mathbf{r}_+, \mathbf{1}_n)$ | 1.50 | 2.92 |
| 20 | 342k | $\hat{\mathbf{r}}_{k,0} = \mathbf{lin} \circ \mathbf{rnn}(\mathbf{r}_{(10)}, \ell)$ | $\hat{\mathbf{r}}_{k,1,n} = \mathbf{lin} \circ \mathbf{rnn}(\mathbf{r}_{(10)}, \mathbf{r}_+, \mathbf{1}_n))$ | 1.48 | 2.89 |
| 21 | 283k | $\hat{\mathbf{r}}_{k,0} = \mathbf{lin} \circ \mathbf{rnn}(\mathbf{f}(\mathbf{r}_{(10)}), \ell)$ | $\hat{\mathbf{r}}_{k,1,n} = \mathbf{lin} \circ \mathbf{rnn}(\mathbf{c}(\mathbf{r}_{(10)}), \mathbf{r}_+, \mathbf{1}_n))$ | 1.48 | 2.88 |
| 22 | 301k | $\hat{\mathbf{r}}_{k,0} = \mathbf{lin} \circ \mathbf{rnn}(\mathbf{f}(\tilde{\mathbf{r}}_{(10)}), \ell)$ | $\hat{\mathbf{r}}_{k,1,n} = \mathbf{lin} \circ \mathbf{rnn}(\mathbf{c}(\tilde{\mathbf{r}}_{(10)}), \mathbf{r}_+, \mathbf{1}_n))$ | 1.59 | 2.93 |

Table 1: Single-part results. Loss is the cross-entropy described in Section 3.1. $\text{Loss}_t$ and $\text{Loss}_n$ are decompositions of the loss described in Section 3.2. For succinctness, define $\mathbf{r}_{(m)} \equiv \mathbf{r}_{k-m:k}$ (a truncated history of length $k$) and $\mathbf{r}_+ \equiv \mathbf{r}_{k,0} \oplus \mathbf{r}_{k,1,1:n}$ (the current frame, masked above pitch $n$). For definition of $\mathbf{r}$ see Section 4, Sequential part factorization. $\mathbf{lin}_n$ indicates a log-linear classifier (sigmoid for $\hat{y}_n$ and softmax for $\hat{y}_t$) and **lin** indicates the relative pitch log-linear classifier and inputs $\mathbf{1}_n$ indicate pitch-class features (Section 5.2, Relative pitch). The inputs $\ell$ indicate location features (Section 5.2, Autoregressive modeling). **fc** indicates a fully connected layer. **f** and **c** indicates pitch embeddings (Section 5.2, Pitch embeddings). $\mathbf{conv}_k$ indicates 1d convolution of width $k$. **rnn** indicates a recurrent layer. All hidden layers are parameterized with 300 nodes. Models were regularized with early stopping when necessary. The subscript $k$ on the history tensor $x_k$ indicates the number of frames of history used in each experiment (either 1, 5, or 10 frames). $\tilde{\mathbf{r}}_{(m)}$ is a modified history discussed in Section 5.1.

