

Thermodynamic Machine Learning through Maximum Work Production

Alexander B. Boyd,^{1,2,*} James P. Crutchfield,^{3,†} and Mile Gu^{1,2,4,‡}

¹*Complexity Institute, Nanyang Technological University, Singapore*

²*School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore*

³*Complexity Sciences Center and Physics Department,*

University of California at Davis, One Shields Avenue, Davis, CA 95616

⁴*Centre for Quantum Technologies, National University of Singapore, Singapore*

(Dated: April 4, 2021)

Adaptive systems—such as a biological organism gaining survival advantage, an autonomous robot executing a functional task, or a motor protein transporting intracellular nutrients—must model the regularities and stochasticity in their environments to take full advantage of thermodynamic resources. Analogously, but in a purely computational realm, machine learning algorithms estimate models to capture predictable structure and identify irrelevant noise in training data. This happens through optimization of performance metrics, such as model likelihood. If physically implemented, is there a sense in which computational models estimated through machine learning are physically preferred? We introduce the thermodynamic principle that work production is the most relevant performance metric for an adaptive physical agent and compare the results to the maximum-likelihood principle that guides machine learning. Within the class of physical agents that most efficiently harvest energy from their environment, we demonstrate that an efficient agent’s model explicitly determines its architecture and how much useful work it harvests from the environment. We then show that selecting the maximum-work agent for given environmental data corresponds to finding the maximum-likelihood model. This establishes an equivalence between nonequilibrium thermodynamics and dynamic learning. In this way, work maximization emerges as an organizing principle that underlies learning in adaptive thermodynamic systems.

Keywords: nonequilibrium thermodynamics, Maxwell’s demon, Landauer’s Principle, extremal principles, machine learning, regularized inference, density estimation

I. INTRODUCTION

What is the relationship, if any, between abiotic physical processes and intelligence? Addressed to either living or artificial systems, this challenge has been taken up by scientists and philosophers repeatedly over the last centuries, from the 19th century teleologists [1] and biological structuralists [2, 3] to cybernetics of the mid-20th century [4, 5] and contemporary neuroscience-inspired debates of the emergence of artificial intelligence in digital simulations [6]. The challenge remains vital today [7–10]. A key thread in this colorful and turbulent history explores issues that lie decidedly at the crossroads of thermodynamics and communication theory—of physics and engineering. In particular, what bridges the dynamics of the physical world and its immutable laws and principles to the purposeful behavior intelligent agents? The following argues that an essential connector lies in a new thermodynamic principle: work maximization drives learning.

Perhaps unintentionally, James Clerk Maxwell laid foundations for a physics of intelligence with what Lord Kelvin (William Thomson) referred to as “intelligent demons”

[11]. Maxwell in his 1857 book *Theory of Heat* argued that a “very observant” and “neat fingered being” could subvert the Second Law of Thermodynamics [12]. In effect, his “finite being” uses its intelligence (Maxwell’s word) to sort fast from slow molecules, creating a temperature difference that drives a heat engine to do useful work. The demon presented an apparent paradox because directly converting disorganized thermal energy to organized work energy is forbidden by the Second Law. The cleverness in Maxwell’s paradox turned on equating the thermodynamic behavior of mechanical systems with the intelligence in an agent that can accurately measure and control its environment. This established an operational equivalence between energetic thermodynamic processes, on the one hand, and intelligence, on the other.

We will explore the intelligence of physical processes, substantially updating the setting from the time of Kelvin and Maxwell, by calling on a wealth of recent results on the nonequilibrium thermodynamics of information [13, 14]. In this, we directly equate the operation of physical agents descended from Maxwell’s demon with notions of intelligence found in modern machine learning. While learning is not necessarily the only capability of a presumed intelligent being, it is certainly a most useful and interesting feature.

