Stochastic Thermodynamics for Autoregressive Generative Models: A Non-Markovian Perspective

Takahiro Sagawa [*]

Department of Applied Physics, The University of Tokyo, Bunkyo-ku, Tokyo 113-8656, Japan;
Quantum-Phase Electronics Center (QPEC), The University of Tokyo, Bunkyo-ku, Tokyo 113-8656, Japan;
and Inamori Research Institute for Science (InaRIS), Kyoto-shi, Kyoto 600-8411, Japan

(Received 16 April 2026; accepted 10 September 2026; published 5 November 2026)

Autoregressive generative models—including Transformers, recurrent neural networks, classical Kalman filters, state space models, and Mamba—all generate sequences by sampling each output from a deterministic summary of the past, producing genuinely non-Markovian observed processes. We develop a general theoretical framework based on stochastic thermodynamics for this class of architectures and introduce the entropy production, which can be efficiently estimated from sampled trajectories without exponential overhead, despite the non-Markovian nature of the observed dynamics. As a proof-of-concept experiment with a large language model (LLM), we evaluate the entropy production for a pretrained Transformer-based model, GPT-2. We find that the token-level entropy production is dominated by a syntactic artifact, while the sentence-level entropy production tends to be larger for causally ordered than for noncausal text sets. This observation is supported by a reevaluation with a substantially larger model, Qwen3-4B-Base. We also demonstrate the framework in the linear Gaussian case, where the model reduces to the Kalman innovation representation and the entropy production admits an analytical expression. We also show that the entropy production decomposes exactly into non-negative per-step contributions in terms of retrospective inference, and each of those terms further splits into information-theoretically meaningful terms: a compression loss and a model mismatch. Our results establish a bridge between stochastic thermodynamics and modern generative models, and provide a starting point for using irreversibility as a quantitative probe of the highly non-Markovian processes generated by models such as LLMs.

DOI: 10.1103/tv38-b23y

I. INTRODUCTION

Stochastic thermodynamics provides a general framework for quantifying irreversibility in stochastic processes, with entropy production playing the role of a central diagnostic [1–5]. Although the theory is most fully developed for Markovian processes, substantial progress has extended it in several complementary directions. One line of work characterizes irreversibility directly from the forward and backward path measures, including for non-Markovian time series [6–14]. A related line of work studies non-Markovian or partially observed entropy production through various constructions [15–23]. In parallel, the interplay among thermodynamics, information, and computation has been extensively studied [24–35]. Stochastic thermodynamics of neural networks has also been explored, including in the context of the learning process [36], and more recently for dense associative memory (modern Hopfield) networks [37].

Autoregressive generative models have become the central architecture of modern generative AI. These models generate a sequence of elements, with each new element drawn from a conditional distribution that depends on a deterministic summary of the preceding observations. For a language model, each such element is a token: a discrete unit such as a word, a subword, or a punctuation mark. The Transformer architecture [38–41], which underlies most contemporary large language models (LLMs), uses causal self-attention to construct a context vector from the full past sequence; the deterministic summary cannot in general be reduced to a fixed-order recursive update, so the resulting observed token sequence is genuinely non-Markovian. Recurrent neural networks (RNNs) [42,43] are also regarded as autoregressive generative models, but have a simpler structure with recursive updates of latent states. The Kalman filter [44–46], long studied in control theory and signal processing, can be viewed as a generative model of this recursive class. Moreover, structured state space models (SSMs) and Mamba (an alternative architecture for LLMs) [47–49] also fall into this recursive class. Despite their diverse origins, all of these architectures share the same abstract structure: a stochastic emission from a deterministic latent state that encodes past observations.

In this paper, we develop a stochastic thermodynamic framework for the class of non-Markovian processes generated by autoregressive models with deterministic internal memory. This class encompasses diverse architectures as described above; we bring them under a single framework (Table I) and develop the stochastic thermodynamics of the resulting non-Markovian observed process in a unified manner, by constructing a backward process with the same architectural components applied in reversed temporal order.

[*]Contact author: sagawa@ap.t.u-tokyo.ac.jp

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

TABLE I. Correspondence between our general framework and representative architectures. In all cases, y_t denotes the observed variable (token or measurement). The last column indicates whether the recursive special case holds [$h_t = \phi(h_{t-1}, y_t)$].

Model	h_t	Φ_t or ϕ	$p_t(y_{t+1} \mid h_t)$	Recursive
Transformer	Attention context	$\Phi(y_1, \ldots, y_t)$ (full sequence)	$\mathrm{softmax}(W_{\mathrm{out}} h_t)_{y_{t+1}}$	×
RNN	RNN latent state	$\phi(h, y) = \tanh(W_h h + W_y y + b)$	$\mathrm{softmax}(W_{\mathrm{out}} h_t + b_{\mathrm{out}})_{y_{t+1}}$	✓
Kalman	$\hat{x}_{t+1\mid t}$	$\phi(h, y) = A[(I - KC)h + Ky]$	$\mathcal{N}(Ch_t, S)$	✓
SSM	SSM state	$\phi(h, y) = Ah + By$	$\mathrm{softmax}(W_{\mathrm{out}} C h_t)_{y_{t+1}}$	✓
Mamba	$h'_t \equiv (h_t,\ y_t)$	$\phi'(h', y) = (A(y)\, h + B(y)\, y,\ y)$	$p(y_{t+1} \mid h'_t) = \mathrm{softmax}(W_{\mathrm{out}}\, C(y_t)\, h_t)_{y_{t+1}}$	✓

Conceptually, our definition of the entropy production builds on a line of research in which the Kullback-Leibler (KL) divergence between forward and backward path measures serves as an operational characterization of irreversibility for the observed process [6–14]. The central observation about the autoregressive class studied here is that every factor in the forward/backward path-probability ratio is supplied explicitly by the emission kernel evaluated at a deterministic latent state, making the stochastic entropy production computable from a single sampled trajectory despite genuine non-Markovianity.

As a proof-of-concept experiment with a pretrained LLM, we evaluate the stochastic entropy production for GPT-2. We show that the token-level entropy production is dominated by a syntactic artifact of token-order reversal, whereas the block-level coarse-grained entropy production, which reverses the order of sentences rather than individual tokens, tends to be larger for causally ordered than for noncausal texts. Additionally, a reevaluation is performed with a substantially larger model, Qwen3-4B-Base, which reproduces the qualitative pattern with improved statistical resolution.

LLMs are often argued to acquire internal representations of the real world, including spatial and temporal structure—a capacity sometimes discussed in terms of a learned *world model* [50,51]. From this perspective, the entropy production of an LLM ultimately aims to probe the irreversibility of the real-world process described by the token sequence—ideally, the temporal or causal ordering of the events narrated in the text. Our GPT-2 and Qwen3-4B-Base experiments should be viewed as a first step toward assessing this possibility.

Meanwhile, as an analytically tractable demonstration, we examine the linear Gaussian case, where the autoregressive model coincides with the innovation representation of the Kalman filter. We derive an analytical expression for the entropy production, which is numerically verified by applying the Monte Carlo sampling procedure.

Furthermore, we show that the entropy production $\mathcal{S}_y$ admits an exact *retrospective* decomposition into a sum of non-negative per-step contributions $\mathcal{D}_t \geqslant 0$, each measuring how well the backward model retrodicts the present observation y_t from the future. Each $\mathcal{D}_t$ further splits into a *compression loss* $\mathcal{L}_t$, arising when the backward latent state is a lossy summary of the future, and a *model mismatch* $\mathcal{M}_t$, arising because the emission kernel designed for forward prediction is reused in the backward direction. The decomposition is formally reminiscent of the evidence lower bound (ELBO) decompositions used in variational inference [52–54], but arises from a fundamentally different starting point (namely, time reversal and entropy production). This connection suggests that the stochastic-thermodynamic and machine-learning perspectives on generative models may benefit from further mutual exchange.

This paper is organized as follows. In Sec. II, we formulate the general autoregressive framework with deterministic latent memory. In Sec. III, we construct the backward process by reusing the architectural components in reversed temporal order and define the entropy production as the KL divergence between the forward and backward path measures. In Sec. IV, we analyze the computational cost of estimating the entropy production, show that the autoregressive structure renders it tractable without exponential sampling overhead, and introduce a temporal coarse-graining. In Sec. V, we present a proof-of-concept experiment with GPT-2, evaluating both the token-level and block-level entropy production. In Sec. VI, we illustrate the framework in the linear Gaussian case, where the model reduces to the Kalman innovation representation and the entropy production admits an analytical expression. In Sec. VII, we derive an exact decomposition of the entropy production into non-negative per-step contributions, each further splitting into a compression loss and a model mismatch, and discuss the connection to the thermodynamics of information. We conclude with a summary and perspectives in Sec. VIII.

II. SETUP

In this section, we introduce a general framework for autoregressive generative models with deterministic latent memory, and show that Transformers, RNNs, Kalman filters, SSMs, and Mamba all fall into this class (Table I). While each of these concrete architectures is well known, establishing such a unified framework itself constitutes one of the contributions of this paper.

A. State variables and forward process

Consider the following stochastic process. The variables are y_t $(t = 1, 2, \ldots, T)$ and h_t $(t = 0, 1, \ldots, T)$, where h_0 is a fixed initial condition. These variables can be discrete or continuous in our general setup; in the continuous case, summations appearing below should be replaced by integrals.

The dynamics proceeds as follows. At each time step:

(1) h_t is updated deterministically: $h_t = \Phi_t(y_1, y_2, \ldots, y_t)$ for $t = 1, 2, \ldots, T$, and $h_0 = \Phi_0(\,)$ is set to a constant.

(2) y_{t+1} is drawn from the conditional distribution $p_t(y_{t+1} \mid h_t)$, also called the *emission kernel*, for $t = 1, \ldots, T-1$, and $p_0(y_1 \mid h_0) \equiv p(y_1)$ is the initial distribution.

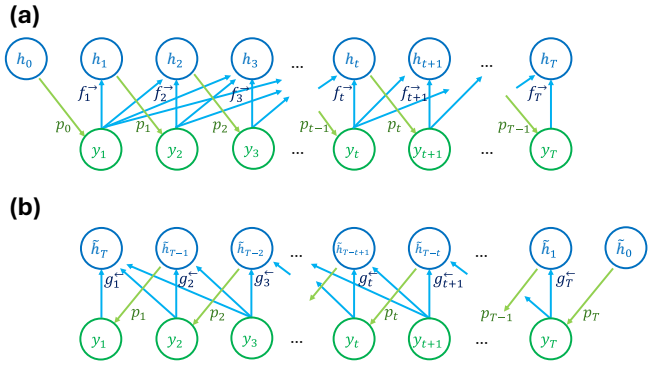


FIG. 1. Schematic of the general causal structure of our setup for the general (nonrecursive) case. (a) The forward process (3) and (b) the backward process (13). The blue arrows indicate deterministic functions, while the green arrows indicate stochastic influences. This figure illustrates the particular realization $\tilde{y}_s = y_{T-s+1}$; even in this case, $\tilde{h}_s \neq h_{T-s+1}$ in general.

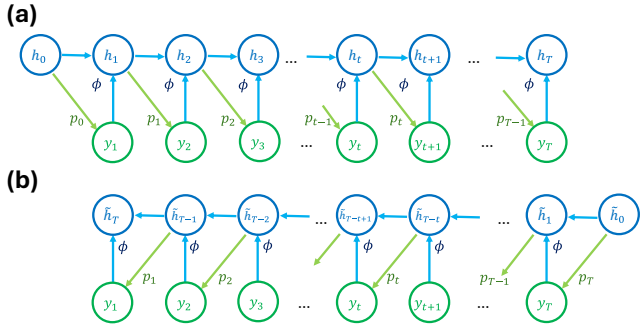


FIG. 2. Schematic of the causal structure for the recursive case. (a) The forward process [Eq. (3) with Eq. (4)] and (b) the backward process [Eq. (13) with Eq. (20)]. The blue arrows indicate deterministic functions, while the green arrows indicate stochastic influences. By following the arrows, one can see that this recursive diagram is a special case of the general diagram (Fig. 1). This figure illustrates the particular realization $\tilde{y}_s = y_{T-s+1}$; even in this case, $\tilde{h}_s \neq h_{T-s+1}$ in general.

Here, each Φ_t is a deterministic map that accepts a variable-length sequence of observations and returns a latent state. The subscript t indexes parameters of the map (e.g., its learnable parameters), but *not* the length of the input sequence; indeed, in the backward process (Sec. III A), Φ_t will be applied to an input whose length differs from t. For Transformers, Φ_t is independent of t (i.e., the same attention mechanism with shared parameters is applied at every step) and can process input sequences of arbitrary length within the context window.

The marginal process y_t is in general non-Markovian, since the latent state h_t accumulates information from past observations. The joint process (h_t, y_t) is generally not Markovian either, because $h_t = \Phi_t(y_1, \ldots, y_t)$ cannot in general be determined from (h_{t-1}, y_t) alone.

We now rewrite the update of the latent state h_t as

$$h_t = f_t^{\rightarrow}(y_{1:t}), \tag{1}$$

where $y_{1:t} \equiv (y_1, y_2, \ldots, y_t)$ and

$$f_t^{\rightarrow}(y_{1:t}) \equiv \Phi_t(y_1, \ldots, y_t). \tag{2}$$

The reason why we introduce the separate notation $f_t^{\rightarrow}$ is that Φ_t will be used for the definition of the backward process as well.

The forward process generates the sequence $y_{1:T} = (y_1, y_2, \ldots, y_T)$ by iterating the update-emission rule above. Its path probability is

$$P_{\rightarrow}(y_{1:T}) = \prod_{t=0}^{T-1} p_t(y_{t+1} \mid f_t^{\rightarrow}(y_{1:t})). \tag{3}$$

See Fig. 1(a) for a graphical representation.

We note that if one simply defined $h_t = (y_1, y_2, \ldots, y_t)$ (i.e., Φ_t were set to the identity map on the entire history), the resulting model could represent an arbitrary non-Markovian process on y_t. However, the important restriction of our architecture lies in treating h_t as having a *fixed finite state space* (or, for continuous variables, a *fixed finite dimensionality*) independent of t: h_t must encode the growing history $y_{1:t}$ into a representation of bounded size. Note that this constraint concerns the size of h_t alone; the input sequence to Φ_t may be of arbitrary length.

Because the emission kernel $p_t(y_{t+1} \mid h_t)$ depends on the history only through the deterministic latent state h_t, each factor in the forward (and backward) path probabilities (3) [and Eq. (13)] is directly evaluable for any given trajectory, and the entropy production $\mathcal{S}_y$ can be estimated by Monte Carlo sampling without an exponential cost from enumerating histories, as shown in Sec. IV.

Moreover, the latent state h_t is not merely a convenient computational device but plays the role of a *sufficient statistic* of the past $y_{1:t}$ for predicting the next observation y_{t+1}. Indeed, the conditional distribution of y_{t+1} given the entire history $y_{1:t}$ factorizes through h_t by construction: $P_{\rightarrow}(y_{t+1} \mid y_{1:t}) = p_t(y_{t+1} \mid h_t)$, which is precisely the defining property of a sufficient statistic. Thus, the bounded-size compression from $y_{1:t}$ to h_t incurs no loss of predictive information in the forward direction, regardless of how long the history is. The interplay between sufficient statistics and entropy production in Markovian systems has been studied in Ref. [55].

B. Recursive case

An important special case is that h_t is determined only by (h_{t-1}, y_t) through a deterministic function ϕ as

$$h_t = \phi(h_{t-1}, y_t). \tag{4}$$

Here, we assume that ϕ is time independent, but the emission kernel p_t may still depend on time. Correspondingly, Φ factors as

$$\Phi(y_1, \ldots, y_t) = \phi(\Phi(y_1, \ldots, y_{t-1}), y_t). \tag{5}$$

The graphical representation of the dependency structure is shown in Fig. 2(a).

In this case, the joint process (h_t, y_t) is Markovian (see also Appendix A). Moreover, the marginal dynamics of h_t alone is Markovian if y_t is marginalized, as is the case for the predicted state $\hat{x}_{t+1|t}$ in the Kalman filter. Even so, the marginal dynamics of y_t is non-Markovian in general.

Most of the results below hold in the general (nonrecursive) setting; when additional structure specific to the recursive case is relevant, it will be noted explicitly.

C. Examples

We show that several well-known architectures can indeed be regarded as special cases of our general setup. In all cases below, implementation details (e.g., layer normalization, multihead decomposition, and residual connections for Transformers) are omitted, and only a simplified description is given. The correspondence between the general framework and each example is summarized in Table I.

Transformers. In a simplified autoregressive Transformer, the observed variable y_t is a discrete token taking values in a finite vocabulary $\mathcal{Y}$. Each token is first mapped to a continuous vector by an embedding matrix $E \in \mathbb{R}^{d\times|\mathcal{Y}|}$, and causal self-attention then computes a context vector $h_t = \Phi(y_1, \ldots, y_t) \in \mathbb{R}^d$ from the full past sequence. (Concretely, the token $y_t \in \mathcal{Y}$ is first encoded as a one-hot vector $e_{y_t} \in \{0,1\}^{|\mathcal{Y}|}$, and the embedding is the matrix-vector product Ee_{y_t}; the variable y_t itself remains a discrete label throughout the framework.) Thus, y_t is discrete while h_t is continuous; the embedding step is absorbed into the deterministic map Φ. The next token is drawn from the emission kernel $p(y_{t+1} \mid h_t) = \mathrm{softmax}(W_{\mathrm{out}}\, h_t)_{y_{t+1}}$, where the subscript y_{t+1} selects the component of the softmax vector corresponding to the token $y_{t+1} \in \mathcal{Y}$.

Since Φ accesses the entire history and cannot in general be reduced to a function of (h_{t-1}, y_t), this is an instance of the general (nonrecursive) case: h_t corresponds to the general map Φ, and the joint process (h_t, y_t) is not Markovian. Note that the model parameters for Φ_t and p_t are both time-independent in Transformers. (While the positional encoding added to each token embedding depends on the token's position within the input sequence to Φ, this position is the argument index of the function Φ and not its external time label t.)

We remark that, in a multilayer Transformer, the key-value pairs cached at each layer up to step t (the KV cache) constitute an alternative representation of the past. If one redefines the latent state as the full KV cache, denoting it by h_t^{KV}, then the update at step t computes new key-value pairs at each layer from h_{t-1}^{KV} and y_t, and the emission kernel $p(y_{t+1} \mid h_t^{\mathrm{KV}})$ is read off from the final-layer output. The resulting update $h_t^{\mathrm{KV}} = \varphi(h_{t-1}^{\mathrm{KV}}, y_t)$ thus formally satisfies the recursive structure. However, h_t^{KV} consists of t key-value pairs per layer per head, so the dimensionality of the state space grows linearly with t, violating the fixed-size requirement on h_t for large t.

RNN. An Elman-type RNN updates its latent state by a two-argument recurrence $h_t = \phi(h_{t-1}, y_t) = \tanh(W_h h_{t-1} + W_y y_t + b)$ (a nonlinear activation function) and predicts via $p(y_{t+1} \mid h_t) = \mathrm{softmax}(W_{\mathrm{out}} h_t + b_{\mathrm{out}})_{y_{t+1}}$. This is the recursive special case (Sec. II B). As in the Transformer case, y_t is a discrete token and $h_t \in \mathbb{R}^d$ is a continuous latent vector; the product $W_y\, y_t$ implicitly performs embedding via the one-hot encoding of y_t.

Steady-state Kalman filter. Consider a linear Gaussian system $x_{t+1} = Ax_t + w_t$, $y_t = Cx_t + v_t$ with $w_t \sim \mathcal{N}(0, Q)$, $v_t \sim \mathcal{N}(0, R)$ independent. The true state x_t has no counterpart in our framework; we interpret the Kalman filter as a generative model that reproduces the trajectory distribution of y_t.

In the innovation representation of the Kalman filter, we set h_{t-1} as $\hat{x}_{t|t-1}$ (the one-step-ahead prediction). We also assume that the Kalman filter has reached steady state. Then, $p(y_t \mid h_{t-1}) = \mathcal{N}(Ch_{t-1}, S)$, with $S = CPC^\top + R$, where P is the steady-state prediction error covariance. The recursive map is the linear update $\phi : (h, y) \mapsto A[(I - KC)h + Ky]$, where $K = PC^\top S^{-1}$. This is the recursive special case with linear activation and Gaussian output. In contrast to the Transformer and RNN cases, both y_t and h_t are continuous variables here (see Sec. VI for details).

SSM. By "SSM," we mean a discrete-time linear state space model layer (e.g., Refs. [47,48]) whose parameters A, B, and C are time independent. The latent state is updated by $h_t = Ah_{t-1} + By_t$, and the next-token distribution is obtained from Ch_t via an output projection and softmax. This is the recursive special case with $\phi : (h, y) \mapsto Ah + By$ (where y stands for its one-hot encoding e_y, as in the Transformer case), which is linear in both arguments. Since C is independent of the input y_t, the emission kernel $p(y_{t+1} \mid h_t) = \mathrm{softmax}(W_{\mathrm{out}}Ch_t)_{y_{t+1}}$ depends on h_t alone. As in the Transformer and RNN cases, y_t is a discrete token while $h_t \in \mathbb{R}^d$ is continuous.