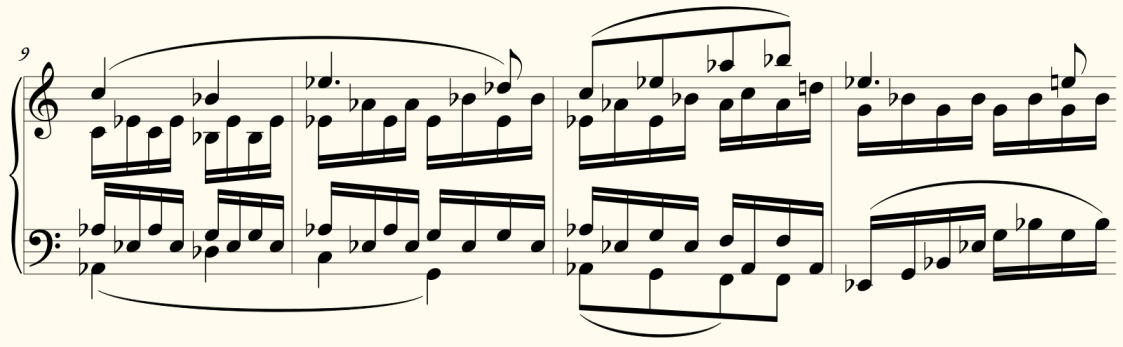

Figure B.1: Beethoven's piano sonata number 8 (Pathetique) movement 2, from measure 9, rendered by the Verovio Humdrum Viewer. Although visually rendered on two staves, this sonata consists of four parts: a high sequence of quarter and eighth notes, two middle sequences of sixteenth notes, and a low sequence of quarter notes.

# B Piano Music

For some piano music, it is necessary to draw a distinction between an instrument and a part. Consider the piano score given in Figure B.1. This single piano part is more comparable to a complete score than the individual parts of, for example, a string quartet (compare the piano score in Figure B.1 to the quartet score in Figure 1 in the main text). Indeed, an educated musician would read this score in four distinct parts: a high sequence of quarter and eighth notes, two middle sequences of sixteenth notes, and a low sequence of quarter notes. In measure 12, the lowest two parts combine into a single bass line of sixteenth notes.

These part divisions are indicated in score through a combination of beams, slurs, and other visual queues. We do not model these visual indicators; instead we rely on part annotations provided by the KernScores dataset. The provision of these annotations is a strong point in favor of the KernScores dataset's Humdrum format; although in principle formats like MIDI can encode this information, in practice they typically collect all notes for a single instrument into a single track, or possibly two tracks (for the treble and bass staves, as seen in the figure) in the case of piano music.

In extremely rare cases, this distinction between instrument and part must also be made for stringed instruments; a notable example is Beethoven's string quartet number 14, in the fourth movement in measures 165 and 173, where the four instruments each separate into two distinct parts creating brief moments of 8-part harmony. The physical constraints of stringed instruments discourage more widespread use of these polyphonies. For vocal music, of course, physical constraints prevent intra-instrument polyphony entirely.

As Figure B.1 illustrates, these more abstract parts can weave in and out of existence. Two parts can merge with each other; a single part can split in two; new parts can emerge spontaneously. The KernScores data provides annotations that describe this behavior. We can represent these dynamics of parts as a $P \times P$ flow matrix at each time step ($P$ is an upper bound on the number of parts; for the KernScores corpus used in this work, we take $P = 6$) that describes where each part moves in the next step. At most time steps, this flow matrix is the identity matrix.

The state-based models discussed in this paper can easily be adjusted to accommodate these flows. If two parts merge, sum their states; if a part splits in two, duplicate its state. These operations amount to hitting the vector of state estimates for the parts with the flow matrix at each time step. However, we do not currently model the flow matrix. Because the flow matrix for piano music contains some (small) amount of entropy, we therefore exclude piano music from the results reported in Table 4. We do however include the piano music in training.

# C Piano-Roll Representations of Scores

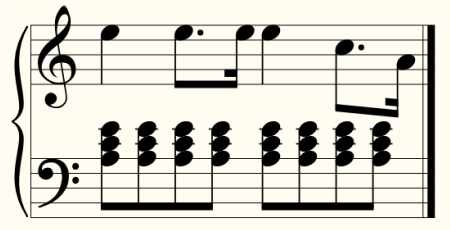

Figure C.1: Mozart's piano sonata number 8 in A minor, movement 1, from measure 1.