The root of many tasks in machine learning lies in discovering structure from data. The analogous process of creating models of the world from incomplete informa-

* alecboy@gmail.com

† chaos@ucdavis.edu

‡ mgu@quantumcomplexity.org

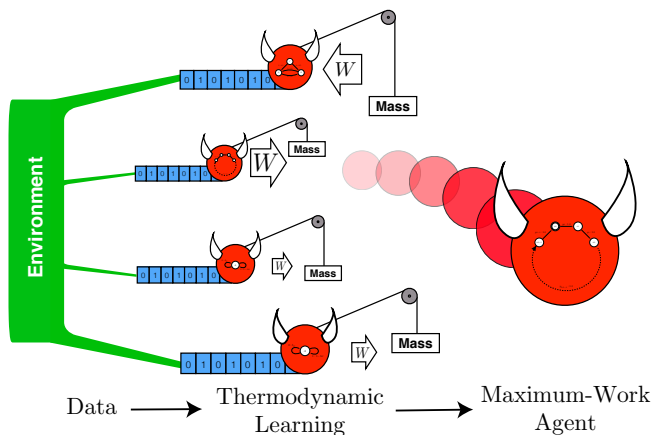


FIG. 1. Thermodynamic learning generates the maximum-work producing agent: (Left) Environment (green) behavior becomes data for agents (red). (Middle) Candidate agents each have an internal model (inscribed stochastic state-machine) that captures the environment’s randomness and regularity to store work energy (e.g., lift a mass against gravity) or to borrow work energy (e.g., lower the mass). (Right) Thermodynamic learning searches the candidate population for the best agent—that producing the maximum work.

tion is essential to adaptive organisms, too, as they must model their environment to categorize stimuli, predict threats, leverage opportunities, and generally prosper in a complex world. Most prosaically, translating *training data* into a generative *model* corresponds to *density estimation* [15–17], where the algorithm uses the data to construct a probability distribution.

This type of model-building at first appears far afield from more familiar machine learning tasks such as categorizing pet pictures into cats and dogs or generating a novel image of a giraffe from a photo travelogue. Nonetheless, it encompasses them both [18]. Thus, by addressing thermodynamic roots of model estimation, we seek a physical foundation for a wide breadth of machine learning.

To carry out density estimation, machine learning invokes the principle of maximum-likelihood to guide intelligent learning. This says, of the possible models consistent with the training data, an algorithm should select that with maximum probability of having generated the data. Our exploration of the physics of learning asks whether a similar thermodynamic principle guides physical systems to adapt to their environments.

The modern understanding of Maxwell’s demon no longer entertains violating the Second Law of Thermodynamics [19]. In point of fact, the Second Law’s primacy has been repeatedly affirmed in modern nonequilibrium theory and experiment. That said, what has emerged is that we now understand how intelligent (demon-like) physical processes can harvest thermal energy as useful work. They do this by exploiting an information reservoir [19–21]—a

storehouse of information as randomness and correlation. That reservoir is the demon’s informational environment, and the mechanism by which the demon measures and controls its environment embodies the demon’s intelligence, according to modern physics. We will show that this mechanism is directly linked to the demon’s model of its environment, which allows us to formalize the connection to machine learning.

Machine learning estimates different likelihoods of different models given the same data. Analogously, in the physical setting of information thermodynamics, different demons harness different amounts of work from the same information reservoir. Leveraging this commonality, Sec. II introduces *thermodynamic learning* as a physical process that infers optimal demons from environmental information. As shown in Fig. 1, thermodynamic learning selects demons that produce maximum work, paralleling parametric density estimation’s selection of models with maximum likelihood. Section III establishes background in density estimation, computational mechanics, and thermodynamic computing necessary to formalize the comparison of maximum-work and maximum-likelihood learning. Our surprising result is that these two principles of maximization are the same, when compared in a common setting. This adds credence to the long-standing perspective that thermodynamics and statistical mechanics underlie many of the tools of machine learning [17, 22–28].

Section IV formally establishes that to construct an intelligent work-harvesting demon, a probabilistic model of its environment is essential. That is, the demon’s Hamiltonian evolution is directly determined by its environmental model. This comes as a result of discarding demons that are ineffective at harnessing energy from *any* input, focusing only on a refined class of *efficient* demons that make the best use of the given data. This leads to the central result, found in Sec. V, that the demon’s work production from environmental “training data” increases linearly with the log-likelihood of the demon’s model of its environment. Thus, if the thermodynamic training process selects the maximum-work demon for given data, it has also selected the maximum-likelihood model for that same data.