Mamba (selective SSM). Mamba [49] introduces input-dependent ("selective") parameters: $A(y_t)$, $B(y_t)$, and $C(y_t)$ are all functions of the current input y_t. While the state update $h_t = A(y_t)\, h_{t-1} + B(y_t)\, y_t$ remains a deterministic function of (h_{t-1}, y_t), the output projection involves $C(y_t)\, h_t$, so that the emission kernel depends on y_t as well as h_t. Since y_t cannot in general be recovered from h_t alone, the SSM state h_t by itself is not a sufficient statistic of the past for predicting y_{t+1}.

To accommodate Mamba within the recursive framework, it suffices to augment the latent state as $h'_t \equiv (h_t,\ y_t)$. The augmented update is $h'_t = \phi'(h'_{t-1}, y_t) = (A(y_t)\, h_{t-1} + B(y_t)\, y_t,\ y_t)$. This is a deterministic function of (h'_{t-1}, y_t) and preserves the recursive structure. As in the SSM case, y_t is discrete and h_t is continuous, so the augmented state $h'_t = (h_t,\ y_t)$ comprises both continuous and discrete components.

III. BACKWARD PROCESS AND ENTROPY PRODUCTION

In this section, we construct the backward process by reusing the same architectural components in reversed temporal order and define the entropy production as the KL divergence between the forward and backward path measures.

A. Backward process

The backward process is defined by reusing the same emission kernels p_t and deterministic maps Φ_t in the backward direction to produce a sequence in the reversed temporal order. Concretely, we fix an initial latent state $\tilde{h}_0$ for the backward process and generate a sequence $(\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_T)$ according to the following:

(1) $\tilde{h}_s$ is updated deterministically: $\tilde{h}_s = \tilde{\Phi}_s(\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_s)$ for $s = 1, 2, \ldots, T$, and $\tilde{h}_0 = \tilde{\Phi}_0(\,)$ is set to a constant (initial condition).

(2) $\tilde{y}_{s+1}$ is drawn from the conditional distribution $\tilde{p}_s(\tilde{y}_{s+1} \mid \tilde{h}_s)$ for $s = 1, \ldots, T-1$, and $\tilde{p}_0(\tilde{y}_1 \mid \tilde{h}_0) \equiv \tilde{p}(\tilde{y}_1)$ is the initial distribution of the backward process.

The backward path probability is then given by

$$P_{\leftarrow}(\tilde{y}_{1:T}) = \prod_{s=0}^{T-1} \tilde{p}_s(\tilde{y}_{s+1} \mid \tilde{\Phi}_s(\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_s)). \tag{6}$$

We now relate these notations to those in the forward process. We interpret this backward process as generating the forward-time sequence in reversed order. That is, we identify

$$\tilde{y}_s = y_{T-s+1}. \tag{7}$$

We also set

$$\begin{aligned} \tilde{p}_s &= p_{T-s} \ (s = 1, \ldots, T-1), \\ \tilde{\Phi}_s &= \Phi_{T-s+1} \ (s = 1, \ldots, T), \end{aligned} \tag{8}$$

which is consistent with the standard notion of backward protocols in stochastic thermodynamics (as explicitly shown later). We note that we assume that y and h do not involve any parity-odd quantity that changes sign under time reversal (such as the momentum of the underdamped Langevin equation).

As mentioned above, the initial distribution of the backward process is given by

$$\tilde{p}(\tilde{y}_1) \equiv \tilde{p}_0(\tilde{y}_1 \mid \tilde{h}_0), \tag{9}$$

where $\tilde{h}_0$ is a fixed constant. In our framework, it is natural to assume that $\tilde{p}_0$ is a known function chosen independently of $p(y_T)$ in the forward process. On the other hand, another natural choice in stochastic thermodynamics is

$$\tilde{p}(\tilde{y}_1) = p(y_T), \tag{10}$$

which means that the initial distribution of the backward process is identical to the final distribution of the forward process. We do *not* assume Eq. (10) in this paper unless otherwise stated, because it is operationally intractable in our framework in general. We emphasize that, also in the standard formulation of stochastic thermodynamics, Eq. (10) is not always assumed (e.g., Ref. [56]), and even in such cases the entropy production still has a clear physical meaning, especially when the initial distribution of the backward process is chosen to be the Gibbs distribution with respect to a fixed Hamiltonian. This point will be discussed in Sec. IV B in more detail, and a concrete example will be illustrated in Sec. VI.

Since each Φ_t accepts variable-length input, the map Φ_{T-s+1} can be applied to the s-element backward sequence $(\tilde{y}_1, \ldots, \tilde{y}_s)$, using the parameters (e.g., attention weights) indexed by $T-s+1$. The backward process thus "runs the same machinery in reverse": The same operators Φ_t and kernels p_t are reused, but they are invoked in the order $t = T, T-1, \ldots, 1, 0$, and their input is the backward sequence accumulated so far. Note that $\Phi_{T+1}(\,) = \tilde{\Phi}_0(\,)$ has no counterpart in the forward process, but it just gives a chosen constant $\tilde{h}_0$.

Then,

$$P_{\leftarrow}(y_{T:1}) = \prod_{s=0}^{T-1} p_{T-s}(y_{T-s} \mid \Phi_{T-s+1}(y_T, y_{T-1}, \ldots, y_{T-s+1})). \tag{11}$$

Here, p_T for $s = 0$ is introduced solely to denote the initial distribution of the backward process: $p_T(y_T) \equiv \tilde{p}(y_T) \equiv \tilde{p}_0(y_T \mid \tilde{h}_0)$. We reindex by $t = T - s$ (i.e., $s = T - t$) to express this in forward-time indices and introduce the notation

$$g^{\leftarrow}_{t+1}(y_{T:t+1}) \equiv \Phi_{t+1}(y_T, y_{T-1}, \ldots, y_{t+1}) = \tilde{h}_{T-t}. \tag{12}$$

We finally obtain

$$P_{\leftarrow}(y_{T:1}) = \prod_{t=1}^{T} p_t(y_t \mid g^{\leftarrow}_{t+1}(y_{T:t+1})). \tag{13}$$

See Fig. 1(b) for a graphical representation.

We emphasize that we cannot assume $\tilde{h}_s = h_{T-s+1}$ even for the event that the time-reversed sequence $\tilde{y}_s = y_{T-s+1}$ is realized. This is because the reverse run of y_t does not necessarily produce the reverse run of h_t by applying the given deterministic function Φ_t in the reverse way, even for the recursive case.

It is also important to distinguish the backward process introduced here from Bayesian retrodiction $P_{\rightarrow}(y_t \mid y_{t+1:T})$ (see Sec. VII); rather, in direct analogy with Crooks-type stochastic thermodynamics, the backward process is obtained by reversing the protocol implemented by p_t and Φ_t.

B. Entropy production

We now define the entropy production of the observed sequence as the KL divergence between the forward and backward path measures:

$$\mathcal{S}_y \equiv D_{\mathrm{KL}}(P_{\rightarrow}(y_{1:T}) \| P_{\leftarrow}(y_{T:1})) = \mathbb{E}_{P_{\rightarrow}}\left[\ln \frac{P_{\rightarrow}(y_{1:T})}{P_{\leftarrow}(y_{T:1})}\right] \geqslant 0. \tag{14}$$

Non-negativity follows from the general property of KL divergence. This can be equivalently represented as

$$\begin{aligned} \mathcal{S}_y &= \mathbb{E}_{P_{\rightarrow}}\left[\ln \frac{\prod_{t=0}^{T-1} p_t(y_{t+1} \mid f^{\rightarrow}_t(y_{1:t}))}{\prod_{t=1}^{T} p_t(y_t \mid g^{\leftarrow}_{t+1}(y_{T:t+1}))}\right] \\ &= \mathbb{E}_{P_{\rightarrow}}[-\ln \tilde{p}(y_T) + \ln p(y_1)] \\ &\quad + \sum_{t=1}^{T-1} \mathbb{E}_{P_{\rightarrow}}\left[\ln \frac{p_t(y_{t+1} \mid f^{\rightarrow}_t(y_{1:t}))}{p_t(y_t \mid g^{\leftarrow}_{t+1}(y_{T:t+1}))}\right]. \end{aligned} \tag{15}$$

If we assume Eq. (10), the first term becomes

$$\begin{aligned} \mathbb{E}_{P_{\rightarrow}}[-\ln p(y_T) + \ln p(y_1)] &= -\sum_{y_T} p(y_T) \ln p(y_T) \\ &\quad + \sum_{y_1} p(y_1) \ln p(y_1), \end{aligned} \tag{16}$$

which is the change in the Shannon entropy of the single-time marginal distribution in the forward process.

Correspondingly, the stochastic entropy production is defined as

$$\sigma(y_{1:T}) \equiv \ln \frac{P_{\rightarrow}(y_{1:T})}{P_{\leftarrow}(y_{T:1})}, \tag{17}$$

which gives

$$\mathcal{S}_y = \mathbb{E}_{P_{\rightarrow}}[\sigma(y_{1:T})]. \tag{18}$$

The integral fluctuation theorem [3–5] is automatically satisfied:

$$\mathbb{E}_{P_{\rightarrow}}[e^{-\sigma(y_{1:T})}] = 1. \tag{19}$$

Here, rigorously speaking, the integral fluctuation theorem requires that $P_{\rightarrow}(y_{1:T}) > 0$ whenever $P_{\leftarrow}(y_{T:1}) > 0$. This condition is satisfied in all examples of Table I: For discrete-token models (Transformers, RNNs, SSMs, and Mamba), the softmax emission kernel assigns strictly positive probability to every token, while for the linear Gaussian case the forward and backward path measures share the same support by construction.

We note that $\mathcal{S}_y$ depends not only on the forward path measure $P_{\rightarrow}(y_{1:T})$ but also on the specific decomposition into emission kernels p_t and deterministic maps Φ_t, since the backward process (13) reuses these architectural components in reversed temporal order. This parallels a situation in stochastic thermodynamics for non-Markovian processes: The entropy production based on Crooks-type protocol reversal depends on the specific underlying dynamical model, such as the memory kernel and bath structure in generalized Langevin systems [13,14]. In the Markovian limit, the backward transition kernel $P_{\leftarrow}(y_t \mid y_{t+1})$ is naturally determined by the forward process, consistent with the reduction to the standard Crooks formulation [4] as shown below.

C. Recursive case

In the recursive special case of Sec. II B, we choose the backward model so that its latent state is updated recursively by

$$\tilde{h}_s = \phi(\tilde{h}_{s-1}, \tilde{y}_s). \tag{20}$$

The graphical representation of the dependency structure is shown in Fig. 2(b). We also assume that $h_0 = \tilde{h}_0$.

As mentioned before, the reverse run of y_t does not necessarily produce the reverse run of h_t by applying the given deterministic function ϕ. To explicitly see this, the following small example suffices: Let $T = 2$, $\phi(h, y) \equiv 2h + 3y$, and $h_0 = \tilde{h}_0 = 0$. Then, $h_1 = 3y_1$ and $h_2 = \phi(h_1, y_2) = \phi(3y_1, y_2) = 6y_1 + 3y_2$. In the reverse run, $\tilde{h}_1 = 3y_2$ and $\tilde{h}_2 = \phi(\tilde{h}_1, y_1) = \phi(3y_2, y_1) = 3y_1 + 6y_2$. Obviously $h_1 \neq \tilde{h}_2$ and $h_2 \neq \tilde{h}_1$.

This mismatch has an important implication when one considers the Markovian embedding $\mathbf{x}_t \equiv (h_t, y_t)$, which is available in the recursive case (see Appendix A for details). From the above consideration, the forward and backward transition kernels of $\mathbf{x}_t$ have disjoint support in general, so that the entropy production $\mathcal{S}_{\mathbf{x}}$ defined at the $\mathbf{x}$ level may diverge and would not be informative. Even so, the entropy production of y itself, $\mathcal{S}_y$, stays finite, as explicitly shown in Sec. VI for a concrete example. This provides an additional motivation for our approach, which defines the entropy production $\mathcal{S}_y$ solely through the path probabilities of the observed sequence $y_{1:T}$, without relying on a Markovian embedding.

Markovian case. As a further specific subclass of the recursive class, we remark on the case where y_t itself is Markovian. Indeed, the standard formulation of stochastic thermodynamics of Markovian dynamics can be regarded as a special case of our formulation, where we can take $\Phi(y_1, \ldots, y_t) = y_t$ so that $h_t = y_t$ for all t. Then, automatically $g^{\leftarrow}_{t+1}(y_{T:t+1}) = \Phi(y_T, y_{T-1}, \ldots, y_{t+1}) = y_{t+1}$. Equivalently, $\phi(h, y) = y$ gives $h_t = y_t$ and $\tilde{h}_s = \tilde{y}_s$. Therefore, the second term of Eq. (15) reduces to

$$\sum_{t=1}^{T-1} \mathbb{E}_{P_{\rightarrow}}\left[\ln \frac{p_t(y_{t+1} \mid y_t)}{p_t(y_t \mid y_{t+1})}\right], \tag{21}$$

which is exactly equivalent to the standard definition in Markovian stochastic thermodynamics *à la* Crooks [4]. Here, it is important that the forward transition $y_t \to y_{t+1}$ and the backward transition $y_{t+1} \to y_t$ are induced by the same kernel p_t with index t. If we further assume the local detailed balance condition, this term is regarded as $-\beta Q$, with β being the inverse temperature and Q being the heat absorbed from a thermal reservoir. We emphasize that we assume neither Markovianity nor detailed balance in our general framework.

D. Discussion

Let us clarify the relation of our definition of the entropy production (14) to existing notions. A line of work directly related to the present approach quantifies irreversibility on the basis of the KL divergence between the forward and backward processes [56]. Seminal works [7,8] adopted this approach for the observed non-Markovian process; see also the stationary Gaussian analysis of Ref. [12]. Closely related observed-path constructions arise from marginal or coarse-grained path measures, including lower bounds from coarse-grained trajectories [6,9–11]. Another approach treats non-Markovian physical dynamics by explicitly assuming the presence of thermal reservoirs, especially generalized Langevin systems with colored baths [13,14].

In contrast, in our deterministic-memory autoregressive setting, each factor of the forward/backward path-probability ratio is supplied directly by the model's emission kernel together with the deterministic-memory update. As a consequence, $\mathcal{S}_y$ is directly computable from the model itself without empirical estimation of long-history conditionals, and without introducing an auxiliary stochastic hidden-state dynamics.

Because our framework does not presuppose any thermal reservoir, we impose no local detailed balance condition. In particular, when the framework is applied to an LLM, $\mathcal{S}_y$ does not directly quantify the physical energy cost or heat dissipation of the hardware that executes the model. If one regards the full physical state of the computing device as a Markovian process x_t that generates the token sequence y_t (see Appendix A and assumptions there), the coarse-graining inequality $\mathcal{S}_x \geqslant \mathcal{S}_y$ implies that $\mathcal{S}_y$ may serve as a lower bound on the physical entropy production, but the two quantities are in general distinct. For a concrete physical implementation, the actual dissipation depends on hardware-specific details and is therefore not determined by $\mathcal{S}_y$ alone. Quantifying the tightness and operational significance of this bound for specific hardware architectures remains an open problem. Constructing a simplified physical model that explicitly implements the autoregressive update and comparing $\mathcal{S}_x$ and $\mathcal{S}_y$ would be a worthwhile direction for future work.

In all architectures listed in Table I, the only externally observable variable is y_t. The latent state $h_t = \Phi_t(y_1, \ldots, y_t)$ is a deterministic function of the observed history and

carries no independent stochastic degrees of freedom. On the other hand, behind the observations y_t there may exist a true environmental state x_t whose dynamics generates y_t. The Kalman filter example (Sec. VI) makes this explicit: x_t evolves as $x_{t+1} = Ax_t + w_t$ and produces observations $y_t = Cx_t + v_t$, but x_t has no counterpart in the framework. In the case of an LLM, there are at least two fundamentally different candidates for such a state x_t: the physical internal state of the hardware executing the model, and the state of the real-world process that the text describes.

Meanwhile, the asymmetry between forward and backward modeling of natural language has been studied from machine-learning perspectives. Reference [57] quantified the irreversibility of language models by separately training the backward model on the time-reversed dataset of natural language, which is distinct from our definition of the backward process for which the same model as the forward process is used. Reference [58] computed the per-trajectory difference between forward and backward losses under a single model, but the model itself is trained on both directions. Note that neither work formulated a connection to stochastic thermodynamics.

We also remark on terminology. Reference [59] introduced a quantity called the entropy production rate for LLMs, defined as the average Shannon entropy of per-token predictive distributions; this involves neither time reversal nor a latent-state architecture, and is conceptually distinct from our entropy production $\mathcal{S}_y$ (14).

IV. ESTIMATION OF ENTROPY PRODUCTION

The entropy production $\mathcal{S}_y$ defined in Eq. (14) involves the log-ratio of the forward and backward path probabilities summed over all time steps. A central question for practical applications—particularly for LLMs and other large-scale autoregressive models—is whether $\mathcal{S}_y$ can be estimated from a finite number of sampled trajectories. In this section, we show that the structure of the autoregressive framework (Sec. II A) makes $\mathcal{S}_y$ itself efficiently computable by standard Monte Carlo sampling. This sampling method will be applied to GPT-2 in Sec. V A as a proof-of-concept demonstration with a pretrained language model and will be demonstrated in Sec. VI E for the linear Gaussian case. We further show that temporal coarse-graining, in which the backward pass reverses the order of blocks rather than individual tokens, can be evaluated at the same per-trajectory cost (Sec. IV D).

A. Sampling cost for general non-Markovian processes

To clarify why the present framework is special, it is instructive to first consider the generic situation. Suppose one observes a non-Markovian stochastic process y_t from an unknown source (e.g., a physical experiment) and wishes to estimate the entropy production of the form $\mathbb{E}_{P_\rightarrow}[\sum_t \ln P_\rightarrow(y_{t+1} \mid y_{1:t}) - \ln P_\leftarrow(y_t \mid y_{T:t+1})]$. The conditional probability $P_\rightarrow(y_{t+1} \mid y_{1:t})$ is then an unknown function of the entire past history. Estimating it from data requires observing many trajectories that share the same prefix $y_{1:t}$; in a continuous or large observation space, the number of such coincidences is negligible, and the required sample size grows combinatorially with the length of the conditioning history. No compression is available unless one makes additional modeling assumptions.

B. Tractability within the autoregressive framework

The framework of Sec. II A circumvents this difficulty through the following structural features that make the path probabilities directly evaluable.

Deterministic latent state. The latent state $h_t = \Phi_t(y_1, \ldots, y_t)$ is a deterministic function of the observed history. Given a single trajectory $y_{1:T}$, every latent state $h_0, h_1, \ldots, h_T$ is uniquely determined without any stochastic marginalization. This is in sharp contrast with models with stochastic latent states, such as hidden Markov models, where $P(y_{t+1} \mid y_{1:t}) = \sum_{x_{t+1}} P(y_{t+1} \mid x_{t+1}) P(x_{t+1} \mid y_{1:t})$ involves a sum over latent states x_{t+1} rather than a direct evaluation from a deterministic state.

Explicit emission kernel. The conditional distribution $p_t(y_{t+1} \mid h_t)$ is an explicit, evaluable function provided by the model. For a Transformer-based LLM, this is the softmax output:

$$p_t(y_{t+1} \mid h_t) = \frac{\exp\big((W_{\mathrm{out}}\, h_t)_{y_{t+1}}/\tau\big)}{\sum_y \exp\big((W_{\mathrm{out}}\, h_t)_y/\tau\big)}, \tag{22}$$

where $\tau > 0$ is the temperature parameter (usually set to 1). No additional sampling or density estimation is needed to obtain $\ln p_t(y_{t+1} \mid h_t)$.

Boundary distributions. The initial distributions of the forward and backward processes are also explicit. In the forward process, the initial distribution is given by $p(y_1) \equiv p_0(y_1 \mid h_0)$, which is a known function of y_1 if p_0 and h_0 are given. In the backward process, the initial distribution is given by Eq. (9), i.e., $\tilde{p}(\tilde{y}_1) \equiv \tilde{p}_0(\tilde{y}_1 \mid \tilde{h}_0)$, which is also a known function of $\tilde{y}_1$ if $\tilde{p}_0$ and $\tilde{h}_0$ are given. This is precisely the strategy adopted in the linear Gaussian example of Sec. VI, where $\tilde{p}(\tilde{y}_1) = \mathcal{N}(C\hat{x}_{1|0},\, S)$. These choices make the boundary terms directly evaluable for any sampled trajectory. On the other hand, if one adopts Eq. (10), i.e., $\tilde{p}(\tilde{y}_1) = p(y_T)$, then evaluating this term requires computing the marginal distribution $p(y_T) = \sum_{y_1,\ldots,y_{T-1}} P_\rightarrow(y_{1:T})$, which involves a summation (or integration in the continuous case) over the entire set of earlier observations and is in general intractable.

C. Cost estimates for specific architectures

Using the product decompositions (15), the stochastic entropy production is written as

$$\sigma(y_{1:T}) = \sum_{t=0}^{T-1} \underbrace{\ln p_t(y_{t+1} \mid \overrightarrow{f_t}(y_{1:t}))}_{\text{forward log-likelihood}} - \sum_{t=1}^{T} \underbrace{\ln p_t(y_t \mid \overleftarrow{g_{t+1}}(y_{T:t+1}))}_{\text{backward log-likelihood}}. \tag{23}$$

For a single sampled trajectory $y_{1:T} \sim P_\rightarrow$, each term in the sum is obtained as follows.