In Section 3.1 we defined a score as a $T \times P \times 2N$ binary tensor, where at each time $t \in T$ in each part $p \in P$, we have two values $\mathbf{x}_{t,p,n}$ and $\mathbf{x}_{t,p,N+n}$ to indicate whether note $n \in N$ is present and whether $n$ begins respectively. While classical piano-roll representations omit the second onset bit, both bits are necessary to faithfully represent a musical score. Consider, for example, Figure C.1. Two scores are demonstrated in Figure C.2 that have identical piano-roll encodings if only a single bit is used to indicate the presence of a note. Many other scores also alias to this same piano-roll encoding. The addition of an onset bit delineates the boundaries between multiple notes of the same pitch, thus resolving this ambiguity.

Figure C.2: Two scores with the same piano-roll representation as the score in Figure B.2. The popular dataset introduced by Boulanger-Lewandowski et al. (2012) uses this single-bit representation. A second bit is used in some more recent work, for example Liang et al. (2017) in which they are referred to as "Tie" bits).

Another pitfall of piano-roll representations is the choice of discretization. In Section 3.1, we defined a continuous-time process with a real-valued index $t$. To use a piano-roll for factorization or featurization, a finite resolution must be chosen. We argued in Section 3.2 that this discretization $\Delta$ is information-preserving, so long as $\Delta$ is chosen to be the resolution of the score process or finer. The consequences of choosing a discretization that is too coarse is illustrated by Figure C.3.

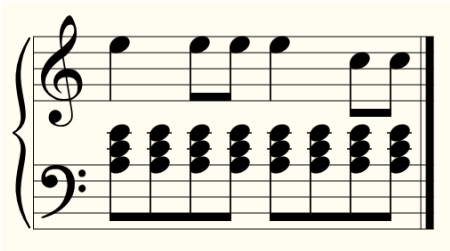

Figure C.3: A corruption of the score from Figure C.1, discretized at eighth-note resolution.

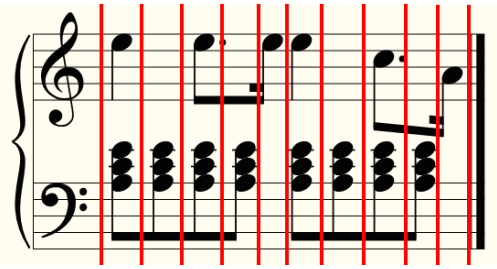

Figure D.1: The Mozart from Figure C.1, with red lines that indicate the boundaries of events under a run-length factorization of the score. Notes in the treble staff are chopped up into eight-note runs, so instead of predicting note durations (quarter, dotted-eighth, sixteenth, etc.) we instead predict fragments of notes (eighth, continue eighth, continue eighth, etc.).

# D Run-Length Factorization

Training and sampling from a model over a discrete factorization of scores at the process resolution $\Delta$ can be expensive, prompting some earlier works to discretize at a coarser resolution (as discussed in the previous section). One approach to preserve fine rhythmic structure (e.g. triplets and thirty-second notes) without committing to a fine discretization is to factor a score into run-lengths. To this end, we define a run-length encoded score $\mathbf{x} \in \left(\mathbb{N} \oplus \{0,1\}^{P \times 2N}\right)^T$ where, at each time index $t \in \{1, \ldots, T\}$, we set

| | |
|---|---|
| $\mathbf{x}_{t,0} = \mathbf{1}_{d_t}$, | where $d_t$ is the duration of the event at time index $t$, |
| $\mathbf{x}_{t,1,p,n} = 1$ | iff note $n$ is on at time $t$ in part $p$, |
| $\mathbf{x}_{t,1,p,2n} = 1$ | iff note $n$ begins at time $t$ in part $p$. |

The sequence $\mathbf{x}_t$ is non-linear in the index $t$: entry $\mathbf{x}_{t+1}$ occurs $d_t$ beats after entry $\mathbf{x}_t$, in contrast to the raster where $\mathbf{x}_{t+1}$ always occurs a constant interval $\Delta$ after $\mathbf{x}_t$.