Ultimately, our work demonstrates an *equivalence between the conditions of maximum work and maximum likelihood*. In this way, *thermodynamic learning is machine learning for thermodynamic machines*—it is a physical process that infers models in the same way a machine learning algorithm does. Thus, work itself can be interpreted as a *thermodynamic performance measure for learning*. In this framing, learning is physical, building on the long-lived narrative of the thermodynamics of organization, which we recount in Sec. VI. While it is natural to argue that

learning confers benefits, our result establishes that the benefit is fundamentally rooted in the physical tradeoff between energy and information.

II. FRAMEWORK

While demons continue to haunt discussions of physical intelligence, the notion of a physical process trafficking in information and energy exchanges need not be limited to mysterious intelligent beings. Most prosaically, we are concerned with any physical system that, while interacting with an environment, simultaneously processes information at some energetic cost or benefit. Avoiding theological distractions, we refer to these processes as *thermodynamic agents*. In truth, any physical system can be thought of as an agent, but only a limited number of them are especially useful for or adept at commandeering information to convert between various kinds of thermodynamic resources, such as between heat and work. Here, we introduce a construction that shows how to find physical systems that are the most capable of processing information to affect thermodynamic transformations.

Consider an environment that produces information in the form of a time series of physical values at regular time intervals of length τ . We denote the particular state realized by the environment’s output at time $j\tau$ by the symbol $y_j \in \mathcal{Y}_j$. Just as the agent must be instantiated by a physical system, so too must the environment and its outputs to the agent. Specifically, \mathcal{Y}_j represents the state space of the j th output, which is a subsystem of the environment.

An agent has no access to the internals of its environment and so treats it as a black box. Thus, the agent can only access and interact with the environment’s output system \mathcal{Y}_j over each time interval $t \in (j\tau, (j+1)\tau)$. In other words, the state y_j realized by the environment’s output is also the agent’s *input* at time $j\tau$. For instance, the environment may produce realizations of a two level spin system $\mathcal{Y}_j = \{\uparrow, \downarrow\}$, which the agent is then tasked to manipulate through Hamiltonian control.

The aim, then, is to find an agent that produces as much work as possible using these black-box outputs. To do so, the agent must come to know something about the black box’s structure. This is the *principle of requisite complexity* [29]—thermodynamic advantage requires that the agent’s organization match that of its environment. We implement this by introducing a method for thermodynamic learning as shown in Fig. 1, that selects a specific agent from a collection of candidates.

Peeking into the internal mechanism of the black box, we wait for a time $L\tau$, receiving the L symbols $y_{0:L} = y_0 y_1 \cdots y_{L-1}$. This is the agent’s training data, which is

copied as needed to allow a population of candidate agents to interact with it. As each agent interacts with a copy, it produces an amount of work, which it stores in the work reservoir for later use. In Fig. 1, the work reservoir is illustrated by a hanging mass which raises when positive work is produced, storing more energy in gravitational potential energy, and lowers when work production is negative, expending that same potential energy. However the work energy is stored, after the agents harvest work from the training data, the agent that produced the most work is selected.

Finding the maximum-work agent is “thermodynamic learning” in the sense that it selects a device based on measuring its thermodynamic performance—the amount of work the device extracts. Ultimately, the goal is that the agent selected by thermodynamic learning continues to extract work as the environment produces new symbols. However, we leave analyzing the long-term effectiveness of thermodynamic learning to the future. Here, we concentrate on the condition of maximum-work itself, deriving and interpreting it.

Section IV begins by describing the general class of physical agents that can harness work from symbol sequence, known as *information ratchets* [30, 31]. While these agents are sufficiently general to implement virtually any (Turing) computation, maximizing work production precludes a wide array of agents. Section IV B then refines our consideration to agents that waste as little work as possible and, in so doing, vastly narrow the search by thermodynamic learning. For this refined class of agents, we find that each agent’s operation is exactly determined by its environment model. This leads to our final result, that the agent’s work increases linearly with the model’s log-likelihood.

For clarity, note that thermodynamic learning differs from physical systems that, evolving in time, dynamically adapt to their environment [26, 32, 33]. Work maximization as described here is thermodynamic in its objective, while these previous approaches to learning are thermodynamic in their mechanism.