(1) *Forward pass*: Feed $y_1, y_2, \ldots, y_T$ sequentially into the model. At each step t, the model computes $h_t = f_t^{\rightarrow}(y_{1:t})$ and outputs $\ln p_t(y_{t+1} \mid h_t)$.

(2) *Backward pass (evaluation only)*: Feed the reversed sequence $y_T, y_{T-1}, \ldots, y_1$ into the same model (with the same emission kernels p_t and maps Φ_t invoked in reverse temporal order, as specified in Sec. III A). At backward step s, the model processes $y_T, \ldots, y_{T-s+1}$, computes $\tilde{h}_s$, and outputs $\ln p_{T-s}(y_{T-s} \mid \tilde{h}_s)$. Note that sampling from the backward process is not necessary to evaluate (23).

Each of the two terms in Eq. (23) is computationally identical to a single log-likelihood evaluation: Given a sequence $(y_1, \ldots, y_T)$, compute the sum $\sum_t \ln p_t(y_{t+1} \mid h_t)$ by feeding the tokens through the model. This is one of the most basic operations in autoregressive modeling; it is in fact performed during training (where the negative log-likelihood serves as the loss function). Note that $\sigma(y_{1:T})$ evaluated on a single trajectory $y_{1:T}$, without taking the sample average (24) below, is itself the stochastic entropy production for that particular realization. It therefore provides trajectory-level information about the irreversibility of the process.

The entropy production is then estimated by Monte Carlo averaging:

$$\mathcal{S}_y = \mathbb{E}_{P_{\rightarrow}}[\sigma] \approx \frac{1}{N}\sum_{n=1}^{N} \sigma\left(y_{1:T}^{(n)}\right). \tag{24}$$

For a target accuracy ϵ, the required number of sample trajectories scales as $N = O(\mathrm{Var}(\sigma)/\epsilon^2)$. Thus, the estimator has the standard $N^{-1/2}$ Monte Carlo convergence, with no additional combinatorial overhead from enumerating histories. In general, $\mathrm{Var}(\sigma)$ itself may depend on the sequence length T and on the statistics of the process. In Appendix B 2, we numerically demonstrate convergence of the estimator (24) for $T = 120$ and $N \simeq 500$.

The total computational cost of the Monte Carlo estimator (24) is written as

$$C_{\mathrm{total}} = NC_1, \tag{25}$$

where C_1 is the cost of evaluating σ on a single trajectory. Here and below, "cost" refers to the number of floating-point operations, the standard hardware-independent measure of arithmetic complexity for numerical computations. Since σ requires one forward and one backward pass, and each pass is a log-likelihood evaluation, the per-trajectory cost is

$$C_1 = 2C_{\mathrm{LL}}, \tag{26}$$

where C_{LL} denotes the cost of a single log-likelihood evaluation for the architecture in question.

Roughly speaking, the cost C_{LL} of a single log-likelihood evaluation scales at most as $O(T^2)$ for Transformers [38] and as $O(T)$ for recursive architectures such as RNNs [42,43] and state space models and Mamba [47–49]; in particular, no combinatorial overhead from enumerating non-Markovian histories enters either factor in Eq. (25), the only T-dependence of N being the Monte Carlo factor $\mathrm{Var}(\sigma)$.

D. Temporal coarse-graining

When our framework is applied to language models, the backward process (13) evaluates the path probability of the token sequence in reversed order. For instance, given the forward sequence $y_{1:4} = (\texttt{This}, \texttt{is}, \texttt{a}, \texttt{book})$, the backward path probability is that of $(\texttt{book}, \texttt{a}, \texttt{is}, \texttt{This})$, which is extremely small under a model trained on natural language. The entropy production is then dominated by this artifact of token-level reversal, rather than by physically or semantically meaningful irreversibility that may reflect the structure of the real-world processes described by the text.

A natural approach to this issue is temporal coarse-graining: reversing the order of *blocks* of tokens rather than individual tokens. In this subsection, we restrict to the time-homogeneous case

$$p_t = p, \quad \Phi_t = \Phi \quad \text{for all } t, \tag{27}$$

with a common initial latent state $\tilde{h}_0 = h_0$. This is the setting directly relevant for typical pretrained LLMs, in which a single set of model parameters is applied at every time step. Under Eq. (27), the forward path probability (3) becomes

$$P(y_{1:T}) = \prod_{t=0}^{T-1} p(y_{t+1} \mid \Phi(y_{1:t})), \tag{28}$$

where $\Phi(\,) \equiv h_0$.

To define coarse-grained reversal, we group consecutive tokens into blocks

$$y'_{t'} \equiv (y_{(t'-1)l+1}, \ldots, y_{t'l}) \tag{29}$$

of length l (with $T = \tilde{T}l$) and define the *block-reversed token sequence*

$$\tilde{y}'_{1:\tilde{T}} \equiv \Bigg(\underbrace{y_{(\tilde{T}-1)l+1}, \ldots, y_{\tilde{T}l}}_{y'_{\tilde{T}}}, \underbrace{y_{(\tilde{T}-2)l+1}, \ldots, y_{(\tilde{T}-1)l}}_{y'_{\tilde{T}-1}}, \ldots, \underbrace{y_1, \ldots, y_l}_{y'_1} \Bigg), \tag{30}$$

which concatenates blocks $y'_{\tilde{T}}, y'_{\tilde{T}-1}, \ldots, y'_1$ in reversed block order while preserving the token order within each block. For example, with $T = 6$, $l = 3$, and $y_{1:6} = (a, b, c, d, e, f)$, the token-level reversal is (f, e, d, c, b, a), while the block-level reversal is $\tilde{y}'_{1:2} = (d, e, f, a, b, c)$.

If one chooses the blocks at the level of sentences or "episodes" (contiguous segments forming semantically coherent units), the backward sequence reverses the order of episodes but generates the tokens within each episode in the forward order. The intraepisode conditional probabilities would remain those of natural language, and thus the entropy production would be governed by interepisode retrodiction, which may carry a more interpretable signal. This expectation is consistent with the GPT-2 demonstration in Sec. V.

Since the model is time homogeneous, the coarse-grained backward path probability is simply the path probability of the block-reversed sequence evaluated by the same model:

$$P'_{\leftarrow}(y'_{\tilde{T}:1}) \equiv P(\tilde{y}'_{1:\tilde{T}}). \tag{31}$$

The coarse-grained stochastic entropy production for a single trajectory $y_{1:T} \sim P_{\rightarrow}$ is then

$$\sigma'(y_{1:T}) \equiv \ln P(y_{1:T}) - \ln P(\tilde{y}'_{1:\tilde{T}}), \tag{32}$$

which is the difference in log-likelihood between the original text and its block-reversed version, both evaluated by the same model in a single forward pass each.

Since the block-reversal map is a bijection on $\mathcal{Y}^T$, $P'_{\leftarrow}$ is a normalized distribution on $y_{1:T}$. The coarse-grained entropy production

$$\mathcal{S}'_y \equiv \mathbb{E}_{P_{\rightarrow}}[\sigma'] = D_{\mathrm{KL}}(P_{\rightarrow}(y_{1:T}) \| P'_{\leftarrow}(y'_{\tilde{T}:1})) \geqslant 0 \tag{33}$$

is non-negative as a KL divergence, and the integral fluctuation theorem $\mathbb{E}_{P_{\rightarrow}}[e^{-\sigma'}] = 1$ follows directly from the normalization of $P'_{\leftarrow}$. The Monte Carlo estimator takes the same form as (24): $\mathcal{S}'_y \approx \frac{1}{N}\sum_{n=1}^{N} \sigma'(y^{(n)}_{1:T})$.

Concretely, $\sigma'(y_{1:T})$ is evaluated as follows. Given a single sampled trajectory $y_{1:T}$, the forward log-likelihood $\ln P(y_{1:T})$ is obtained by the standard forward pass: Feed $y_1, y_2, \ldots, y_T$ sequentially into the model, compute the latent states $h_t = \Phi(y_{1:t})$ at each step, and accumulate $\sum_t \ln p(y_{t+1} \mid h_t)$. For the second term, one constructs the block-reversed token sequence $\tilde{y}'_{1:\tilde{T}}$ (30) and feeds it into the *same* model as if it were an ordinary input sequence: The model computes new latent states $\tilde{h}_1, \tilde{h}_2, \ldots$ by applying Φ to successive prefixes of $\tilde{y}'_{1:\tilde{T}}$, and one accumulates the log-likelihood of each token in $\tilde{y}'_{1:\tilde{T}}$ conditioned on the corresponding $\tilde{h}$, yielding $\ln P(\tilde{y}'_{1:\tilde{T}})$. No sampling from the backward process is required; the entire computation consists of two forward passes of the model (one on the original sequence and one on its block-reversed version), so the per-trajectory cost is $C_1 = 2C_{\mathrm{LL}}$, the same as for the token-level σ (26). Setting $l = 1$ reduces the block-reversed sequence to the fully reversed sequence, and (31) recovers the token-level backward path probability (13).

The construction extends to variable-length blocks. Let S be a segmentation rule that partitions every sequence $y_{1:T}$ into consecutive blocks $B_1, \ldots, B_k$ of possibly unequal lengths summing to T. Given a forward sample $y_{1:T} \sim P$, one applies S to obtain the blocks, concatenates them in reversed order $R_S(y_{1:T}) \equiv B_k B_{k-1} \cdots B_1$, and evaluates the log-likelihood of $R_S(y_{1:T})$ under the same model; the stochastic entropy production is again $\sigma'(y_{1:T}) = \ln P(y_{1:T}) - \ln P(R_S(y_{1:T}))$, as in Eq. (32). For $P(R_S(y_{1:T}))$ to be a normalized distribution [and hence for Eq. (33) to remain a valid KL divergence], R_S must be a bijection. A natural strategy is to segment at every occurrence of a distinguished delimiter token (e.g., a sentence-final period or an end-of-sentence marker) and to require that y_T itself be such a delimiter. In this case, the segmentation of the reversed sequence is uniquely determined, and thus R_S is a bijection on the restricted space conditioned on this terminal constraint. We then replace P with the conditional distribution P_{c}, which leaves σ' unchanged. The fluctuation theorem for the coarse-grained entropy production is then satisfied:

$$\mathbb{E}_{P_{\mathrm{c}}}[e^{-\sigma'(y_{1:T})}] = 1. \tag{34}$$

V. PROOF-OF-CONCEPT EXPERIMENT WITH GPT-2

As a demonstration of the estimation method formulated in Sec. IV, we evaluate the stochastic entropy production for a pretrained Transformer-based language model, GPT-2 (124M parameters) [39]. Since GPT-2 uses time-independent parameters, it satisfies the time-homogeneous condition (27), and the path probability takes the form (28). For each text $y_{1:T}$, we compute two quantities: The first is the token-level stochastic entropy production, here written as σ_{token}, and the second is the block-level stochastic entropy production with variable-length blocks (32), written as σ_{block}. The segmentation and reversal are both performed at the level of the token-ID sequence.

In the following, we evaluate these quantities for the trajectory probabilities of GPT-2, using sampling from GPT-2 itself (Sec. V A) and using fixed (externally prepared) text sets (Sec. V B).

A. Monte Carlo sampling from GPT-2

We generate sequences of $T = 120$ tokens from GPT-2 (see Appendix B 1 for details). Sampling is carried out by an explicit autoregressive loop that draws one token at a time using the model's KV cache, rather than by the library's default `model.generate()` method; this excludes any post-processing that deforms distributions, and ensures that exactly T tokens are always produced. The forward log-likelihood can be accumulated from the same logits used for sampling.

The path probability $P(y_{1:T})$ in Eq. (28) includes the $t = 0$ factor $p(y_1 \mid h_0)$, where $h_0 = \Phi(\,)$ is the initial latent state. We compute this factor by conditioning on GPT-2's special `<|endoftext|>` token as the initial token. Intuitively, this choice of the initial token means that any user-specified prompt is absent. The same initial token is used for both the original and reversed sequences, so that the common initial latent state $\tilde{h}_0 = h_0$ is satisfied.

For the token-level entropy production σ_{token}, we use each generated sequence of length T. For the block-level entropy production σ_{block}, we truncate each sequence at the last sentence-final punctuation token so that $y_{T'}$ is a delimiter, as required by the bijection condition (Sec. IV D), where $T' \leqslant T$ denotes the truncated length. Sequences containing no sentence-final punctuation are excluded from the block-level analysis but retained for the token-level one. For each sample, we compute the token-level per-token entropy production $\sigma_{\mathrm{token}}/T$ using the full sequence, and the block-level $\sigma_{\mathrm{block}}/T'$ using the truncated sequence. Additionally, for the sake of comparison, we compute the token-level entropy production $\sigma_{\mathrm{token}}(T') \equiv \sigma_{\mathrm{token}}(y_{1:T'})$ on the same truncated sequence used for the block-level analysis.

Figure 3 shows the resulting distributions of the stochastic entropy production. The token-level values [panel (a)] are concentrated well above zero, confirming the large irreversibility of token-order reversal: The token-level entropy production is dominated by the artifact of syntactic destruction—a reversed token sequence such as "`book a is This`" receives extremely low probability under GPT-2. This dominance of the syntactic artifact in σ_{token} motivates the block-level coarse-graining, which has no counterpart in previous studies of forward-backward asymmetry in LLMs [57,58]. In fact, the block-level values shown in Fig. 3(b) are much smaller, which is consistent with the discussion in Sec. IV D. Note that the reference distribution $\sigma_{\mathrm{token}}(T')/T'$,

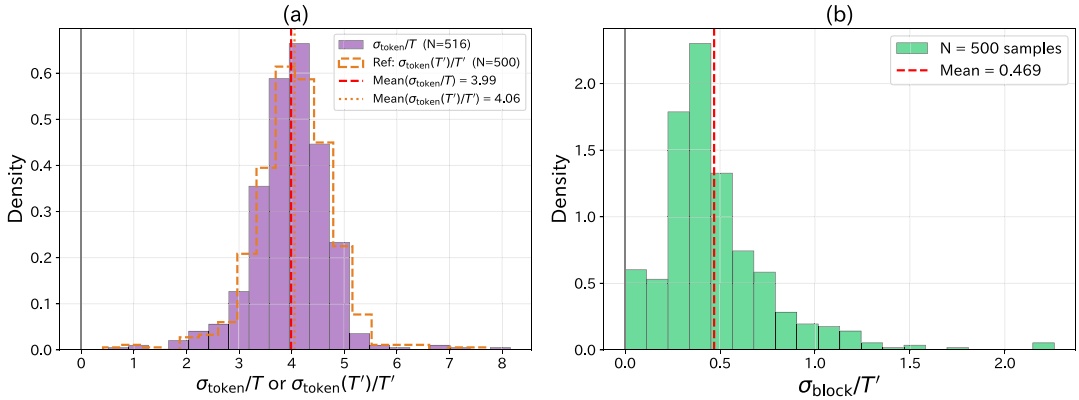


FIG. 3. Distribution of the per-token stochastic entropy production for sequences of $T = 120$ tokens sampled from GPT-2 (no top-k or nucleus truncation). The temperature parameter is $\tau = 1$. (a) Token-level reversal σ_{token}/T, computed on full generated sequences (filled purple); the dashed orange step histogram shows the reference $\sigma_{\text{token}}(T')/T'$, i.e., token-level reversal applied to the truncated sequences of length T'. (b) Block-level reversal σ_{block}/T', where each sequence is *post hoc* truncated at the last sentence-final punctuation token (length $T' \leqslant T$). Dashed red lines indicate the sample means; the dotted orange line in panel (a) indicates the mean of the reference distribution. We collect samples until $N = 500$ of them satisfy the bijection condition for block reversal [panel (b)]. Samples that fail this condition are excluded from the block-level analysis but retained for the token-level one, so the token-reversal count in panel (a) is slightly larger (namely, 516). Note the different horizontal scales between panels (a) and (b).

overlaid in panel (a), has almost the same mean as that of σ_{token}/T, indicating that the much smaller block-level values in panel (b) are mainly due to the coarser reversal rather than the truncation from T to T'.

We also numerically observe that, for both token reversal and block reversal, the Monte Carlo estimate of the entropy production converges to a positive value by around $N = 500$ (see Appendix B 2 for supplemental numerical results, including a verification of the integral fluctuation theorem for token reversal, as a consistency check).

The token-level Monte Carlo sampling without truncation [the filled purple in Fig. 3(a)] is taken over the true distribution $P(y_{1:T})$ of GPT-2, which exactly fits our general theoretical description. For the block-level reversal, however, the estimated quantity deviates from the true distribution in two respects. First, sequences lacking any sentence-final punctuation are excluded; this exclusion rate is about 3% in our experiment. Second, each retained sequence is truncated at the position of its last delimiter. Denoting the truncation point $T'(y_{1:T})$, the block-level Monte Carlo estimator converges to $\mathbb{E}_{y_{1:T}\sim P'}[\sigma_{\text{block}}(y_{1:T'})/T'(y_{1:T})]$, where P' is the conditional distribution under the abovementioned exclusion. Therefore, rigorously speaking, our Monte Carlo estimate for the block-reversal case deviates from the ideal theoretical setting described in Sec. IV D. However, we expect this deviation to be small, given that the analogous deviation for token-level reversal is minor [compare the filled purple and dashed orange distributions in Fig. 3(a)].

In addition, the dependence of the coarse-grained entropy production on the block scale, for both fixed-length blocks and k-sentence superblocks, is examined in Appendix B 3 (Fig. 9).

B. Evaluation of externally prepared texts

An important challenge is to clarify the meaning and implications of the entropy production at both the token-reversal and block-reversal levels, when applied to real language models. As a first step in this direction, we present an experiment in which we evaluate the stochastic entropy production of GPT-2 on externally prepared texts, rather than on sequences generated by GPT-2 itself. This is still meaningful as a probe of GPT-2, because the stochastic entropy production quantifies the irreversibility of a *single* realized trajectory. A practical difficulty, however, is how to prepare test texts systematically while avoiding arbitrary choices by the experimenter.

To address this issue, we use multiple input-text generators and examine whether the observed causal/noncausal difference depends on the model used to prepare the texts. Specifically, we submit the same fixed prompt to five language models: Claude Opus 4.6, Claude Fable 5, Gemini 3.1 Pro, GPT-5.4 Pro, and GPT-5.6 Pro. For each generator, we prepare the following two sets of English-language texts, each containing 100 four-sentence samples:

(1) *Causal texts*: Short narratives in which the sentences describe a temporally ordered chain of events (e.g., "The glass slipped from Alice's hand. The glass fell to the floor. The glass broke into many pieces. The glass left a wet stain on the tile floor"). Reversing the sentence order yields a description in which effects precede their causes.

(2) *Noncausal texts*: Collections of independent factual statements whose ordering carries no temporal or causal implication (e.g., "The library lent a heavy atlas. The library hosted a chess tournament. The library displayed old framed photographs. The library received a donated wooden shelf from a volunteer"). Reversing the sentence order leaves the meaning essentially unchanged.

The two examples above are included as fixed entries in every corpus. To make the surface form of the two sets more comparable, the prompt imposes a common format: All four sentences in each text begin with the same noun phrase, every sentence is written in the simple past tense, and pronouns,

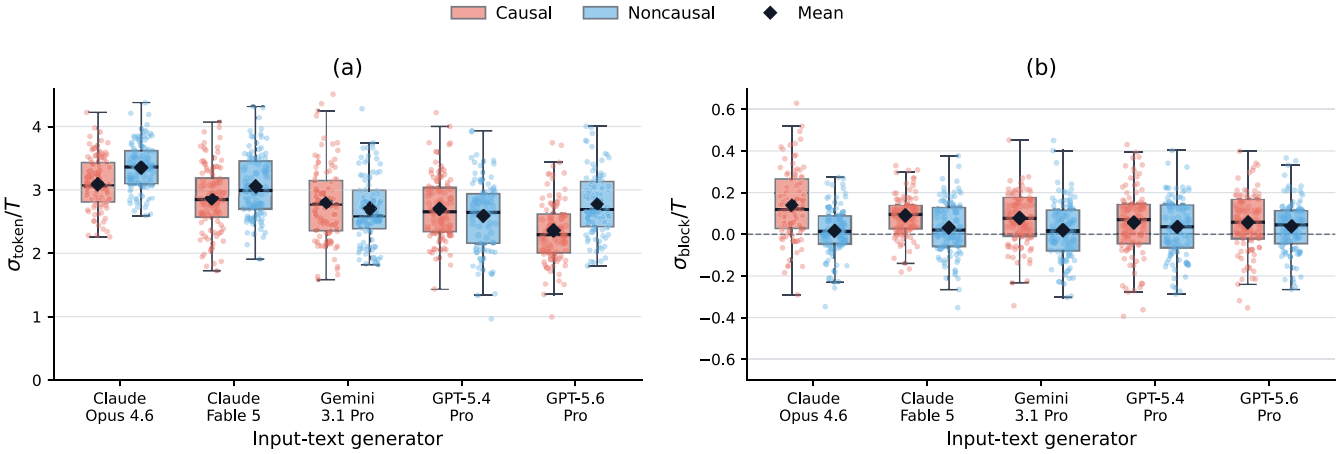


FIG. 4. Distributions of the stochastic entropy production evaluated using GPT-2 for the fixed texts generated by the five input-text models. (a) Token-level reversal σ_{token}/T. (b) Sentence-block reversal σ_{block}/T. For each generator, red and blue denote the 100 causal and 100 noncausal texts, respectively. All individual observations are shown with horizontal jitter, and no observations are removed as outliers. Boxes span the interquartile range, horizontal lines indicate the median, whiskers extend to the most extreme observation within 1.5 times the interquartile range from the box, and black diamonds indicate the mean. Since the input texts are not sampled from GPT-2's distribution, the averages of σ are not guaranteed to be non-negative. Note the different vertical scales in the two panels.

demonstratives, and intersentence connectives are excluded. The complete text sets, prompts, analysis code, and raw numerical results are provided in the GitHub repository (see the Data Availability statement).