We can then factor the distribution over scores into conditional distributions over binary note values and natural-number duration values:

$$p(S) = \prod_{t=1}^{T} p(\mathbf{x}_t | \mathbf{x}_{1:t}) = \prod_{t=1}^{T} p(\mathbf{x}_{t,0} | \mathbf{x}_{1:t}) \prod_{p=1}^{P} \prod_{n=1}^{2N} p(\mathbf{x}_{t,p,n} | \mathbf{x}_{1:t}, \mathbf{x}_{t,0}, \mathbf{x}_{t,1,1:p}, \mathbf{x}_{t,1,p,1:n}).$$

Because music typically doesn't evolve at the finest possible resolution $\Delta$, we save a substantial amount of computation by predicting run-lengths $\mathbf{x}_{t,0} \in \mathbb{N}$ rather than re-iterating the predictions $\mathbf{x}_{t,p,n}$ at successive time steps.

One criticism of the run-length factorization is that, when notes of different durations overlap in a score, the longer notes are chopped up along the boundaries of the short notes as illustrated in Figure D.1. Rather than predicting musically meaningful quantities like note values (quarter, eighth, dotted-eighth, etc.) instead we predict run-length chunks.

# E User Study Details

To understand our model qualitatively, we asked 20 study participants to evaluate compositions produced by one of our best models: experiment 6 from Table 4. Each user was asked to listen to 5 sets of 10 audio clips, synthesized from scores ranging from 10 to 50 frames of composition (a frame is a run-length as defined by the representation discussed in Section 5.1). Every user was presented with their own set of audio clips, randomly sampled from either the training set or the model. Users were given the following prompt before beginning the study:

> This is a musical Turing test. You will be presented with a selection of audio clips, beginning with short clips and progressing to longer clips. For each audio clip, you will be asked whether you believe the clip was composed by a human or a computer. Half the clips you will be presented with belong in each category. This data contains many famous classical compositions, ranging from the Renaissance to early 20th century. If you specifically recognize a piece, please let me know. Finally, all recordings you hear–both human and artificial–are performed at a tempo of 120bpm.

Additionally, we asked users two questions about their background:

- Do you self-identify as musically educated? (8 responded 'yes')
- Do you self-identify as educated in machine learning? (13 responded 'yes')

Table 2 summarizes results of our listening study, including conditional results for the educated subgroups.

| Frames | 10 | 20 | 30 | 40 | 50 |
|---|---|---|---|---|---|
| All | 5.3 | 5.7 | 6.6 | 6.7 | 6.8 |
| Music | 4.9 | 6.0 | 6.4 | 6.9 | 7.0 |
| ML | 4.8 | 5.5 | 6.2 | 6.7 | 6.8 |

Table 2: Qualitative evaluation of the 10-frame hierarchical model: Experiment 6 in Table 4. Twenty participant were asked to judge 50 audio clips each of varying length. The scores indicate participants' average correct discriminations out of 10 (5.0 would indicate random guessing; 10.0 would indicate perfect discrimination). The categories indicate breakdowns for listeners who identified as educated in music or educated in machine learning.

We asked users to identify pieces if they specifically recognized them, because we were concerned that this knowledge of the classical music canon could confound the question of musical plausibility of our model's samples. In the end, only one user positively identified a piece in our study. This may be explained because our models do not predict tempo. Therefore, to make fair comparisons between human compositions and model outputs, we synthesized all scores at a tempo of 120bpm. This may serve to obscure recognizable pieces. However, it also makes the task less informative because all audio clips in the listening test sound less like "real music." Participants were informed of this fact, but it is not clear how effectively they could use this knowledge.

# F Evaluation Metric Details

We demonstrate here the claim from the text that any refinement of the $\Delta$-interval constant discretization of the support of the distribution $p$ over scores yields no further contributions to the cross entropy. Formally, if $\mathcal{P} = (0, \Delta, 2\Delta, \ldots, T)$ is the discrete partition of the interval $[0, T]$ and $\mathcal{R}$ is any refinement of this partition (i.e. a partition of $[0, T]$ such that contains the points of $\mathcal{P}$) then the following proposition holds.