That said, the perspectives are linked. In particular, it was suggested that physical systems spontaneously decrease work absorbed from driving [32]. Note that work absorbed by the system is opposite the work produced. And so, as they evolve over time, these physical systems appear to seek higher work production, paralleling how thermodynamic learning selects for the highest work production. And, the synchronization by which a physical system decreases work absorption is compared to learning [32]. Reference [33] goes further, comparing the effectiveness of physical evolution to maximum-likelihood estimation employing an autoencoder. Notably, it reports that that form of machine learning performs markedly

better than physical evolution, for the particular system considered there. By contrast, we show that the advantage of machine learning over thermodynamic learning does not hold in our framework. Simply speaking, they are synonymous.

We compare thermodynamic learning to machine learning algorithms that use maximum-likelihood to select models consistent with given data. As Fig. 1 indicates, each agent has an internal model of its environment; a connection Sec. IV F formalizes. Each agent's work production is then evaluated for the training data. Thus, arriving at a maximum-work agent also selects that agent's internal model as a description of the environment. Moreover and in contrast with Ref. [33], which compares thermodynamic and machine learning methods numerically, the framework here leads to an analytic derivation of the equivalence between thermodynamic learning and maximum-likelihood density estimation.

III. PRELIMINARIES

Directly comparing thermodynamic learning and density estimation requires explicitly demonstrating that thermodynamically-embedded computing and machine learning share the framework just laid out. The following introduces what we need for this: concepts from machine learning, computational mechanics, and thermodynamic computing. (Readers preferring fuller detail should refer to App. A.)

A. Parametric Density Estimation

Parametric estimation determines, from training data, the parameters θ of a probability distribution. In the present setting, θ parametrizes a family of probabilities $\Pr(Y_{0:\infty} = y_{0:\infty} | \Theta = \theta)$ over sequences (or *words*) of any length. Here, $Y_{0:\infty} = Y_0 Y_1 \dots$ is the infinite-sequence random variable, composed of the random variables Y_j that each realize the environment's output y_j at time $j\tau$, and Θ is the random variable for the model. In other words, a given model θ predicts the probability of any sequence $y_{0:L}$ of any length L that one might see. For convenience, we introduce random variables Y_j^θ that define a model:

$$\Pr(Y_{0:\infty}^\theta) \equiv \Pr(Y_{0:\infty} | \Theta = \theta) .$$

With training data $y_{0:L}$, the *likelihood* of model θ is given

by the probability of the data given the model:

$$\begin{aligned} \mathcal{L}(\theta | y_{0:L}) &= \Pr(Y_{0:L} = y_{0:L} | \Theta = \theta) \\ &= \Pr(Y_{0:L}^\theta = y_{0:L}) . \end{aligned}$$

Parametric density estimation seeks to optimize the likelihood $\mathcal{L}(\theta | y_{0:L})$ [15, 17, 34]. However, the procedure that finds maximum-likelihood estimates usually employs the *log-likelihood* instead:

$$\ell(\theta | y_{0:L}) = \ln \Pr(Y_{0:L}^\theta = y_{0:L}) , \quad (1)$$

since it is maximized for the same models, but converges more effectively [35].

B. Computational Mechanics

Given that our data is a time series of arbitrary length starting with y_0 , we must choose a model class whose possible parameters $\Theta = \{\theta\}$ specify a wide range of possible distributions $\Pr(Y_{0:\infty}^\theta)$ —the *semi-infinite processes*. ϵ -Machines, a class of finite-state machines introduced to describe bi-infinite processes $\Pr(Y_{-\infty:\infty}^\theta)$, provide a systematic means to do this [36]. As described in App. A these finite-state machines comprise just such a flexible class of representations; they can describe any semi-infinite process. This follows from the fact that they are the minimal sufficient statistic for prediction explicitly constructed from the process.