For every text, both the token-reversal quantity σ_{token}/T and the sentence-block-reversal quantity σ_{block}/T are evaluated using the same GPT-2 model; the five models listed above are used only to generate the input texts.

The resulting distributions are shown in Fig. 4. The token-level difference between the two categories varies strongly across input-text generators, whereas the block-level difference is consistently larger for the causal texts.

For each generator and reversal scheme, we compare the causal and noncausal samples using a two-sided asymptotic Mann-Whitney U test under an independence approximation. We quantify the direction and magnitude of the difference by the rank-biserial correlation

$$r \equiv \frac{2U_{\text{C}}}{n_{\text{C}}n_{\text{NC}}} - 1 = \frac{N_{>} - N_{<}}{n_{\text{C}}n_{\text{NC}}}, \tag{35}$$

where U_{C} is the Mann-Whitney statistic for the causal sample, n_{C} and n_{NC} are the causal and noncausal sample sizes, and $N_{>}$ ($N_{<}$) is the number of cross-group pairs for which the causal value is larger (smaller) than the noncausal value. Tied pairs contribute one half to U_{C} and zero to $N_{>} - N_{<}$. Thus, $-1 \leqslant r \leqslant 1$: $r > 0$ means that the causal value tends to be larger, $r < 0$ means that the noncausal value tends to be larger, and $r = 0$ indicates no net pairwise ordering preference.

Figure 5 reveals a qualitative difference between the two reversal schemes. For token reversal, the causal/noncausal difference depends strongly on the input-text generator: Three of the five models show significantly smaller σ_{token}/T for the causal texts ($r = -0.46$ to -0.20), while the other two show small positive estimates whose 95% confidence intervals include zero. This variation, which far exceeds the sampling uncertainty, is consistent with sensitivity to generator-specific stylistic or structural properties, although the present experiment does not identify which such properties drive it.

For block reversal, in contrast, every significant difference (three of the five generators) is in the direction of larger σ_{block}/T for the causal texts ($r = +0.26$ to $+0.46$), which is the direction expected by construction. The remaining two estimates are positive but small and not significant, and no generator shows a significant difference in the opposite direction. That is, the magnitude of the block-level effect varies across generators, but its sign always agrees with the expected direction. Detailed values of U, p, and r are reported in Appendix B 4.

These results favor block reversal over token reversal as a probe of intersentence ordering in the present setting: Its response to the causal/noncausal manipulation is consistent in sign and interpretable in terms of event ordering.

In Appendix C (Figs. 10 and 11), we repeat the experiment with the same five corpora but with a substantially larger evaluator, Qwen3-4B-Base (4.0 billion parameters). The block-level difference is then statistically significant for all five generators (largest $p = 0.045$, Table III; all five individual unadjusted tests are significant). In contrast, the token-level difference remains generator dependent, with significant effects in both directions. This reevaluation thus supports the GPT-2 results with improved statistical resolution.

We do not claim, however, that these results establish σ_{block}/T as a general quantitative measure of causal structure. The causal/noncausal labels are defined by the generation prompt rather than by a formal criterion, and correlations between these labels and generator-specific stylistic features cannot be completely excluded. A more rigorous assessment would require a prompt-independent definition of causal ordering; constructing such a definition remains an important direction for future work.

Finally, our "causal" and "noncausal" categories do not rigorously distinguish mere temporal ordering from genuine

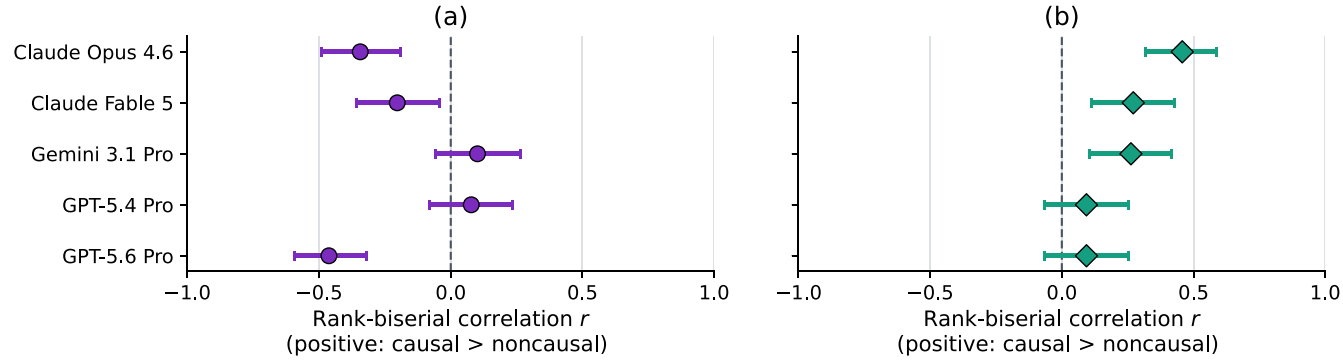


FIG. 5. Cross-generator comparison of the causal/noncausal difference in the per-token stochastic entropy production evaluated by GPT-2. The labels on the vertical axis indicate the language model used to generate each input-text corpus. (a) Token-level reversal $\sigma_{\mathrm{token}}/T$. (b) Sentence-block reversal $\sigma_{\mathrm{block}}/T$. Each point is the rank-biserial correlation r defined in Eq. (35) for 100 causal and 100 noncausal texts. Positive values indicate that the causal texts tend to have larger values than the noncausal texts. Error bars show pointwise 95% percentile bootstrap confidence intervals obtained by resampling independently within the two text categories ($B = 10^4$). All observations are included; no outlier removal is performed.

causal dependence—a distinction that is inherently difficult even in principle [60]. Recent work has shown that LLMs in fact tend to confound these two relations [61].

VI. LINEAR GAUSSIAN CASE: KALMAN INNOVATION REPRESENTATION

As an analytically tractable demonstration of the general framework, we specialize to the linear Gaussian case, where the autoregressive model coincides with the innovation representation of the steady-state Kalman filter [44–46]. We obtain an analytical expression for the entropy production $\mathcal{S}_y$ by introducing the innovation reversal matrix $\mathcal{R}$, which is also verified by Monte Carlo sampling.

We note that Refs. [62–65] studied entropy production associated with the information flow between the signal process x_t and the observation process y_t in continuous-time Kalman-Bucy and nonlinear filters. In contrast, the quantity computed below is the KL divergence between the forward and time-reversed path measures of the observation sequence y_t alone, with no reference to the underlying state x_t.

Separately, thermodynamic aspects of linear Gaussian systems involving Kalman filtering have also been explored from complementary perspectives in Refs. [55,66–68]. We note that the entropy production in continuous-time linear Gaussian systems has also been analyzed in the context of multivariate Ornstein-Uhlenbeck processes [69–71].

A. Setup

Consider a linear Gaussian process with state dimension n_x and observation dimension n_y:

$$x_{t+1} = Ax_t + w_t, \quad w_t \sim \mathcal{N}(0, Q), \tag{36}$$

$$y_t = Cx_t + v_t, \quad v_t \sim \mathcal{N}(0, R), \tag{37}$$

where we assume, for simplicity, $A \in \mathbb{R}^{n_x \times n_x}$, $C \in \mathbb{R}^{n_y \times n_x}$, $Q \in \mathbb{R}^{n_x \times n_x}$ (positive semidefinite), and $R \in \mathbb{R}^{n_y \times n_y}$ (positive definite) are all time independent. The noise sequences $\{w_t\}$ and $\{v_t\}$ are mutually independent and independent across time.

We assume that the Kalman filter [44] has reached steady state (see, e.g., Ref. [45] for standard results used below). Then, the following relevant quantities are all time independent: The prediction error covariance P is given by the positive semidefinite solution of the Riccati equation

$$P = A(P - PC^\top (CPC^\top + R)^{-1} CP)A^\top + Q, \tag{38}$$

the innovation covariance is given by $S = CPC^\top + R$, and the Kalman gain is given by $K = PC^\top S^{-1}$. This stationary assumption concerns the filter parameters only; by itself, it does not imply that the finite-time observation path probability is stationary.

To connect with the general framework of Sec. II A, note that the true state x_t in Eqs. (36) and (37) has no counterpart in our framework: It appears neither in the latent state h_t nor in the emission kernel p_t. Instead, the latent state is identified as $h_t = \hat{x}_{t+1|t}$, the one-step-ahead predicted estimate (with $h_0 = \hat{x}_{1|0}$), which is a deterministic function of past observations $y_1, \ldots, y_t$, and the emission kernel is $p_t(y_{t+1} \mid h_t) = \mathcal{N}(Ch_t, S)$. The generative process of Sec. VI B thus produces $y_{1:T}$ entirely through the deterministic-stochastic loop between h_t and y_{t+1}, with x_t absent; this is the sense in which we regard the Kalman filter as a generative model (see also Sec. II C and Table I).

B. Forward process

The generative process produces a sequence $y_1, y_2, \ldots, y_T$ as follows. Fix an initial predicted state estimate $\hat{x}_{1|0}$ (e.g., $\hat{x}_{1|0} = 0$). Then, for $t = 1, 2, \ldots, T$:

(1) *Draw an innovation*: Sample $e_t \sim \mathcal{N}(0, S)$ independently.

(2) *Generate the observation*: Set

$$y_t = C\hat{x}_{t|t-1} + e_t. \tag{39}$$

(3) *Kalman filter update*: Compute the next predicted state estimate deterministically:

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K(y_t - C\hat{x}_{t|t-1}), \tag{40}$$

$$\hat{x}_{t+1|t} = A\hat{x}_{t|t}. \tag{41}$$

Since $e_t = y_t - C\hat{x}_{t|t-1}$ and $\hat{x}_{t|t-1}$ is a deterministic function of $y_1, \ldots, y_{t-1}$, the conditional distribution is

$$y_t \mid y_{1:t-1} \sim \mathcal{N}(C\hat{x}_{t|t-1}, S). \tag{42}$$

By the innovation decomposition [45], the joint density of the forward path is

$$P_{\rightarrow}(y_{1:T}) = \prod_{t=1}^{T} \mathcal{N}(e_t; 0, S) = \frac{1}{(2\pi)^{Tn_y/2}(\det S)^{T/2}} \exp\left(-\frac{1}{2}\sum_{t=1}^{T} e_t^\top S^{-1} e_t\right). \tag{43}$$

This coincides with the marginal distribution of $y_{1:T}$ obtained by integrating out the states $x_{1:T}$ in the original dynamics, for a compatible initial distribution of x_1. Hence, for any fixed initial predictor $\hat{x}_{1|0}$, the forward path vector $y_{1:T} = (y_1, \ldots, y_T)$ is Gaussian:

$$P_{\rightarrow}(y_{1:T}) = \mathcal{N}(y_{1:T}; m, \Sigma), \tag{44}$$

where m and Σ are the mean vector and the covariance matrix, determined by the update rule described above. If $\hat{x}_{1|0} = 0$, then $m = 0$; this is always assumed in the following.

Combining the filter update (40) and prediction (41) and iterating from $\hat{x}_{1|0} = 0$, one obtains the causal moving-average (innovation) representation [45,46]

$$y_t = \sum_{k=1}^{t} H_{t-k} e_k, \tag{45}$$

where the impulse-response coefficients are

$$H_0 \equiv I_{n_y}, \quad H_l \equiv CA^l K \quad (l \geqslant 1). \tag{46}$$

With the stacked vectors $\mathbf{y} \equiv (y_1^\top, \ldots, y_T^\top)^\top$ and $\mathbf{e} \equiv (e_1^\top, \ldots, e_T^\top)^\top$, Eq. (45) reads

$$\mathbf{y} = \mathcal{H}\mathbf{e}, \tag{47}$$

where $\mathcal{H} \in \mathbb{R}^{Tn_y \times Tn_y}$ is the block lower-triangular matrix with (i, j) block H_{i-j} for $i \geqslant j$ and zero otherwise. The covariance matrix of $\mathbf{e}$ is given by

$$\mathbb{E}_{P_{\rightarrow}}[\mathbf{e}\,\mathbf{e}^\top] = I_T \otimes S, \tag{48}$$

where I_T is the identity matrix of the auxiliary T-dimensional system that encodes the labels of time and $\otimes$ denotes the tensor product between the auxiliary space and the original n_y-dimensional space. The path covariance matrix is then given by

$$\Sigma = \mathbb{E}_{P_{\rightarrow}}[\mathbf{y}\,\mathbf{y}^\top] = \mathcal{H}(I_T \otimes S)\mathcal{H}^\top. \tag{49}$$

C. Backward process

Following the general protocol described in Sec. III A, we run the generative mechanism of Sec. VI B on the backward sequence $\tilde{y}_{1:T}$:

(1) *Draw an innovation*: Sample $\tilde{e}_s^B \sim \mathcal{N}(0, S)$ independently.

(2) *Generate the observation*: Set

$$\tilde{y}_s = C\hat{x}_{s|s-1}^B + \tilde{e}_s^B. \tag{50}$$

(3) *Kalman filter update*: Compute the next predicted state estimate deterministically:

$$\hat{x}_{s|s}^B = \hat{x}_{s|s-1}^B + K\left(\tilde{y}_s - C\hat{x}_{s|s-1}^B\right), \tag{51}$$

$$\hat{x}_{s+1|s}^B = A\hat{x}_{s|s}^B. \tag{52}$$

We set the initial condition as $\hat{x}_{1|0}^B = \hat{x}_{1|0}$.

Since each innovation in the generative process is an independent draw from $\mathcal{N}(0, S)$, the backward path density is the product of the densities $\mathcal{N}(\tilde{e}_s^B; 0, S)$ evaluated at the values (50)

$$P_{\leftarrow}(\tilde{y}_{1:T}) = \prod_{s=1}^{T} \mathcal{N}(\tilde{e}_s^B; 0, S) = \frac{1}{(2\pi)^{Tn_y/2}(\det S)^{T/2}} \exp\left(-\frac{1}{2}\sum_{s=1}^{T} (\tilde{e}_s^B)^\top S^{-1} \tilde{e}_s^B\right). \tag{53}$$

Here, we do not assume Eq. (10) but assume that $\tilde{y}_1$ is sampled from an independent Gaussian distribution $\mathcal{N}(C\hat{x}_{1|0}, S)$. Since the forward and backward path probabilities [Eqs. (43) and (53)] share the same functional form,

$$P_{\leftarrow}(\tilde{\mathbf{y}}) = \mathcal{N}(\tilde{\mathbf{y}}; 0, \Sigma). \tag{54}$$

We now consider the particular event that the backward-trajectory realization is the exact time-reversal of the forward trajectory,

$$\tilde{y}_s = y_{T-s+1}, \quad s = 1, 2, \ldots, T. \tag{55}$$

With the time-reversal (permutation) matrix

$$J = \begin{pmatrix} 0 & \cdots & 0 & I_{n_y} \\ 0 & \cdots & I_{n_y} & 0 \\ \vdots & & \vdots & \vdots \\ I_{n_y} & \cdots & 0 & 0 \end{pmatrix} \in \mathbb{R}^{Tn_y \times Tn_y}, \tag{56}$$

we have $J\mathbf{y} = (y_T^\top, \ldots, y_1^\top)^\top$. Note that $J = J^\top = J^{-1}$ and $\det J = \pm 1$. By substituting $\tilde{\mathbf{y}} = J\mathbf{y}$ into Eq. (54),

$$P_{\leftarrow}(J\mathbf{y}) = \mathcal{N}(\mathbf{y}; 0, \widetilde{\Sigma}), \tag{57}$$

with covariance

$$\widetilde{\Sigma} = J\Sigma J. \tag{58}$$

Therefore, the sampling of observed trajectories in the backward process can be mapped to the sampling in the forward process, just by applying J as above.

We now introduce $(e_1^B, e_2^B, \ldots, e_T^B)$ via $J\mathbf{y} = \mathcal{H}\mathbf{e}^B$, that is,

$$\mathbf{e}^B \equiv \mathcal{H}^{-1} J\mathbf{y}. \tag{59}$$

Note that $\mathcal{H}$ is always invertible because the diagonal blocks are given by $H_0 = I_{n_y}$. From $\mathbf{y} = \mathcal{H}\mathbf{e}$, $\mathbf{e}^B$ is related to the original forward innovation as

$$\mathbf{e}^B = \mathcal{R}\mathbf{e}, \tag{60}$$

where we introduced the *innovation reversal matrix*

$$\mathcal{R} \equiv \mathcal{H}^{-1} J \mathcal{H}. \tag{61}$$

The backward path density (53) defines a probability measure $P_{\leftarrow}$ on the space of observation sequences via the generative process (50)–(52), in which the innovations $\tilde{e}_s^B$ are independently sampled from $\mathcal{N}(0, S)$ and causally produce the observations $\tilde{y}_s$. To compute the entropy production $\sigma = \ln P_{\rightarrow}(\mathbf{y}) - \ln P_{\leftarrow}(J\mathbf{y})$ for a given forward realization $\mathbf{y}$, one needs the value of $P_{\leftarrow}(J\mathbf{y})$. This value is obtained without sampling from the backward process: One substitutes $\tilde{y}_s = y_{T-s+1}$ into the density (53), which requires the innovations $\tilde{e}_s^B$ as a function of the observations. To this end, one reads the relation (50) in the reverse direction: given an observation $\tilde{y}_s$, one extracts the innovation as $\tilde{e}_s^B = \tilde{y}_s - C\hat{x}_{s|s-1}^B$, and then updates the state via Eqs. (51) and (52). Substituting the particular values $\tilde{y}_s = y_{T-s+1}$ and writing e_s^B for the resulting innovations, one obtains, starting from $\hat{x}_{1|0}^B = \hat{x}_{1|0}$, for $s = 1, 2, \ldots, T$,

$$e_s^B = y_{T-s+1} - C\hat{x}_{s|s-1}^B, \tag{62}$$

$$\hat{x}_{s|s}^B = \hat{x}_{s|s-1}^B + Ke_s^B, \tag{63}$$

$$\hat{x}_{s+1|s}^B = A\hat{x}_{s|s}^B. \tag{64}$$

This is precisely the operation encoded by $\mathcal{H}^{-1}$ in $\mathbf{e}^B = \mathcal{H}^{-1}J\mathbf{y}$: It runs the Kalman filter on the time-reversed observation sequence $(y_T, y_{T-1}, \ldots, y_1)$ and deterministically extracts the innovation at each step. We note that $\hat{x}_{s|s-1}^B$ is not the time reversal of $\hat{x}_{t|t-1}$ even when $\tilde{y}_s = y_{T-s+1}$, as mentioned for the general case in Sec. III.

For the particular realization $\tilde{y}_s = y_{T-s+1}$, the quantities $\tilde{e}_s^B$ and e_s^B take the same numerical value, since they are computed by the same recursion applied to the same input sequence; the distinction lies solely in the probability distributions. In the backward generative process, $\tilde{e}_s^B$ is the independent random input that produces $\tilde{y}_s$ via Eq. (50); under $P_{\leftarrow}$, it is independent and identically distributed (i.i.d.) according to $\mathcal{N}(0, S)$ by construction. In contrast, e_s^B is deterministically extracted from the forward trajectory $y_{1:T}$ via Eqs. (62)–(64); under $P_{\rightarrow}$, $e_s^B = [\mathcal{R}\mathbf{e}]_s$ is in general correlated across time steps and not identically distributed. It is precisely this mismatch between the i.i.d. statistics assumed by the backward model and the actual statistics of e_s^B under $P_{\rightarrow}$ that gives rise to a nonzero entropy production.

D. Analytical expression for entropy production

In general, the KL divergence between two Gaussians $\mathcal{N}(\mu_1, \Sigma_1)$ and $\mathcal{N}(\mu_2, \Sigma_2)$ is (see, e.g., Ref. [72])

$$\begin{aligned} &D_{\mathrm{KL}}(\mathcal{N}(\mu_1, \Sigma_1)\|\mathcal{N}(\mu_2, \Sigma_2)) \\ &= \frac{1}{2}\Big[\mathrm{tr}\big(\Sigma_2^{-1}\Sigma_1\big) + (\mu_1 - \mu_2)^{\top}\Sigma_2^{-1}(\mu_1 - \mu_2) \\ &\quad - d + \ln\frac{\det\Sigma_2}{\det\Sigma_1}\Big], \end{aligned} \tag{65}$$

where d is the dimension of the random vector.