**Proposition 1.** *For any refinement $\mathcal{R}$ of $\mathcal{P}$, $H_{\mathcal{P}}(p||q) = H_{\mathcal{R}}(p||q)$.*

*Proof.* Let $K'$ denote the size of $\mathcal{R}$. By definition of relative entropy of $p$ restricted to the partition $\mathcal{R}$,

$$H_{\mathcal{R}}(p||q) = \mathop{\mathbb{E}}_{\mathbf{x}\sim p} \log q(\mathbf{x}_{\mathcal{R}_1}, \mathbf{x}_{\mathcal{R}_1}, \ldots, \mathbf{x}_{\mathcal{R}_{K'}}).$$

And applying the chain rule for conditional probabilities,

$$H_{\mathcal{R}}(p||q) = \sum_{k=1}^{K'} \mathop{\mathbb{E}}_{\mathbf{x}\sim p} \log q(\mathbf{x}_{\mathcal{R}_k} | \mathbf{x}_{\mathcal{R}_1}, \ldots, \mathbf{x}_{\mathcal{R}_{k-1}}).$$

Consider terms $q(\mathbf{x}_{\mathcal{R}_k} | \mathbf{x}_{\mathcal{R}_1}, \ldots, \mathbf{x}_{\mathcal{R}_{k-1}})$ where $\mathcal{R}_k \notin \mathcal{P}$. There exists some $n$ such that $n\Delta < \mathcal{R}_k < (n+1)\Delta$. We must have $\mathbf{x}_{\mathcal{R}_k} = \mathbf{x}_{n\Delta}$ because by definition of $\Delta$, all change-points in $\mathbf{x}$ occur at integer multiples of $\Delta$. Because $\mathcal{R}$ is a refinement of $\mathcal{P}$ and $n\Delta \in \mathcal{P}$, it follows that $n\Delta \in \mathcal{R}$. Furthermore, $n\Delta < \mathcal{R}_k$ and therefore $n\Delta \in (\mathcal{R}_0, \ldots, \mathcal{R}_{k-1})$. We conclude that

$$\mathop{\mathbb{E}}_{\mathbf{x}\sim p} \log q(\mathbf{x}_{\mathcal{R}_k} | \mathbf{x}_{\mathcal{R}_1}, \ldots, \mathbf{x}_{\mathcal{R}_{k-1}}) = \mathop{\mathbb{E}}_{\mathbf{x}\sim p} \log q(\mathbf{x}_{\mathcal{R}_k} | \mathbf{x}_{n\Delta}, \ldots) = 0.$$

In words: conditioned on $\mathbf{x}_{n\Delta}$, $\mathbf{x}_{\mathcal{R}_k}$ is known and its relative entropy vanishes. Dropping all such terms $k$ with $\mathcal{R}_k \notin \mathcal{P}$ we see that

$$H_{\mathcal{R}}(p||q) = \sum_{k\,:\,\mathcal{R}_k \in \mathcal{P}} \mathop{\mathbb{E}}_{\mathbf{x}\sim p} \log q(\mathbf{x}_{\mathcal{R}_k} | \mathbf{x}_{\mathcal{R}_0}, \ldots, \mathbf{x}_{\mathcal{R}_{k-1}})$$

$$= \sum_{k=1}^{T/\Delta} \mathop{\mathbb{E}}_{\mathbf{x}\sim p} \log q(\mathbf{x}_{k\Delta} | \mathbf{x}_0, \ldots, \mathbf{x}_{(k-1)\Delta}) = \mathop{\mathbb{E}}_{\mathbf{x}\sim p} \log q(\mathbf{x}_0, \mathbf{x}_\Delta, \ldots, \mathbf{x}_T) = -T\Delta H(p||q). \qquad \square$$