We now evaluate each term with $\mu_1 = \mu_2 = 0$, $\Sigma_1 = \Sigma$, $\Sigma_2 = \widetilde{\Sigma} = J\Sigma J$ (58), and $d = Tn_y$. Since both distributions have zero mean, the quadratic mean-difference term vanishes. Since J is an orthogonal matrix ($J^{\top}J = I$),

$$\det\widetilde{\Sigma} = \det(J\Sigma J) = (\det J)^2\det\Sigma = \det\Sigma. \tag{66}$$

Therefore,

$$\ln\frac{\det\widetilde{\Sigma}}{\det\Sigma} = 0. \tag{67}$$

The inverse of $\widetilde{\Sigma}$ is $\widetilde{\Sigma}^{-1}{=}(J\Sigma J)^{-1}{=}J^{-1}\Sigma^{-1}J^{-1}{=}J\Sigma^{-1}J$, where we used $J^{-1} = J$. Thus,

$$\mathrm{tr}(\widetilde{\Sigma}^{-1}\Sigma) = \mathrm{tr}(J\Sigma^{-1}J\Sigma). \tag{68}$$

Combining Eqs. (67) and (68) into Eq. (65), we obtain

$$D_{\mathrm{KL}}(P_{\rightarrow}(\mathbf{y})\|P_{\leftarrow}(J\mathbf{y})) = \tfrac{1}{2}[\mathrm{tr}(J\Sigma^{-1}J\Sigma) - Tn_y]. \tag{69}$$

We next rewrite Eq. (69) in terms of $\mathcal{R}$ (61). From the factorization (49) and by letting $\mathcal{G} \equiv \mathcal{H}^{-1}$, $\Sigma^{-1} = \mathcal{G}^{\top}(I_T \otimes S^{-1})\mathcal{G}$. Substituting into the trace term in Eq. (69) and using the cyclic property of the trace together with $J = J^{\top} = J^{-1}$,

$$\mathrm{tr}(J\Sigma^{-1}J\Sigma) = \mathrm{tr}\left((I_T \otimes S^{-1})\underbrace{\mathcal{G}J\mathcal{H}}_{\mathcal{R}}(I_T \otimes S)\underbrace{\mathcal{H}^{\top}J\mathcal{G}^{\top}}_{\mathcal{R}^{\top}}\right). \tag{70}$$

Therefore, we obtain

$$\begin{aligned} &D_{\mathrm{KL}}(P_{\rightarrow}(\mathbf{y})\|P_{\leftarrow}(J\mathbf{y})) \\ &\quad = \tfrac{1}{2}[\mathrm{tr}((I_T \otimes S^{-1})\mathcal{R}(I_T \otimes S)\mathcal{R}^{\top}) - Tn_y]. \end{aligned} \tag{71}$$

Meanwhile, since $\mathbf{e}^B = \mathcal{R}\mathbf{e}$ and $\mathbf{e} \sim \mathcal{N}(0, I_T \otimes S)$ under the forward measure $P_{\rightarrow}$, each $e_s^B = \sum_{k=1}^{T}[\mathcal{R}]_{sk}e_k$ has covariance

$$\begin{aligned} \Sigma_s^B \equiv \mathbb{E}_{P_{\rightarrow}}\big[e_s^B\big(e_s^B\big)^{\top}\big] &= \sum_{k,k'=1}^{T}[\mathcal{R}]_{sk}\underbrace{\mathbb{E}_{P_{\rightarrow}}[e_k e_{k'}^{\top}]}_{S\delta_{kk'}}[\mathcal{R}]_{sk'}^{\top} \\ &= \sum_{k=1}^{T}[\mathcal{R}]_{sk}S[\mathcal{R}]_{sk}^{\top}, \end{aligned} \tag{72}$$

where the subscripts run over the auxiliary subspace of the time indexes. Thus, Eq. (71) can be written as

$$D_{\mathrm{KL}}(P_{\rightarrow}(\mathbf{y})\|P_{\leftarrow}(J\mathbf{y})) = \frac{1}{2}\sum_{s=1}^{T}\big[\mathrm{tr}\big(S^{-1}\Sigma_s^B\big) - n_y\big]. \tag{73}$$

By noticing that

$$\mathbb{E}_{P_{\rightarrow}}\big[\big(e_s^B\big)^{\top}S^{-1}e_s^B\big] = \mathrm{tr}\big(S^{-1}\Sigma_s^B\big), \tag{74}$$

Eq. (73) can be further rewritten as

$$D_{\mathrm{KL}}(P_{\rightarrow}(\mathbf{y})\|P_{\leftarrow}(J\mathbf{y})) = \frac{1}{2}\sum_{s=1}^{T}\big(\mathbb{E}_{P_{\rightarrow}}\big[\big(e_s^B\big)^{\top}S^{-1}e_s^B\big] - n_y\big). \tag{75}$$

As a consistency check, we directly calculate the log-ratio of the forward and backward path probabilities from Eqs. (43) and (53) (i.e., the stochastic entropy production):

$$\sigma(\mathbf{y}) = \ln\frac{P_{\rightarrow}(\mathbf{y})}{P_{\leftarrow}(J\mathbf{y})} = \frac{1}{2}\sum_{s=1}^{T}\big(e_s^B\big)^{\top}S^{-1}e_s^B - \frac{1}{2}\sum_{t=1}^{T}e_t^{\top}S^{-1}e_t, \tag{76}$$

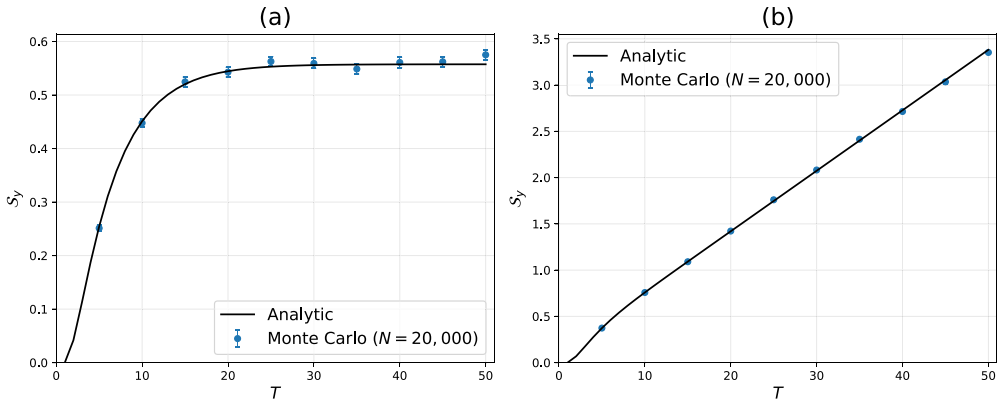


FIG. 6. Numerical verification of the analytical entropy production (71) by Monte Carlo sampling (23)–(24) with $N = 20\,000$ trajectories. Solid curves: analytical values; circles with error bars: Monte Carlo estimates. Error bars indicate ± 1 standard error of the mean, $\mathrm{SE} = \mathrm{std}(\sigma)/\sqrt{N}$. (a) Scalar case ($n_x = n_y = 1$) with $A = 0.9$, $C = 1$, $Q = 1$, and $R = 1$, and (b) multivariate case ($n_x = n_y = 2$) with $A = \left(\begin{smallmatrix} 0.8 & 0.3 \\ 0 & 0.5 \end{smallmatrix}\right)$, $C = \left(\begin{smallmatrix} 1 & 0.5 \\ 0 & 1 \end{smallmatrix}\right)$, $Q = I_2$, and $R = I_2$.

where we used $\tilde{e}_s^B = e_s^B$ for the realization $\tilde{y}_s = y_{T-s+1}$ (see the discussion above). Indeed, Eq. (75) is reproduced from

$$\mathbb{E}_{P_\rightarrow}\left[\ln \frac{P_\rightarrow(\mathbf{y})}{P_\leftarrow(J\mathbf{y})}\right] = \frac{1}{2}\sum_{s=1}^{T}\left(\mathbb{E}_{P_\rightarrow}\left[\left(e_s^B\right)^\top S^{-1} e_s^B\right] - n_y\right), \quad (77)$$

where we used

$$\mathbb{E}_{P_\rightarrow}\left[e_t^\top S^{-1} e_t\right] = \mathbb{E}_{P_\rightarrow}\left[\mathrm{tr}\left[S^{-1} e_t e_t^\top\right]\right] = \mathrm{tr}\left[I_{n_y}\right] = n_y. \quad (78)$$

The expression (75) provides an operational meaning of the entropy production. Under the forward measure $P_\rightarrow$, the covariance Σ_s^B of the backward innovation e_s^B generally differs from S, and Eq. (75) quantifies this cumulative mismatch between the backward innovation statistics expected by the model and those actually realized under the forward dynamics.

Finally, we remark on the asymptotic regime $T \to \infty$, where we assume that A is stable (that is, all eigenvalues strictly lie inside the unit circle). In the scalar case ($n_y = 1$), every stationary Gaussian process is time reversible [73], so the entropy production remains bounded as $T \to \infty$ and is a boundary effect of the deterministic initial condition $\hat{x}_{1|0} = 0$. In the multivariate case ($n_y > 1$), stationary Gaussian processes can be time irreversible whenever the cross-covariance matrices are not symmetric [74,75], and the entropy production can grow linearly with T.

Scalar case ($n_x = n_y = 1$). When $n_x = n_y = 1$, the factors S^{-1} and S in Eq. (71) cancel, yielding

$$D_{\mathrm{KL}}(P_\rightarrow(\mathbf{y}) \| P_\leftarrow(J\mathbf{y})) = \tfrac{1}{2}\left(\|\mathcal{R}\|_F^2 - T\right), \quad (79)$$

where $\|\mathcal{R}\|_F^2 = \sum_{s,k} \mathcal{R}_{sk}^2$ is the squared Frobenius norm.

We consider a further specific case: $T = 2$. Let $H \equiv CAK = H_1$. Then,

$$\mathcal{H} = \begin{pmatrix} 1 & 0 \\ H & 1 \end{pmatrix}, \quad \mathcal{G} = \begin{pmatrix} 1 & 0 \\ -H & 1 \end{pmatrix}, \quad J = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad (80)$$

so that

$$\mathcal{R} = \mathcal{G}J\mathcal{H} = \begin{pmatrix} H & 1 \\ 1 - H^2 & -H \end{pmatrix}. \quad (81)$$

Hence, $\|\mathcal{R}\|_F^2 = 2 + H^4$ and

$$D_{\mathrm{KL}}(P_\rightarrow(\mathbf{y}) \| P_\leftarrow(J\mathbf{y})) = \frac{H^4}{2} = \frac{(CAK)^4}{2}. \quad (82)$$

E. Numerical verification by Monte Carlo sampling

As a demonstration of the sampling procedure proposed in Sec. IV C, we numerically verify the analytical expression (71) for the entropy production in the linear Gaussian setting. Two parameter sets for Eqs. (36) and (37) are examined: a scalar case ($n_x = n_y = 1$) and a multivariate case ($n_x = n_y = 2$); the specific values are listed in the caption of Fig. 6. All eigenvalues of A lie strictly inside the unit circle. We set $\hat{x}_{1|0} = \hat{x}_{1|0}^B = 0$.

For each sequence length T, $N = 20\,000$ trajectories $y_{1:T}^{(n)}$ are sampled from the forward generative process (Sec. VI B), and the stochastic entropy production $\sigma(y_{1:T}^{(n)})$ (23) is computed via the forward and backward passes as described in Sec. IV C. In the present Gaussian setting, the normalization constants of the emission kernels cancel between the forward and backward sums, reducing σ to the difference in quadratic forms as shown in Eq. (76). The entropy production is then estimated by Eq. (24).

Figure 6 shows the analytical entropy production (71) together with the Monte Carlo estimates as functions of T. For both parameter sets, the Monte Carlo estimates are in good agreement with the analytical values across the entire range $T = 5, 10, \ldots, 50$. In the scalar case [panel (a)], the entropy production saturates to a finite value, consistent with the time reversibility of stationary scalar Gaussian processes [73]. In the multivariate case [panel (b)], the entropy production grows approximately linearly with T, reflecting the genuine time irreversibility of the multivariate process [74,75].

We emphasize that the Monte Carlo procedure involves stochastic sampling only in the forward direction. For each sample trajectory $y_{1:T}^{(n)} \sim P_\rightarrow$, the forward innovations e_t are drawn independently from $\mathcal{N}(0, S)$, from which $y_{1:T}$ is generated via Eqs. (39)–(41). The backward innovations e_s^B are then obtained deterministically from the same $y_{1:T}$ via

Eqs. (62)–(64), without any additional random sampling. As discussed in Sec. VI C, for the realization $\tilde{y}_s = y_{T-s+1}$, the quantities e_s^B and $\tilde{e}_s^B$ take the same numerical value, but their statistical properties under $P_{\rightarrow}$ differ from the i.i.d. $\mathcal{N}(0, S)$ assumed by the backward model.

VII. RETROSPECTIVE DECOMPOSITIONS OF ENTROPY PRODUCTION

Turning to the general setup of Sec. II A, we derive exact decompositions of the entropy production that hold in the general, nonrecursive setting. The entropy production is first split into non-negative per-step contributions via retrospective Bayesian inference, and each contribution is further decomposed into a compression loss and a model mismatch. These decompositions reveal a set of information-theoretic structures and are of fundamental interest for the thermodynamics of information in non-Markovian processes. Note that *retrospective* refers only to the Bayesian decomposition below, not to the definition of the protocol-reversed backward process introduced in Sec. III.

A. Per-step decomposition

We decompose the entropy production into a sum of non-negative per-step terms by comparing the forward and backward path measures at each time step through retrospective Bayesian inference.

From the standard Bayesian rule (i.e., the chain rule),

$$P_{\rightarrow}(y_{t:T}) = P_{\rightarrow}(y_t \mid y_{t+1:T}) P_{\rightarrow}(y_{t+1:T}). \tag{83}$$

By applying this iteratively, we obtain

$$P_{\rightarrow}(y_{1:T}) = \prod_{t=1}^{T} P_{\rightarrow}(y_t \mid y_{t+1:T}). \tag{84}$$

Also by definition,

$$P_{\leftarrow}(y_{T:1}) = \prod_{t=1}^{T} p_t(y_t \mid g_{t+1}^{\leftarrow}(y_{T:t+1})). \tag{85}$$

Thus,

$$\mathcal{S}_y = \mathbb{E}_{P_{\rightarrow}}\left[\ln \frac{P_{\rightarrow}(y_{1:T})}{P_{\leftarrow}(y_{T:1})}\right] = \mathbb{E}_{P_{\rightarrow}}\left[\sum_{t=1}^{T} \ln \frac{P_{\rightarrow}(y_t \mid y_{t+1:T})}{p_t(y_t \mid g_{t+1}^{\leftarrow}(y_{T:t+1}))}\right]. \tag{86}$$

We finally obtain

$$\mathcal{S}_y = \sum_{t=1}^{T} \mathcal{D}_t, \tag{87}$$

where $\mathcal{D}_t \equiv \mathbb{E}_{y_{t+1:T}\sim P_{\rightarrow}}[D_{\mathrm{KL}}(P_{\rightarrow}(y_t \mid y_{t+1:T}) \| p_t(y_t \mid g_{t+1}^{\leftarrow}(y_{T:t+1})))] \geqslant 0$, or equivalently,

$$\mathcal{D}_t \equiv \mathbb{E}_{P_{\rightarrow}}\left[\ln \frac{P_{\rightarrow}(y_t \mid y_{t+1:T})}{p_t(y_t \mid g_{t+1}^{\leftarrow}(y_{T:t+1}))}\right]. \tag{88}$$

The above result implies that the total entropy production can be decomposed into non-negative terms associated with individual time steps, even when the entire process is non-Markovian. From this, the equality $\mathcal{S}_y = 0$ holds if and only if $\mathcal{D}_t = 0$ for every t, equivalently, if and only if $p_t(\cdot \mid g_{t+1}^{\leftarrow}(y_{T:t+1})) = P_{\rightarrow}(\cdot \mid y_{t+1:T})$ for almost every $y_{t+1:T}$ and every t.

The quantity $\mathcal{D}_t$ is the expected mismatch between the Bayesian retrospective distribution $P_{\rightarrow}(y_t \mid y_{t+1:T})$ and the conditional distribution used by the actual backward model. This sheds light on the concept of entropy production; the total entropy production can be interpreted as the gap between the retrospective Bayesian distribution and the physical (that is, protocol-reversed) backward model. In the spirit of variational inference theory [52–54], $p_t(y_t \mid g_{t+1}^{\leftarrow}(y_{T:t+1}))$ can be interpreted as a test function that approximates the Bayesian posterior $P_{\rightarrow}(y_t \mid y_{t+1:T})$.

We note that the reverse process employed in diffusion models [76–78] corresponds to the Bayesian retrodiction, where the time evolution of the probability distribution itself is time reversed [79]. This is conceptually distinct from the backward process in stochastic thermodynamics, where the control protocol is time reversed. Their gap is measured precisely by the entropy production for continuous-time diffusion models [80]; our result above represents another manifestation of this gap.

A simple everyday example may help illustrate this gap. Consider a statement: "If I don't study, then my mom will get angry." Here, we take y_1 as an episode "I don't study" and y_2 as "my mom gets angry." The retrospective Bayesian inference is "If my mom gets angry, that implies I didn't study," which we intuitively expect to be true in daily life (its probability should be close to one). (Note that the contrapositive is "If my mom doesn't get angry, that implies I studied," which is logically true if the original statement is true.) On the other hand, the backward process of our framework is given by reversing the real-time ordering, "If my mom gets angry, then I will not study," which we intuitively expect to be false in daily life (its probability should be close to zero). Therefore, this daily-life situation is highly irreversible.

B. Compression loss and model mismatch

We further decompose each per-step contribution $\mathcal{D}_t$ into two separately non-negative terms: A compression loss, which quantifies the retrospective information discarded by the backward summary, and a model mismatch, which quantifies the cost of reusing the forward emission kernel in the backward direction.

Write $g_{t+1}^{\leftarrow} \equiv g_{t+1}^{\leftarrow}(y_{T:t+1})$ for brevity. For each fixed $y_{t+1:T}$, insert the intermediate distribution $P_{\rightarrow}(y_t \mid g_{t+1}^{\leftarrow})$ into the KL divergence:

$$\begin{aligned} &D_{\mathrm{KL}}(P_{\rightarrow}(y_t \mid y_{t+1:T}) \| p_t(y_t \mid g_{t+1}^{\leftarrow})) \\ &\quad = D_{\mathrm{KL}}(P_{\rightarrow}(y_t \mid y_{t+1:T}) \| P_{\rightarrow}(y_t \mid g_{t+1}^{\leftarrow})) \\ &\qquad + \sum_{y_t} P_{\rightarrow}(y_t \mid y_{t+1:T}) \ln \frac{P_{\rightarrow}(y_t \mid g_{t+1}^{\leftarrow})}{p_t(y_t \mid g_{t+1}^{\leftarrow})}. \end{aligned} \tag{89}$$

Here, $P_{\rightarrow}(y_t \mid g_{t+1}^{\leftarrow})$ denotes the Bayesian retrospective distribution of y_t conditioned on $g_{t+1}^{\leftarrow}$, instead of the full future $y_{t+1:T}$. Taking the expectation over $y_{t+1:T} \sim P_{\rightarrow}$, the two terms become the terms called the compression loss and the model mismatch as follows.

Compression loss. Since $g^{\leftarrow}_{t+1}$ is a deterministic function of $y_{t+1:T}$, $P_{\rightarrow}(y_t \mid y_{t+1:T}) = P_{\rightarrow}(y_t \mid y_{t+1:T}, g^{\leftarrow}_{t+1})$. Therefore,

$$\begin{aligned}&\mathbb{E}_{y_{t+1:T}}[D_{\mathrm{KL}}(P_{\rightarrow}(y_t \mid y_{t+1:T}) \| P_{\rightarrow}(y_t \mid g^{\leftarrow}_{t+1}))] \\ &\quad = I_{P_{\rightarrow}}(y_t; y_{t+1:T} \mid g^{\leftarrow}_{t+1}) \equiv \mathcal{L}_t, \end{aligned} \tag{90}$$

where $I_{P_{\rightarrow}}(y_t; y_{t+1:T} \mid g^{\leftarrow}_{t+1})$ is the mutual information between y_t and $y_{t+1:T}$ under the condition of $g^{\leftarrow}_{t+1}$, with respect to the forward distribution $P_{\rightarrow}$. This is the information about y_t contained in the full future $y_{t+1:T}$ that is discarded by the compression from $y_{t+1:T}$ to $g^{\leftarrow}_{t+1}$.

Model mismatch. The logarithmic factor in the second term depends on $y_{t+1:T}$ only through $g^{\leftarrow}_{t+1}$. Hence,

$$\begin{aligned}&\sum_{y_{t+1:T}} P_{\rightarrow}(y_{t+1:T}) \sum_{y_t} P_{\rightarrow}(y_t \mid y_{t+1:T}) \ln \frac{P_{\rightarrow}(y_t \mid g^{\leftarrow}_{t+1})}{p_t(y_t \mid g^{\leftarrow}_{t+1})} \\ &\quad = \sum_{g^{\leftarrow}_{t+1}} P_{\rightarrow}(g^{\leftarrow}_{t+1}) \sum_{y_t} P_{\rightarrow}(y_t \mid g^{\leftarrow}_{t+1}) \ln \frac{P_{\rightarrow}(y_t \mid g^{\leftarrow}_{t+1})}{p_t(y_t \mid g^{\leftarrow}_{t+1})}. \end{aligned} \tag{91}$$

Therefore, the second term in Eq. (89) equals

$$\mathbb{E}_{g^{\leftarrow}_{t+1} \sim P_{\rightarrow}}[D_{\mathrm{KL}}(P_{\rightarrow}(y_t \mid g^{\leftarrow}_{t+1}) \| p_t(y_t \mid g^{\leftarrow}_{t+1}))] \equiv \mathcal{M}_t. \tag{92}$$

This represents the additional cost of using $p_t(\cdot \mid g^{\leftarrow}_{t+1})$ instead of the Bayesian retrospective distribution $P_{\rightarrow}(\cdot \mid g^{\leftarrow}_{t+1})$.

The decomposition. Combining Eqs. (90) and (92) gives

$$\mathcal{D}_t = \mathcal{L}_t + \mathcal{M}_t, \tag{93}$$

where $\mathcal{L}_t$ and $\mathcal{M}_t$ are non-negative.

The decomposition (93) is formally similar to the ELBO decompositions familiar from variational inference [52–54], where the average KL term in the ELBO likewise splits into a mutual-information term and a distributional-mismatch term. The shared algebraic origin is the chain rule for KL divergence applied via an intermediate distribution. In the present setting, however, the object being decomposed is not the KL term of an ELBO but a per-step contribution to the entropy production $\mathcal{S}_y$, defined through the path-probability ratio [Eq. (14)]. Accordingly, $\mathcal{L}_t$ [Eq. (90)] and $\mathcal{M}_t$ [Eq. (92)] acquire a physical interpretation as distinct sources of temporal irreversibility; the former from lossy compression of the future, the latter from reuse of the forward emission kernel. We note that the term of mismatch cost also appears in the stochastic thermodynamics of computation [32,34] with a related but distinct meaning, quantifying the extra dissipation due to a nonoptimal initial distribution of a physical process.

C. Refined second law

Combining the compression-loss decomposition with the chain rule for mutual information, we obtain a lower bound on the entropy production in terms of the gap between the predictive information carried by the forward and backward latent-state summaries.

Since $g^{\leftarrow}_{t+1}$ is a deterministic function of $y_{t+1:T}$, the chain rule for mutual information gives

$$\mathcal{L}_t = I_{P_{\rightarrow}}(y_t; y_{t+1:T}) - I_{P_{\rightarrow}}(y_t; g^{\leftarrow}_{t+1}). \tag{94}$$

On the forward side, since $f^{\rightarrow}_{t-1}(y_{1:t-1})$ is a sufficient summary of the past for predicting y_t,

$$I_{P_{\rightarrow}}(y_t; y_{1:t-1}) = I_{P_{\rightarrow}}(y_t; f^{\rightarrow}_{t-1}(y_{1:t-1})). \tag{95}$$

That is, the forward summary loses no predictive information by construction of the dynamics, whereas the backward summary generally does. We also note that the forward and reverse chain rules for conditional Shannon entropies, written as $H_{P_{\rightarrow}}(\cdot \mid \cdot)$, give

$$\sum_{t=1}^{T} H_{P_{\rightarrow}}(y_t \mid y_{1:t-1}) = H_{P_{\rightarrow}}(y_{1:T}) = \sum_{t=1}^{T} H_{P_{\rightarrow}}(y_t \mid y_{t+1:T}), \tag{96}$$

and thus

$$\sum_{t=1}^{T} I_{P_{\rightarrow}}(y_t; y_{1:t-1}) = \sum_{t=1}^{T} I_{P_{\rightarrow}}(y_t; y_{t+1:T}). \tag{97}$$

Combining all of the above, we obtain

$$\sum_{t=1}^{T} \mathcal{L}_t = \sum_{t=1}^{T} [I_{P_{\rightarrow}}(y_t; f^{\rightarrow}_{t-1}(y_{1:t-1})) - I_{P_{\rightarrow}}(y_t; g^{\leftarrow}_{t+1}(y_{T:t+1}))], \tag{98}$$

where the raw past and future are replaced by their deterministic summaries $f^{\rightarrow}_{t-1}$ and $g^{\leftarrow}_{t+1}$. Therefore,

$$\mathcal{S}_y \geqslant \sum_{t=1}^{T} [I_{P_{\rightarrow}}(y_t; f^{\rightarrow}_{t-1}(y_{1:t-1})) - I_{P_{\rightarrow}}(y_t; g^{\leftarrow}_{t+1}(y_{T:t+1}))] \geqslant 0. \tag{99}$$

This is regarded as a refined version of the second law; the entropy production is bounded from below by the gap between the mutual information terms associated with the past and the future. The more memory of the past the current state has lost, or the more the future summary retains of the information about the current state, the smaller the entropy production can be. In the special case where $g^{\leftarrow}_{t+1}$ is injective for all t, every $\mathcal{L}_t$ vanishes, so that $\mathcal{D}_t = \mathcal{M}_t$ and Eq. (99) reduces to $\mathcal{S}_y \geqslant 0$.

We note that Ref. [28] related dissipation to the predictive information retained by a driven system, but assumed that the system is Markovian and did not define the entropy production at the level of an observed non-Markovian output. We emphasize that their main result in Ref. [28] is distinct from our result (99) even in the Markovian case.

Another structurally relevant concept is the backward transfer entropy (BTE) [31], which considers a bipartite stochastic process (x_t, y_t) and defines the backward transfer entropy as the transfer entropy computed on time-reversed trajectories. Expressed in forward-time variables, the BTE between two variables x, y is defined as

$$T^{B}_{y \to x}(t) = I(y_{t+1:T}; x_t \mid x_{t+1:T}), \tag{100}$$

which quantifies how much the future of y retrodicts the present of x beyond what the future of x itself reveals. The compression-loss term of the present work, $\mathcal{L}_t$ in Eq. (90), shares a similar information-theoretic form. However, the BTE is an intervariable quantity: It measures the retrospective information that one subsystem (y) carries about a distinct

subsystem (x). In contrast, $\mathcal{L}_t$ is an intravariable quantity involving a single observed process y_t at different times; the conditioning variable $g^{\leftarrow}_{t+1}(y_{T:t+1})$ is a deterministic function of y's own future.

D. Estimation of retrospective decomposition terms

While $\mathcal{S}_y$ is directly computable by Monte Carlo sampling, the individual decomposition terms $\mathcal{D}_t$, $\mathcal{L}_t$, and $\mathcal{M}_t$ are not. The obstacle is the Bayesian retrospective distribution $P_{\rightarrow}(y_t \mid y_{t+1:T})$, which appears in the definition of $\mathcal{D}_t$ (88). This distribution is obtained by marginalizing the forward path measure over all possible pasts:

$$P_{\rightarrow}(y_t \mid y_{t+1:T}) = \frac{\sum_{y_1,\ldots,y_{t-1}} P_{\rightarrow}(y_{1:T})}{\sum_{y_1,\ldots,y_t} P_{\rightarrow}(y_{1:T})}. \tag{101}$$

Even though each factor $p_t(y_{t+1} \mid h_t)$ in the forward path probability is explicitly evaluable, the summations in the numerator and denominator couple all time steps through the deterministic maps Φ_t. Unlike the forward conditional $P_{\rightarrow}(y_{t+1} \mid y_{1:t}) = p_t(y_{t+1} \mid h_t)$, which the model provides by construction, the retrospective conditional $P_{\rightarrow}(y_t \mid y_{t+1:T})$ is not directly supplied by any component of the autoregressive architecture.

As discussed above, the Bayesian retrospective distribution $P_{\rightarrow}(y_t \mid y_{t+1:T})$ plays a role analogous to the reverse-time transition in diffusion models [76–79], where the intractable reverse-time transition is approximated by a neural network trained on samples from the forward process. This suggests a parallel strategy for the present setting: Training a reverse-direction autoregressive model [e.g., $r_\theta(y_t|y_{t+1:T})$ with model parameters θ] on trajectories sampled from the forward process to approximate $P_{\rightarrow}(y_t \mid y_{t+1:T})$, from which the individual terms $\mathcal{D}_t$, $\mathcal{L}_t$, and $\mathcal{M}_t$ could be estimated. Developing such estimation procedures remains an open problem.

VIII. SUMMARY AND PERSPECTIVES

We have developed a stochastic-thermodynamic framework for non-Markovian processes generated by autoregressive models with deterministic internal memory, encompassing Transformers, RNNs, the Kalman filter, state space models, and Mamba under a single formalism (Table I). The entropy production $\mathcal{S}_y$, defined as the KL divergence between the forward and backward path measures (14), is efficiently computable from sampled trajectories without combinatorial overhead from enumerating histories, owing to the deterministic latent state and the explicit emission kernel (Sec. IV); the same cost applies to the temporally coarse-grained variant that reverses blocks rather than individual tokens. We have demonstrated the framework through a proof-of-concept experiment on GPT-2 (Sec. V), where the sentence-level entropy production tends to be larger for causally ordered than for noncausal texts (with statistically significant differences for three of the five corpora with GPT-2, and for all five with a larger evaluator, Qwen3-4B-Base; Appendix C). Moreover, we have analyzed the linear Gaussian (Kalman) case (Sec. VI), where the analytical entropy production is confirmed by Monte Carlo sampling (Fig. 6). We have further shown that $\mathcal{S}_y$ decomposes exactly into non-negative per-step contributions $\mathcal{D}_t$ [Eq. (87)], each splitting into a compression loss $\mathcal{L}_t$ and a model mismatch $\mathcal{M}_t$ [Eq. (93)]. This decomposition is formally reminiscent of the ELBO decompositions for variational inference [52–54], which reveals an interesting connection between fundamental concepts in stochastic thermodynamics and machine learning.

From the viewpoint of stochastic thermodynamics, an interesting direction concerns finite-time trade-off relations. In Markovian stochastic thermodynamics, thermodynamic uncertainty relations [81,82] and thermodynamic speed limits [83] constrain the interplay among the precision of currents, the speed of state transformations, and the entropy production. Indeed, such trade-off relations have been demonstrated for diffusion models [84] and dense associative memory networks [37]. Extending these relations to the non-Markovian autoregressive setting could yield new bounds linking the speed, accuracy, and irreversibility of sequence generation in autoregressive generative models.

In standard implementations, LLMs often terminate generation upon producing an end-of-sequence (EOS) token, making the sequence length a stopping time. References [33,34] developed thermodynamic bounds at stochastic stopping times, which could be applied to generalize the present framework to variable-length processes. It would also be of interest to explore connections with computational mechanics [85–87], where the minimal sufficient statistic of the past (the causal state or ϵ machine) plays a role analogous to the forward latent state.

Applying our framework to larger and more capable language models than those examined here remains an important direction for future research and raises further challenges at different levels. At the technical level, a single semantic episode can often be expressed by multiple distinct token sequences (e.g., paraphrases), so that coarse-graining at the token-sequence level may not fully capture the irreversibility at the level of meaning; an additional coarse-graining over token sequences that encode the same semantic content may be needed. At a more fundamental level, the block-level entropy production reflects at least three intertwined contributions: genuine causal dependence among the described events, mere temporal ordering without causal relationship, and the conventions of discourse structure (e.g., narrative arc and rhetorical organization). Disentangling these contributions remains a highly nontrivial open problem. If these challenges can be addressed, the entropy production could provide a quantitative probe of irreversibility and causality of the real-world processes whose structure is implicitly encoded in the internal representations of an LLM—often referred to as a world model in the machine-learning literature.

ACKNOWLEDGMENTS

Claude Opus 4.6, Claude Fable 5, GPT-5.4 Thinking/Pro, and GPT-5.6 Sol were used for assistance with manuscript preparation. The author is grateful to Daisuke Okanohara, Ken Funo, Igen Koh, and David H. Wolpert for valuable discussions. The author is also grateful to Shoki Sugimoto for assistance with code review. This work was supported by JST ERATO Grant No. JPMJER2302, Japan, and also by Institute for AI and Beyond of the University of Tokyo.

DATA AVAILABILITY

The code and data used in this study are available in Ref. [88].

APPENDIX A: MARKOVIAN EMBEDDING

In this Appendix, we clarify the relationship between the autoregressive framework developed in the main text and the concept of Markovian embedding.

1. General definition

In this paper, we say that a process $\mathbf{x}_t$ provides a *Markovian embedding* of the observed non-Markovian process y_t if $\mathbf{x}_t$ is Markovian and the joint law factorizes as

$$P_{\rightarrow}(\mathbf{x}_{1:T}, y_{1:T}) = p(\mathbf{x}_1)\prod_{t=1}^{T-1} p_t(\mathbf{x}_{t+1} \mid \mathbf{x}_t)\prod_{t=1}^{T} q_t(y_t \mid \mathbf{x}_t). \quad \text{(A1)}$$

That is, y_t is emitted memorylessly from the Markovian state $\mathbf{x}_t$ at each time step. This definition can be graphically represented as follows, where $\Rightarrow$ and $\Downarrow$ describe stochastic influences:

$$\begin{array}{ccccccccc} \cdots & \mathbf{x}_{t-2} & \Rightarrow & \mathbf{x}_{t-1} & \Rightarrow & \mathbf{x}_t & \Rightarrow & \mathbf{x}_{t+1} & \cdots \\ & \Downarrow & & \Downarrow & & \Downarrow & & \Downarrow & \\ \cdots & y_{t-2} & & y_{t-1} & & y_t & & y_{t+1} & \cdots \end{array} \quad \text{(A2)}$$

Note that a more standard definition of Markovian embedding is a special case of the above definition, where y_t is obtained from $\mathbf{x}_t$ by a deterministic projection [89].

The corresponding backward process is defined as

$$P_{\leftarrow}(\tilde{\mathbf{x}}_{1:T}, \tilde{y}_{1:T}) = \tilde{p}(\tilde{\mathbf{x}}_1)\prod_{s=1}^{T-1} \tilde{p}_s(\tilde{\mathbf{x}}_{s+1} \mid \tilde{\mathbf{x}}_s)\prod_{s=1}^{T} \tilde{q}_s(\tilde{y}_s \mid \tilde{\mathbf{x}}_s), \quad \text{(A3)}$$

where we choose $\tilde{p}_s = p_{T-s}$ and $\tilde{q}_s = q_{T-s+1}$ to be consistent with Crooks' notion of the backward process [4]. Here, we assume that the backward y-marginal coincides with the distribution defined in Sec. III A, but we do not assume Eq. (10).

We define the $\mathbf{x}$ marginal and y marginal of these path distributions, and write $P_{\rightarrow}(\mathbf{x}_{1:T})$, $P_{\leftarrow}(\mathbf{x}_{T:1})$, $P_{\rightarrow}(y_{1:T})$, and $P_{\leftarrow}(y_{T:1})$ by substituting $\tilde{\mathbf{x}}_{1:T} = \mathbf{x}_{T:1}$ and $\tilde{y}_{1:T} = y_{T:1}$ for the backward process. We then introduce the entropy production associated with $\mathbf{x}$,

$$\mathcal{S}_{\mathbf{x}} \equiv D_{\mathrm{KL}}(P_{\rightarrow}(\mathbf{x}_{1:T}) \| P_{\leftarrow}(\mathbf{x}_{T:1})), \quad \text{(A4)}$$

and with y,

$$\mathcal{S}_y \equiv D_{\mathrm{KL}}(P_{\rightarrow}(y_{1:T}) \| P_{\leftarrow}(y_{T:1})). \quad \text{(A5)}$$

Here, the product channel $\mathbf{x}_{1:T} \mapsto y_{1:T}$ defined by $\prod_{t=1}^{T} q_t(y_t \mid \mathbf{x}_t) = \prod_{s=1}^{T} \tilde{q}_s(y_{T-s+1} \mid \mathbf{x}_{T-s+1})$ is a stochastic map ("Markovian" map) from $\mathbf{x}$ paths to y paths. Applying it to $P_{\rightarrow}(\mathbf{x}_{1:T})$ yields $P_{\rightarrow}(y_{1:T})$, and applying it to $P_{\leftarrow}(\mathbf{x}_{T:1})$ yields $P_{\leftarrow}(y_{T:1})$. The monotonicity of KL divergence under Markovian maps (data-processing inequality) [72] therefore gives

$$\mathcal{S}_{\mathbf{x}} \geqslant \mathcal{S}_y \geqslant 0. \quad \text{(A6)}$$

This is the coarse-graining inequality familiar from the stochastic thermodynamics literature [6–11]: The entropy production of the Markovian process $\mathbf{x}_t$ is at least as large as the irreversibility detectable from the observation y_t alone.

2. Relation to the present framework

We examine how the autoregressive framework of the main text relates to the Markovian embedding defined above. First, remember that in the general setting including Transformers, $h_t = \Phi_t(y_1, \ldots, y_t)$ does not factor through a two-argument recursion, and thus even (h_t, y_t) is not Markovian in general.

In the recursive case discussed in Sec. II B with $h_t = \phi(h_{t-1}, y_t)$, the joint process (h_t, y_t) is Markovian. Therefore, $\mathbf{x}_t \equiv (h_t, y_t)$ is a Markovian embedding of y_t. (Note that h_t alone does not constitute a Markovian embedding of y_t in general.)

In this case, the forward path probability $P_{\rightarrow}(y_{1:T})$ defined in Sec. II A coincides with the y marginal of the joint forward process (A1). Explicitly, we set

$$p_t(\mathbf{x}_{t+1} \mid \mathbf{x}_t) = p_t(y_{t+1} \mid h_t)\delta(h_{t+1} - \phi(h_t, y_{t+1})); \quad \text{(A7)}$$

the recursive integration like

$$\begin{aligned} &\int dh_{t+1} dh_t\, p_t(y_{t+1}|h_t)\delta(h_{t+1} - \phi(h_t, y_{t+1})) \\ &\qquad \times p_{t-1}(y_t|h_{t-1})\delta(h_t - \phi(h_{t-1}, y_t)) \\ &= \int dh_t\, p_t(y_{t+1}|h_t)p_{t-1}(y_t|h_{t-1})\delta(h_t - \phi(h_{t-1}, y_t)) \\ &= p_t(y_{t+1}|\phi(h_{t-1}, y_t))p_{t-1}(y_t|h_{t-1}) \end{aligned} \quad \text{(A8)}$$

confirms that it produces $P_{\rightarrow}(y_{1:T})$ defined in Sec. II A. Moreover, the backward path probability $P_{\leftarrow}(y_{T:1})$ defined in Eq. (A3) above and that in Sec. III A coincide, if we choose the appropriate boundary term

$$\tilde{p}(\tilde{\mathbf{x}}_1) = \tilde{p}(\tilde{h}_1, \tilde{y}_1) \equiv \tilde{p}_0(\tilde{y}_1|\tilde{h}_0)\delta(\tilde{h}_1 - \phi(\tilde{h}_0, \tilde{y}_1)), \quad \text{(A9)}$$

where $\tilde{h}_0 = h_0$ is a fixed constant as in our general framework. Indeed, by letting $\tilde{y}_s = y_{T-s+1}$, a similar recursive integration gives

$$\begin{aligned} &\int d\tilde{h}_{s+1} d\tilde{h}_s p_{T-s}(y_{T-s}|\tilde{h}_s)\delta(\tilde{h}_{s+1} - \phi(\tilde{h}_s, y_{T-s})) \\ &\qquad \times p_{T-s+1}(y_{T-s+1}|\tilde{h}_{s-1})\delta(\tilde{h}_s - \phi(\tilde{h}_{s-1}, y_{T-s+1})) \\ &= \int d\tilde{h}_s p_{T-s}(y_{T-s}|\tilde{h}_s)p_{T-s+1}(y_{T-s+1}|\tilde{h}_{s-1}) \\ &\qquad \times \delta(\tilde{h}_s - \phi(\tilde{h}_{s-1}, y_{T-s+1})) \\ &= p_{T-s}(y_{T-s}|\phi(\tilde{h}_{s-1}, y_{T-s+1}))p_{T-s+1}(y_{T-s+1}|\tilde{h}_{s-1}). \end{aligned} \quad \text{(A10)}$$

Therefore, $\mathcal{S}_y$ defined above coincides with our general definition of the entropy production (14).

On the other hand, as noted in Sec. III, applying the deterministic maps ϕ backward to the time-reversed sequence $y_{T:1}$ does not reproduce the reversed sequence of $h_{1:T}$ in general; $\tilde{h}_s = h_{T-s+1}$ does not necessarily hold. Explicitly, if we substitute $\tilde{\mathbf{x}}_{1:T} = \mathbf{x}_{T:1}$ into the backward path probability (A3), each transition kernel becomes

$$p_t(\mathbf{x}_t \mid \mathbf{x}_{t+1}) = p_t(y_t \mid h_{t+1})\,\delta(h_t - \phi(h_{t+1}, y_t)). \quad \text{(A11)}$$

In general, the arguments of the delta functions in Eqs. (A7) and (A11) cannot be zero at the same time. As a consequence, $\tilde{\mathbf{x}}_{1:T} = \mathbf{x}_{T:1}$ is not realized in general, implying that the deterministic part of the dynamics may be completely irreversible. Therefore, $\mathcal{S}_{\mathbf{x}}$ defined in Eq. (A4) may diverge and would not be informative. This consideration provides another basis for our approach to the non-Markovian entropy production, which is defined solely by the path probabilities of $y_{1:T}$ without embedding into another Markovian dynamics.

3. True environmental state as a possible Markovian embedding

Behind the observations y_t there may exist a true environmental state x_t whose dynamics generates y_t. If the joint process (x_t, y_t) satisfies the factorization (A1) with $\mathbf{x}_t = x_t$, then x_t constitutes a Markovian embedding of y_t. The Kalman filter example (Sec. VI) is a concrete instance: x_t evolves as $x_{t+1} = Ax_t + w_t$ and produces observations $y_t = Cx_t + v_t$, which is exactly of the form (A1).

In the present general framework, however, a true environmental state x_t is not explicitly assumed. The latent state $h_t = \Phi_t(y_1, \ldots, y_t)$ is a deterministic function of the observations and carries no independent stochastic degrees of freedom; it is not an environmental state but a computational latent state.

APPENDIX B: DETAILS OF THE GPT-2 EXPERIMENT

We describe the details of the GPT-2 experiment in Sec. V and show some supplemental results. Our implementation uses the Hugging Face `transformers` library [90], which provides pretrained weights and a tokenizer for GPT-2.

The Hugging Face implementation may differ from the original OpenAI release [39] in default generation parameters and tokenizer behavior; the details relevant to our experiment are described in the subsections below. Note that the original paper [39] reported 117M parameters for GPT-2; OpenAI later noted that this count was erroneous [91], and `model.parameters()` in Hugging Face yields approximately 124M, as in our metadata.

All GPT-2 sampling and likelihood-evaluation runs were performed on NVIDIA T4 or V100 graphics processing units (GPUs). Exact token-by-token reproducibility of sampled trajectories is not guaranteed across different hardware/software environments, even with a fixed random seed.

1. Details of sampling from GPT-2

For the sampling experiment from GPT-2 itself (Sec. V A), the path probability in Eq. (28) contains the boundary term $p(y_1 \mid h_0)$. This term must be specified separately, because the tokenizer used in our implementation does not prepend an initial beginning-of-sequence (BOS) token automatically. In the Hugging Face GPT-2 tokenizer, the EOS token and the BOS token are both assigned to the same token `<|endoftext|>`, and the tokenizer backend adds the BOS token only when explicitly requested [92,93].

Accordingly, our implementation explicitly prepends `<|endoftext|>` to the tokenized sequence before each forward pass: The model receives [`<|endoftext|>`, $y_1, \ldots, y_T$]. Therefore,

$$P(y_{1:T}) = \prod_{t=0}^{T-1} p(y_{t+1} \mid \texttt{<|endoftext|>}, y_1, \ldots, y_t); \quad \text{(B1)}$$

the map $\Phi(y_1, \ldots, y_t)$ corresponds to the latent state after processing [`<|endoftext|>`, $y_1, \ldots, y_t$]. Note that GPT-2 uses absolute position embeddings. For $t = 0$, we adopt the initial condition

$$p(y_1 \mid h_0) \equiv p(y_1 \mid \texttt{<|endoftext|>}). \quad \text{(B2)}$$

Since `<|endoftext|>` is always fixed, the above construction exactly fits our general theoretical framework.

Hugging Face has a default function `model.generate()` for generating tokens. In our experiment, to make the construction more explicit, we replace `model.generate()` with an explicit autoregressive loop that uses the KV cache of GPT-2 directly. At each step t, the logits are computed from the cached latent state, the next token y_{t+1} is drawn from $p(\cdot \mid \texttt{<|endoftext|>}, y_1, \ldots, y_t)$, and the log-likelihood $\ln p(y_{t+1} \mid \texttt{<|endoftext|>}, y_1, \ldots, y_t)$ is recorded from the same logits. Therefore, in our implementation, the forward log-likelihood is taken from exactly the same logits within each run.

Meanwhile, Hugging Face's `model.generate()` terminates decoding upon emitting the EOS token `<|endoftext|>`. On the other hand, our above implementation always executes exactly T steps of token generation, where the EOS token may appear within the generated sequence but does not trigger termination.

For the reversed sequence, we use [`<|endoftext|>`, $y_T, y_{T-1}, \ldots, y_1$] to calculate the backward log-likelihood, so that the boundary term is treated identically in both directions. That is, the same fixed initial context is used for $y_{1:T}$ and for $y_{T:1}$, and the assumption $\tilde{h}_0 = h_0$ is satisfied. The reverse log-likelihood $\ln P(y_T, y_{T-1}, \ldots, y_1)$ is obtained by feeding the reversed sequence [`<|endoftext|>`, $y_T, y_{T-1}, \ldots, y_1$] to the model by using the same protocol as the forward process with the KV cache.

We do not need any stronger statement about the unpublished details of the original training-data pipeline. For the numerical experiment, the only required point is that the rule above gives a concrete and reproducible definition of the boundary term for the publicly released model.

Note that, in the fixed-text experiment of Sec. V B where no sampling is performed, the log-likelihood of each sequence is evaluated by a single full-sequence forward pass of GPT-2 (one pass per sequence), rather than by the KV-cache incremental path used in Sec. V A.

2. Convergence of entropy production and fluctuation theorem

We next show supplemental numerical results for the Monte Carlo sampling from GPT-2.

To examine how the Monte Carlo estimates stabilize as the sample size grows, Fig. 7 plots the cumulative sample mean of σ_{token}/T and σ_{block}/T' for $T = 120$, as a function of N. The shaded bands indicate 95% confidence intervals obtained by the bootstrap percentile method ($B = 2000$ resamples at each

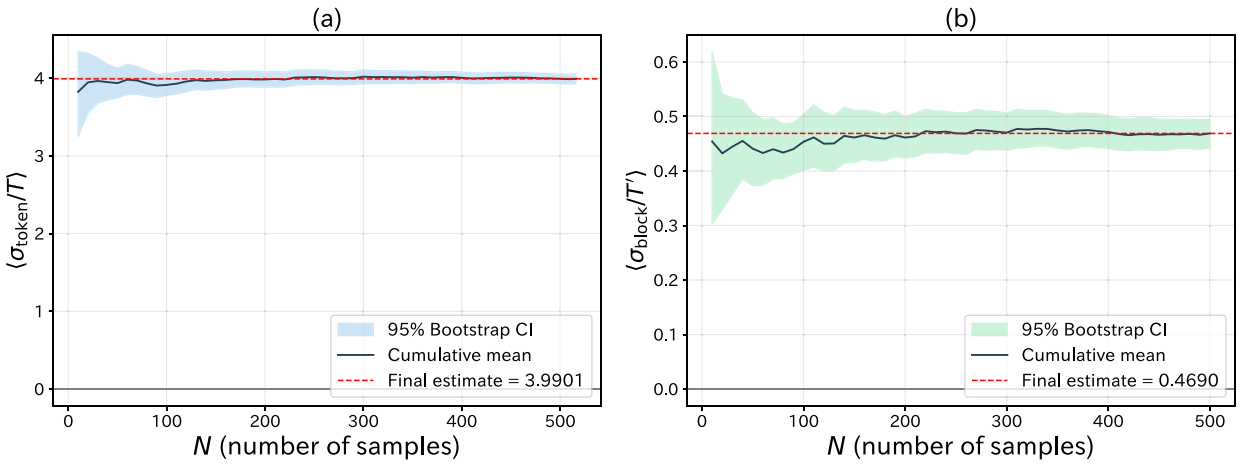


FIG. 7. Convergence of the Monte Carlo estimates of the per-token entropy production for $T = 120$ in GPT-2, as a function of the number of samples N. The temperature parameter is $\tau = 1$. (a) Token-level reversal σ_{token}/T; (b) block-level reversal σ_{block}/T'. Solid lines show the cumulative sample mean; shaded regions are 95% bootstrap percentile confidence intervals ($B = 2000$). Dashed red lines indicate the final estimates at $N = 516$ or 500 (see the caption of Fig. 3).

N), which does not assume normality of the sampling distribution. Both estimators converge to positive values, consistent with the positivity of the entropy production.

We also perform a numerical verification of the fluctuation theorem $\mathbb{E}_P[e^{-\sigma_{\text{token}}}] = 1$, where σ_{token} is the stochastic entropy production with token-level reversal without truncation, and the samples are drawn from the forward distribution of GPT-2. Because σ_{token} grows with T, choosing a large sequence length makes $e^{-\sigma_{\text{token}}}$ extremely small for typical $\sigma_{\text{token}} > 0$, resulting in intractably slow convergence of the exponential average. We therefore set $T = 5$ in this experiment. Similarly, since the convergence is poor at $\tau = 1$, we instead adopt $\tau = 3$ and $\tau = 4$, for which the cumulative sample mean roughly converges toward the theoretical value of 1 as the number of samples N increases. Figure 8 shows this convergence behavior for $N = 5000$ samples at each temperature. While the fluctuation theorem directly follows from the definition of the entropy production, our numerical result serves as a consistency check for our sampling method.

3. Dependence on the block scale

We further examine how the entropy production depends on the scale of the block reversal, using the same $N = 500$ truncated sequences $y_{1:T'}$ as in Fig. 3. This analysis reuses the saved trajectories; no new text is sampled. For each original or reordered sequence, the log-likelihood is reevaluated with GPT-2.

We consider two block constructions. First, for the sentence-based construction, sentence boundaries are reconstructed using the same sentence-final punctuation-token rule as in Fig. 3(b). Consecutive groups of k sentences are treated

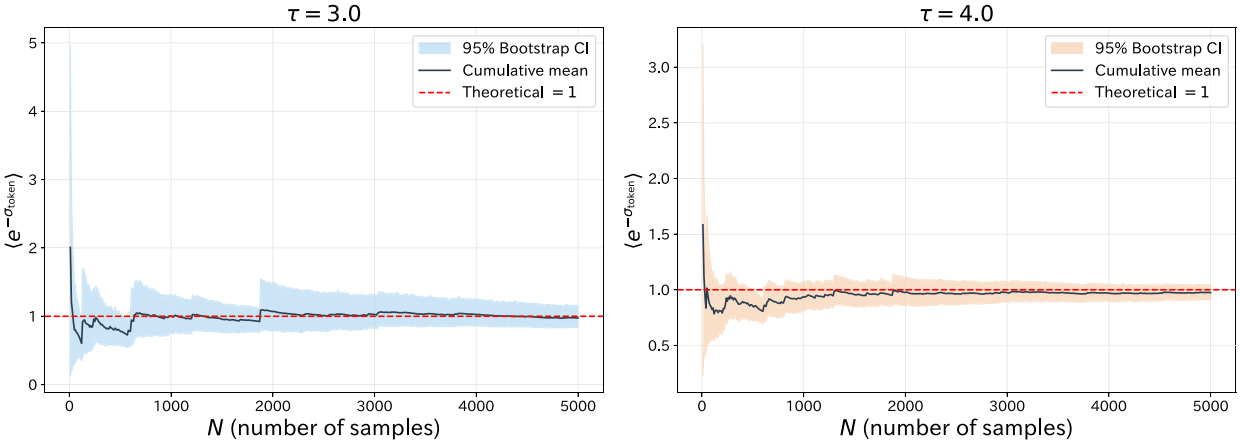


FIG. 8. Convergence of the Monte Carlo mean of $e^{-\sigma_{\text{token}}}$ as a function of the number of samples N, for sequences of $T = 5$ tokens sampled from GPT-2. Temperatures are $\tau = 3$ and $\tau = 4$. Solid lines show the cumulative sample mean; shaded regions are 95% bootstrap percentile confidence intervals ($B = 2000$). The dashed red line indicates the theoretical prediction $\mathbb{E}_P[e^{-\sigma_{\text{token}}}] = 1$, namely, the integral fluctuation theorem. In both panels, the cumulative mean trends toward the theoretical value 1, as N grows.

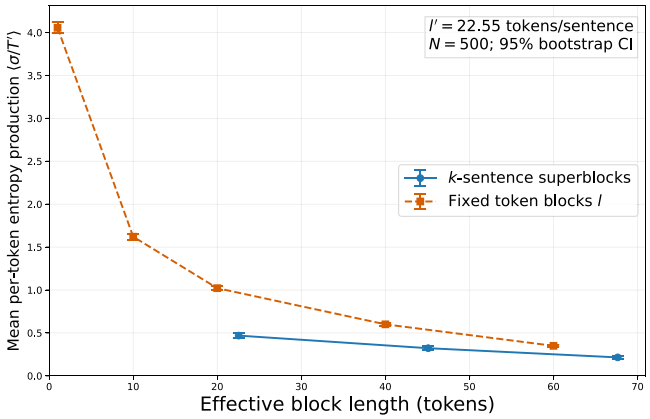


FIG. 9. Dependence of the mean per-token entropy production on the block scale for the $N = 500$ GPT-2 samples used in Fig. 3. The blue circles show reversals of superblocks formed from k consecutive sentences and are plotted at kl', with $l' = 22.55$ tokens per sentence; the plotted values are $k = 1, 2, 3$. The orange squares show reversals of fixed-length blocks with block length $l = 1, 10, 20, 40, 60$ and are plotted at l. Each point is the arithmetic mean of the per-sample values divided by T', and error bars show 95% percentile bootstrap confidence intervals ($B = 2000$). Lines are guides to the eye.

as superblocks and the superblock order is reversed, while the order of sentences and tokens within each superblock is retained. The final superblock is allowed to contain fewer than k sentences. The blue circles in Fig. 9 show the mean per-token entropy production for this sentence-based construction. The horizontal axis is kl', where $l' = 22.55$ is the mean sentence length. To focus on block scales up to approximately 60–70 tokens, we used $k = 1, 2, 3$. The case $k = 1$ corresponds to the sentence-block reversal used in Fig. 3(b).

Second, we consider the fixed-length construction discussed in Sec. IV D, where each sequence was split from its beginning into consecutive blocks of l tokens with l being fixed. Note that the final block was allowed to be shorter than l. The orange squares of Fig. 9 show the mean per-token entropy production for this fixed-length construction. We use $l = 1, 10, 20, 40, 60$, and plot each result at l on the horizontal axis. In particular, $l = 1$ corresponds to the token-level reversal of the same truncated sequences $y_{1:T'}$.

As seen from Fig. 9, the mean entropy production decreases over the plotted block-scale range for both constructions. The entropy production for the fixed-length construction is always greater than that for the sentence-based construction. This is consistent with the observation that the former contains syntactic artifacts, because block boundaries do not coincide with sentence boundaries.

4. Input text sets and statistical analysis

For the fixed-text experiment in Sec. V B, we used five language models as input-text generators: Claude Opus 4.6, Claude Fable 5, Gemini 3.1 Pro, GPT-5.4 Pro, and GPT-5.6 Pro. The same fixed prompt was submitted to each generator. For every generator, the resulting corpus contains 100 causal and 100 noncausal English texts, with four sentences in each text. The causal texts are short narratives in which the sentences describe a temporally ordered sequence of events, whereas the noncausal texts consist of independent statements whose ordering has no intended temporal or causal implication. These categories are defined operationally by the prompt and are not based on a formal criterion for causal dependence. The complete prompts, text sets, analysis code, and raw numerical results are provided in the GitHub repository (see the Data Availability statement).

The prompt contains one causal example and one noncausal example, both of which are required to appear as fixed entries in the generated lists. Consequently, these two texts are shared across all five corpora, while the remaining 99 causal and 99 noncausal texts are generator specific. The shared examples are the two texts displayed in Sec. V B. When a generator's output violated the required format (e.g., invalid JSON or an incorrect number of texts or sentences), the generation was repeated with the same fixed prompt until the format constraints were satisfied; the resulting texts were used verbatim, with no editing and no selection based on their content. Note that three of the 100 causal texts generated by Gemini 3.1 Pro contain "then" or "later" despite the prompt's prohibition of such words. We note that each corpus was obtained from one model response, and therefore, strictly

TABLE II. Statistical comparison of the GPT-2 experiment with causal and noncausal fixed-text samples discussed in Sec. V B. For every row, $n_C = 100$ and $n_{NC} = 100$, and all observations are included without outlier removal. U_C is the Mann-Whitney statistic for the causal sample, and $r = 2U_C/(n_C n_{NC}) - 1$ is the rank-biserial correlation; $r > 0$ indicates that the causal values tend to be larger than the noncausal values. The reported p values are from the individual two-sided asymptotic Mann-Whitney test.

Input-text generator	Reversal	U_C	r	p
Claude Opus 4.6	Token	3282	−0.3436	2.710×10^{-5}
Claude Opus 4.6	Block	7286	0.4572	2.346×10^{-8}
Claude Fable 5	Token	3984	−0.2032	0.013 09
Claude Fable 5	Block	6356	0.2712	9.263×10^{-4}
Gemini 3.1 Pro	Token	5507	0.1014	0.2159
Gemini 3.1 Pro	Block	6312	0.2624	0.001 353
GPT-5.4 Pro	Token	5389	0.0778	0.3425
GPT-5.4 Pro	Block	5467	0.0934	0.2544
GPT-5.6 Pro	Token	2687	−0.4626	1.601×10^{-8}
GPT-5.6 Pro	Block	5468	0.0936	0.2533

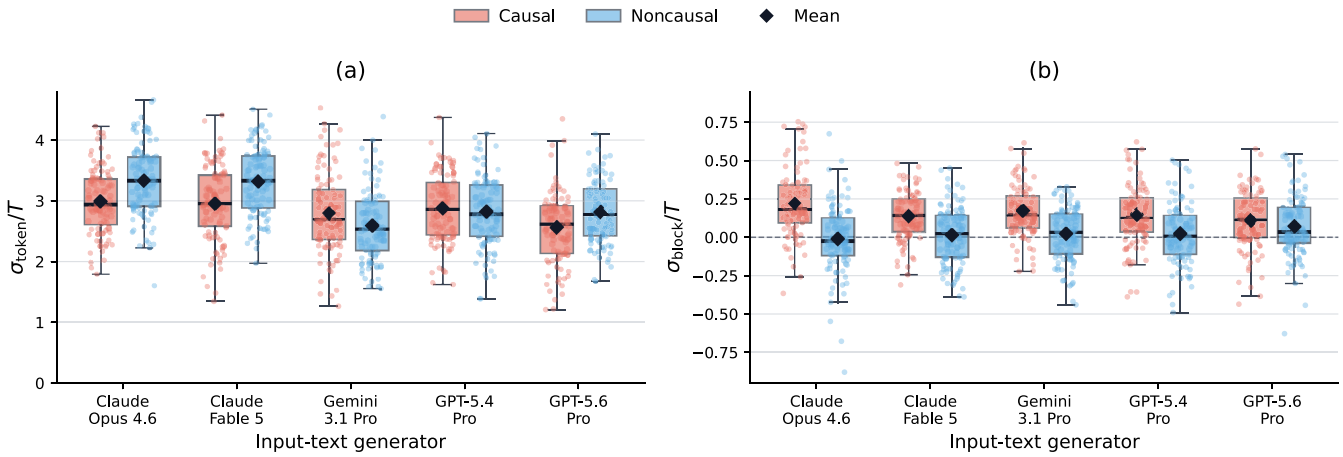


FIG. 10. Distributions of the stochastic entropy production evaluated using Qwen3-4B-Base for the fixed texts generated by the five input-text models. (a) Token-level reversal σ_{token}/T. (b) Sentence-block reversal σ_{block}/T. The plotting conventions are the same as those in Fig. 4.

speaking, the texts within a corpus are not guaranteed to be statistically independent.

All likelihoods are evaluated using GPT-2 at temperature $\tau = 1$. To preserve interblock whitespace when sentence blocks are permuted at the token-ID level, each fixed text is tokenized with the GPT-2 tokenizer using `add_prefix_space=True`; this encodes the first word in the same space-prefixed form as words preceded by whitespace elsewhere in the text. As in Appendix B 1, every original or reversed sequence is evaluated with the fixed `<|endoftext|>` token as its initial context.

For each input-text generator and reversal scheme, we compare the causal and noncausal samples using a two-sided asymptotic Mann-Whitney U test with the standard tie correction to the variance and with continuity correction. The direction and magnitude of each difference are quantified by the rank-biserial correlation r defined in Sec. V B. Pointwise 95% confidence intervals for r are obtained using the percentile bootstrap with independent resampling within the causal and noncausal categories. The summary of these test statistics is shown in Table II. All reported p values are individual and unadjusted for multiplicity.

APPENDIX C: FIXED-TEXT ANALYSIS WITH QWEN3-4B-BASE

In this Appendix, we examine whether the fixed-text results of Sec. V B (Appendix B 4) depend on the relatively small GPT-2 evaluator, by repeating the entire analysis with a substantially larger evaluator, Qwen3-4B-Base [94,95]. Because the input corpora are unchanged, this analysis tests robustness across evaluator models and tokenizers.

Qwen3-4B-Base is an open-weight, pretrained language model with 4.0 billion parameters. We used the base model rather than a posttrained instruction or chat model. Accordingly, no chat template, system prompt, or generation prompt was applied. All likelihood-evaluation runs were performed on NVIDIA V100 GPUs.

The evaluator was applied separately to the same five fixed-text corpora used in Sec. V B (Appendix B 4). For likelihood evaluation, one ASCII space was prepended when a string did not already begin with a space, and the resulting string was tokenized by the Qwen3 tokenizer with `add_special_tokens=False`. This explicit leading-space rule plays the role of `add_prefix_space=True` in the GPT-2 implementation and preserves the whitespace attached

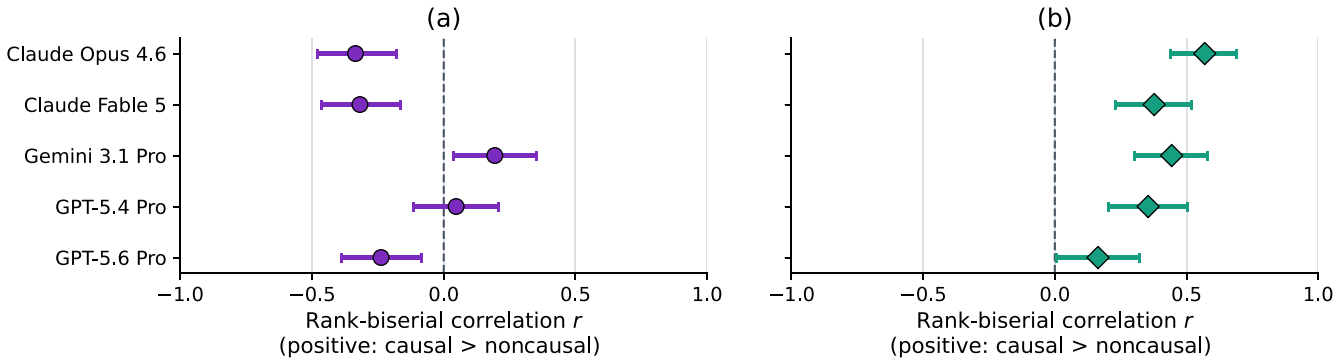


FIG. 11. Cross-generator comparison of the causal/noncausal difference in the per-token stochastic entropy production evaluated by Qwen3-4B-Base. The labels on the vertical axis indicate the language model used to generate each input-text corpus. (a) Token-level reversal σ_{token}/T. (b) Sentence-block reversal σ_{block}/T. The plotting conventions and statistical annotations are the same as those in Fig. 5.

TABLE III. Statistical comparison of the Qwen3-4B-Base experiment with causal and noncausal fixed-text samples. The statistical annotations are the same as those in Table II.

Input-text generator	Reversal	U_C	r	p
Claude Opus 4.6	Token	3324	−0.3352	4.242×10^{-5}
Claude Opus 4.6	Block	7846	0.5692	3.585×10^{-12}
Claude Fable 5	Token	3409	−0.3182	1.018×10^{-4}
Claude Fable 5	Block	6886	0.3772	4.085×10^{-6}
Gemini 3.1 Pro	Token	5969	0.1938	0.017 96
Gemini 3.1 Pro	Block	7219	0.4438	5.938×10^{-8}
GPT-5.4 Pro	Token	5235	0.0470	0.5667
GPT-5.4 Pro	Block	6771	0.3542	1.518×10^{-5}
GPT-5.6 Pro	Token	3810	−0.2380	0.003 656
GPT-5.6 Pro	Block	5820	0.1640	0.045 25

to sentence blocks when their token-ID sequences are permuted. No EOS target was appended; T counts only the tokens obtained from the fixed input text. Reversal was performed only after tokenization; reversed strings were not retokenized.

As in the GPT-2 analysis, we used an explicit fixed initial-context token, although the relevant tokenizer metadata differs. The Qwen3 tokenizer does not automatically prepend a BOS token; we explicitly chose `model.config.bos_token_id`, namely, token ID 151643 (`<|endoftext|>`), as the common fixed initial-context token for the original and reversed paths. The official model configuration designates this token as both BOS and EOS, and it coincides with the tokenizer's EOS token. Thus, the resulting boundary term parallels the `<|endoftext|>` convention of the GPT-2 implementation (Appendix B 1).

For each text, the forward log-likelihood was evaluated once and reused in the two reversal analyses. Likelihood evaluation used `torch.inference_mode()` and `use_cache=False`. We report $\sigma_{\mathrm{token}}/T$ and $\sigma_{\mathrm{block}}/T$, where T is the number of tokens assigned by the Qwen3 tokenizer. The GPT-2 and Qwen3 tokenizers generally produce different token sequences and different values of T.

The resulting distributions are shown in Fig. 10, and the corresponding rank-biserial correlations are shown in Fig. 11. The summary of the test statistics is reported in Table III.

The qualitative pattern closely reproduces the GPT-2 result of Sec. V B, and the statistical resolution improves. For token reversal, the causal/noncausal difference again depends strongly on the input-text generator, with significant differences now occurring in both directions: significantly smaller $\sigma_{\mathrm{token}}/T$ for the causal texts of Claude Opus 4.6, Claude Fable 5, and GPT-5.6 Pro, significantly larger for those of Gemini 3.1 Pro, and no significant difference for GPT-5.4 Pro. Thus, even with the substantially larger Qwen3 evaluator, the token-level direction remains a generator-specific property of the text corpora.

For block reversal, in contrast, the causal texts show significantly larger $\sigma_{\mathrm{block}}/T$ for all five input-text generators, including GPT-5.4 Pro and GPT-5.6 Pro, for which no significant difference was resolved with GPT-2. The same direction is therefore recovered across the five input-text generators and the two evaluators, despite the large difference in model size and the use of different tokenizers. This provides additional support for the conclusion of Sec. V B that block reversal is the more suitable probe of intersentence ordering. As before, however, it does not by itself establish $\sigma_{\mathrm{block}}/T$ as a general quantitative measure of causal structure.

[1] U. Seifert, Stochastic thermodynamics, fluctuation theorems and molecular machines, Rep. Prog. Phys. **75**, 126001 (2012).

[2] L. Peliti and S. Pigolotti, *Stochastic Thermodynamics: An Introduction* (Princeton University Press, Princeton, 2021).

[3] C. Jarzynski, Nonequilibrium equality for free energy differences, Phys. Rev. Lett. **78**, 2690 (1997).

[4] G. E. Crooks, Entropy production fluctuation theorem and the nonequilibrium work relation for free energy differences, Phys. Rev. E **60**, 2721 (1999).

[5] U. Seifert, Entropy production along a stochastic trajectory and an integral fluctuation theorem, Phys. Rev. Lett. **95**, 040602 (2005).

[6] A. Gomez-Marin, J. M. R. Parrondo, and C. Van den Broeck, Lower bounds on dissipation upon coarse graining, Phys. Rev. E **78**, 011107 (2008).

[7] É. Roldán and J. M. R. Parrondo, Estimating dissipation from single stationary trajectories, Phys. Rev. Lett. **105**, 150607 (2010).

[8] É. Roldán and J. M. R. Parrondo, Entropy production and Kullback–Leibler divergence between stationary trajectories of discrete systems, Phys. Rev. E **85**, 031129 (2012).

[9] K. Kawaguchi and Y. Nakayama, Fluctuation theorem for hidden entropy production, Phys. Rev. E **88**, 022147 (2013).

[10] M. Kahlen and J. Ehrich, Hidden slow degrees of freedom and fluctuation theorems: An analytically solvable model, J. Stat. Mech. (2018) 063204.

[11] G. E. Crooks and S. E. Still, Marginal and conditional second laws of thermodynamics, Europhys. Lett. **125**, 40005 (2019).

[12] D. S. Seara, B. B. Machta, and M. P. Murrell, Irreversibility in dynamical phases and transitions, Nat. Commun. **12**, 392 (2021).

[13] T. Speck and U. Seifert, The Jarzynski relation, fluctuation theorems, and stochastic thermodynamics for non-Markovian processes, J. Stat. Mech. (2007) L09002.

[14] T. Ohkuma and T. Ohta, Fluctuation theorems for non-linear generalized Langevin systems, J. Stat. Mech. (2007) P10010.

[15] S. Rahav and C. Jarzynski, Fluctuation relations and coarse-graining, J. Stat. Mech. (2007) P09012.

[16] A. Puglisi, S. Pigolotti, L. Rondoni, and A. Vulpiani, Entropy production and coarse graining in Markov processes, J. Stat. Mech. (2010) P05015.

[17] M. Esposito, Stochastic thermodynamics under coarse graining, Phys. Rev. E **85**, 041125 (2012).

[18] T. Munakata and M. L. Rosinberg, Entropy production and fluctuation theorems for Langevin processes under continuous non-Markovian feedback control, Phys. Rev. Lett. **112**, 180601 (2014).

[19] M. L. Rosinberg, T. Munakata, and G. Tarjus, Stochastic thermodynamics of Langevin systems under time-delayed feedback control: Second-law-like inequalities, Phys. Rev. E **91**, 042114 (2015).

[20] M. Polettini and M. Esposito, Effective thermodynamics for a marginal observer, Phys. Rev. Lett. **119**, 240601 (2017).

[21] J. van der Meer, J. Degünther, and U. Seifert, Time-resolved statistics of snippets as general framework for model-free entropy estimators, Phys. Rev. Lett. **130**, 257101 (2023).

[22] J. Degünther, J. van der Meer, and U. Seifert, Fluctuating entropy production on the coarse-grained level: Inference and localization of irreversibility, Phys. Rev. Res. **6**, 023175 (2024).

[23] K. Kanazawa and A. Dechant, Stochastic thermodynamics for classical non-Markov jump processes, arXiv:2506.04726.

[24] J. M. R. Parrondo, J. M. Horowitz, and T. Sagawa, Thermodynamics of information, Nat. Phys. **11**, 131 (2015).

[25] T. Sagawa and M. Ueda, Generalized Jarzynski equality under nonequilibrium feedback control, Phys. Rev. Lett. **104**, 090602 (2010).

[26] T. Sagawa and M. Ueda, Fluctuation theorem with information exchange: Role of correlations in stochastic thermodynamics, Phys. Rev. Lett. **109**, 180602 (2012).

[27] T. Sagawa and M. Ueda, Nonequilibrium thermodynamics of feedback control, Phys. Rev. E **85**, 021104 (2012).

[28] S. Still, D. A. Sivak, A. J. Bell, and G. E. Crooks, Thermodynamics of prediction, Phys. Rev. Lett. **109**, 120604 (2012).

[29] S. Ito and T. Sagawa, Information thermodynamics on causal networks, Phys. Rev. Lett. **111**, 180603 (2013).

[30] J. M. Horowitz and M. Esposito, Thermodynamics with continuous information flow, Phys. Rev. X **4**, 031015 (2014).

[31] S. Ito, Backward transfer entropy: Informational measure for detecting hidden Markov models and its interpretations in thermodynamics, gambling and causality, Sci. Rep. **6**, 36831 (2016).

[32] D. H. Wolpert and A. Kolchinsky, Thermodynamics of computing with circuits, New J. Phys. **22**, 063047 (2020).

[33] G. Manzano, D. Subero, O. Maillet, R. Fazio, J. P. Pekola, and É. Roldán, Thermodynamics of gambling demons, Phys. Rev. Lett. **126**, 080603 (2021).

[34] G. Manzano, G. Kardeş, É. Roldán, and D. H. Wolpert, Thermodynamics of computations with absolute irreversibility, unidirectional transitions, and stochastic computation times, Phys. Rev. X **14**, 021026 (2024).

[35] D. H. Wolpert, J. Korbel, C. W. Lynn, F. Tasnim, J. A. Grochow, G. Kardeş, J. B. Aimone, V. Balasubramanian, E. De Giuli, D. Doty, N. Freitas, M. Marsili, T. E. Ouldridge, A. W. Richa, P. Riechers, É. Roldán, B. Rubenstein, Z. Toroczkai, and J. Paradiso, Is stochastic thermodynamics the key to understanding the energy costs of computation?, Proc. Natl. Acad. Sci. USA **121**, e2321112121 (2024).

[36] S. Goldt and U. Seifert, Stochastic thermodynamics of learning, Phys. Rev. Lett. **118**, 010601 (2017).

[37] S. Rooke, D. Krotov, V. Balasubramanian, and D. H. Wolpert, Stochastic thermodynamics of associative memory, arXiv:2601.01253.

[38] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, Attention is all you need, in *Advances in Neural Information Processing Systems 30* (Curran Associates, Inc., Red Hook, NY, 2017), pp. 5998–6008.

[39] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, Language models are unsupervised multitask learners, OpenAI Technical Report, 2019.

[40] T. Brown *et al.*, Language models are few-shot learners, in *Advances in Neural Information Processing Systems* (Curran Associates, Inc., Red Hook, NY, 2020), Vol. 33, pp. 1877–1901.

[41] H. Ramsauer *et al.*, Hopfield networks is all you need, in *The 9th International Conference on Learning Representations* (ICLR, 2021).

[42] J. L. Elman, Finding structure in time, Cogn. Sci. **14**, 179 (1990).

[43] S. Hochreiter and J. Schmidhuber, Long short-term memory, Neural Comput. **9**, 1735 (1997).

[44] R. E. Kalman, A new approach to linear filtering and prediction problems, J. Basic Eng. **82**, 35 (1960).

[45] B. D. O. Anderson and J. B. Moore, *Optimal Filtering* (Prentice-Hall, Englewood Cliffs, NJ, 1979).

[46] A. Lindquist and G. Picci, *Linear Stochastic Systems: A Geometric Approach to Modeling, Estimation and Identification* (Springer, Berlin, 2015).

[47] A. Gu, K. Goel, and C. Ré, Efficiently modeling long sequences with structured state spaces, in *International Conference on Learning Representations* (ICLR, 2022).

[48] J. T. H. Smith, A. Warrington, and S. W. Linderman, Simplified state space layers for sequence modeling, in *International Conference on Learning Representations* (ICLR, 2023).

[49] A. Gu and T. Dao, Mamba: Linear-time sequence modeling with selective state spaces, in *Conference on Language Modeling (COLM)* (2024).

[50] K. Li, A. K. Hopkins, D. Bau, F. Viégas, H. Pfister, and M. Wattenberg, Emergent world representations: Exploring a sequence model trained on a synthetic task, in *The 11th International Conference on Learning Representations* (ICLR, 2023).

[51] W. Gurnee and M. Tegmark, Language models represent space and time, in *The 12th International Conference on Learning Representations* (ICLR, 2024).

[52] D. P. Kingma and M. Welling, Auto-encoding variational Bayes, in *International Conference on Learning Representations* (ICLR, 2014).

[53] M. D. Hoffman and M. J. Johnson, ELBO surgery: Yet another way to carve up the variational evidence lower bound, in *Proceedings of the Workshop in Advances in Approximate Bayesian Inference* (NIPS, Barcelona, Spain, 2016).

[54] A. A. Alemi *et al.*, Fixing a broken ELBO, in *Proceedings of the 35th International Conference on Machine Learning (ICML)* (PMLR, 2018), pp. 159–168.

[55] T. Matsumoto and T. Sagawa, Role of sufficient statistics in stochastic thermodynamics and its implication to sensory adaptation, Phys. Rev. E **97**, 042103 (2018).

[56] R. Kawai, J. M. R. Parrondo, and C. Van den Broeck, Dissipation: The phase-space perspective, Phys. Rev. Lett. **98**, 080602 (2007).

[57] V. Papadopoulos, J. Wenger, and C. Hongler, Arrows of time for large language models, in *Proceedings of the 41st International Conference on Machine Learning (ICML)* (PMLR, 2024), pp. 39509–39528.

[58] S. Yu *et al.*, Reverse modeling in large language models, in *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies* (Association for Computational Linguistics, Kerrville, TX, 2025), Vol. 2, pp. 306–320.

[59] C. Moslonka, H. Randrianarivo, A. Garnier, and E. Malherbe, Learned hallucination detection in black-box LLMs using token-level entropy production rate, in *Advances in Information Retrieval* (ECIR 2026), Lecture Notes in Computer Science (Springer, Cham, 2026), Vol. 16483, pp. 115–130.

[60] R. G. James, N. Barnett, and J. P. Crutchfield, Information flows? A critique of transfer entropies, Phys. Rev. Lett. **116**, 238701 (2016).

[61] M. Miliani *et al.*, ExpliCa: Evaluating explicit causal reasoning in large language models, in *Findings of the Association for Computational Linguistics: ACL, 2025* (Association for Computational Linguistics, Kerrville, TX, 2025), pp. 17335–17355.

[62] S. K. Mitter and N. J. Newton, Information and entropy flow in the Kalman–Bucy filter, J. Stat. Phys. **118**, 145 (2005).

[63] N. J. Newton, Dual Kalman–Bucy filters and interactive entropy production, SIAM J. Control Optim. **45**, 998 (2006).

[64] N. J. Newton, Dual nonlinear filters and entropy production, SIAM J. Control Optim. **46**, 1637 (2007).

[65] N. J. Newton, Interactive statistical mechanics and nonlinear filtering, J. Stat. Phys. **133**, 711 (2008).

[66] J. M. Horowitz and H. Sandberg, Second-law-like inequalities with information and their interpretations, New J. Phys. **16**, 125007 (2014).

[67] H. Sandberg, J.-C. Delvenne, N. J. Newton, and S. K. Mitter, Maximum work extraction and implementation costs for nonequilibrium Maxwell's demons, Phys. Rev. E **90**, 042119 (2014).

[68] K. Kumasaki, K. Tojo, T. Sagawa, and K. Funo, Thermodynamic uncertainty relation for feedback cooling, Phys. Rev. E **113**, 024134 (2026).

[69] C. Godrèche and J. M. Luck, Characterising the nonequilibrium stationary states of Ornstein–Uhlenbeck processes, J. Phys. A: Math. Theor. **52**, 035002 (2019).

[70] G. T. Landi, T. Tomé, and M. J. de Oliveira, Entropy production in linear Langevin systems, J. Phys. A: Math. Theor. **46**, 395001 (2013).

[71] M. Gilson, E. Tagliazucchi, and R. Cofré, Entropy production of multivariate Ornstein–Uhlenbeck processes correlates with consciousness levels in the human brain, Phys. Rev. E **107**, 024121 (2023).

[72] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. (Wiley-Interscience, Hoboken, NJ, 2006).

[73] G. Weiss, Time-reversibility of linear stochastic processes, J. Appl. Probab. **12**, 831 (1975).

[74] H. Tong and Z. Zhang, On time-reversibility of multivariate linear processes, Stat. Sin. **15**, 495 (2005).

[75] T. T. Georgiou and A. Lindquist, On time-reversibility of linear stochastic models, in *Proceedings of the 19th IFAC World Congress* (IFAC, Cape Town, 2014), pp. 10403–10408.

[76] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, Deep unsupervised learning using nonequilibrium thermodynamics, in *Proceedings of the 32nd International Conference on Machine Learning (ICML)* (PMLR, 2015), pp. 2256–2265.

[77] J. Ho, A. Jain, and P. Abbeel, Denoising diffusion probabilistic models, in *Advances in Neural Information Processing Systems (NeurIPS 2020)* (Curran Associates, Inc., Red Hook, NY, 2020), Vol. 33, pp. 6840–6851.

[78] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, Score-based generative modeling through stochastic differential equations, in *Proceedings of the 9th International Conference on Learning Representations* (ICLR, 2021).

[79] B. D. O. Anderson, Reverse-time diffusion equation models, Stoch. Process. Appl. **12**, 313 (1982).

[80] A. Premkumar, Neural entropy, in *NeurIPS 2025* (spotlight) (Curran Associates, Inc., Red Hook, NY, 2025).

[81] A. C. Barato and U. Seifert, Thermodynamic uncertainty relation for biomolecular processes, Phys. Rev. Lett. **114**, 158101 (2015).

[82] J. M. Horowitz and T. R. Gingrich, Thermodynamic uncertainty relations constrain non-equilibrium fluctuations, Nat. Phys. **16**, 15 (2020).

[83] N. Shiraishi, K. Funo, and K. Saito, Speed limit for classical stochastic processes, Phys. Rev. Lett. **121**, 070601 (2018).

[84] K. Ikeda, T. Uda, D. Okanohara, and S. Ito, Speed-accuracy relations for diffusion models: Wisdom from nonequilibrium thermodynamics and optimal transport, Phys. Rev. X **15**, 031031 (2025).

[85] C. R. Shalizi and J. P. Crutchfield, Computational mechanics: Pattern and prediction, structure and simplicity, J. Stat. Phys. **104**, 817 (2001).

[86] A. B. Boyd, D. Mandal, and J. P. Crutchfield, Thermodynamics of modularity: Structural costs beyond the Landauer bound, Phys. Rev. X **8**, 031036 (2018).

[87] A. B. Boyd, D. Mandal, and J. P. Crutchfield, Leveraging environmental correlations: The thermodynamics of requisite variety, J. Stat. Phys. **167**, 1555 (2017).

[88] T. Sagawa, Data for Stochastic thermodynamics for autoregressive generative models: A non-Markovian perspective, GitHub, 2026, https://github.com/taksagawa/Paper2026.

[89] J. G. Kemeny and J. L. Snell, *Finite Markov Chains* (Van Nostrand, Princeton, NJ, 1960).

[90] T. Wolf *et al.*, Transformers: State-of-the-art natural language processing, in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (Association for Computational Linguistics, Stroudsburg, PA, 2020), pp. 38–45.

[91] OpenAI, GitHub repository openai/gpt-2, README, https://github.com/openai/gpt-2, 2026.

[92] Hugging Face, https://github.com/huggingface/transformers/blob/main/src/transformers/models/gpt2/tokenization_gpt2.py, 2026.

[93] Hugging Face, https://github.com/huggingface/transformers/blob/main/src/transformers/tokenization_utils_tokenizers.py, 2026.

[94] A. Yang *et al.*, Qwen3 Technical Report, arXiv:2505.09388.

[95] Hugging Face, https://huggingface.co/Qwen/Qwen3-4B-Base, 2026.