A process's ϵ -machine consists of a set of hidden states \mathcal{S} , a set of output states \mathcal{Y} , a start state $s^* \in \mathcal{S}$, and conditional output-labeled transition matrix $\theta_{s \rightarrow s'}^{(y)}$ over the hidden states:

$$\theta_{s \rightarrow s'}^{(y)} = \Pr(S_{j+1}^\theta = s', Y_j^\theta = y | S_j^\theta = s) .$$

$\theta_{s \rightarrow s'}^{(y)}$ specifies the probability of transitioning to hidden state s' and emitting symbol y given that the machine is in state s . In other words, the model is fully specified by the tuple:

$$\theta = \{\mathcal{S}, \mathcal{Y}, s^*, \{\theta_{s \rightarrow s'}^{(y)}\}_{s, s' \in \mathcal{S}, y \in \mathcal{Y}}\} .$$

As an example, Fig. 2 shows an ϵ -machine that generates a periodic process with initially uncertain phase. ϵ -Machines are *unifilar*, meaning that the current causal state s_j along with the next k symbols uniquely determines the following causal state through the *propagator function*:

$$s_{j+k} = \epsilon(s_j, y_{j:j+k}) .$$

This yields a simple expression for the probability of any

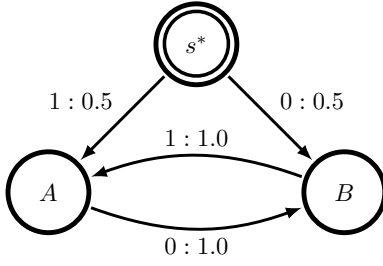


FIG. 2. ϵ -Machine generating the phase-uncertain period-2 process: With probability 0.5, an initial transition is made from the start state s^* to state A . From there, it emits the sequence 1010... However, with probability 0.5, the start state transitions to state B and outputs the sequence 0101...

word in terms of the model parameters:

$$\Pr(Y_{0:L}^\theta = y_{0:L}) = \prod_{j=0}^{L-1} \theta_{\epsilon(s^*, y_{0:j}) \rightarrow \epsilon(s^*, y_{0:j+1})}^{(y_j)}.$$

In addition to being uniquely determined by the semi-infinite process, the ϵ -machine uniquely generates that same process. This means that our model class Θ is equivalent to the class of possible distributions over time series data. Moreover, knowledge of the causal state of an ϵ -machine at any time step j contains all information about the future that could be predicted from the past. In this sense, the causal state is *predictive* of the process. These and other properties have motivated a long investigation of ϵ -machines, in which the memory cost of storing the causal states is frequently used as a measure of process structure. Appendix A gives an extended review.

C. Thermodynamic Computing

Computation is physical—any computation takes place embedded in a physical system. Here, we refer to substrate of the physically-embedded computation as the *system of interest* (SOI). Its states, denoted $\mathcal{Z} = \{z\}$, are taken as the underlying physical system's *information bearing degrees of freedom* [19]. The SOI's dynamic evolves the state distribution $\Pr(Z_t = z_t)$, where Z_t is the random variable describing state at time t . Computation over time interval $t \in [\tau, \tau']$ specifies how the dynamic maps the SOI from the initial time $t = \tau$ to the final time $t = \tau'$. It consists of two components:

1. An initial distribution over states $\Pr(Z_\tau = z_\tau)$ at time $t = \tau$.
2. Application of a Markov channel M , characterized by the conditional probability of transitioning to

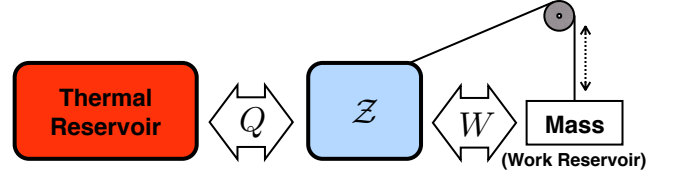


FIG. 3. Thermodynamic computing: The system of interest \mathcal{Z} 's states store information, processing it as they evolve. The work reservoir, represented as the suspended mass, supplies work energy W to drive the SOI Hamiltonian along a deterministic trajectory $\mathcal{H}_Z(t)$. Meanwhile, heat energy Q is exchanged with the thermal reservoir, driving the system toward thermal equilibrium.

the final state $z_{\tau'}$ given the initial state z_τ :

$$M_{z_\tau \rightarrow z_{\tau'}} = \Pr(Z_{\tau'} = z_{\tau'} | Z_\tau = z_\tau).$$

Together, these specify the SOI's computational elements. In this, z_τ is the *input* to the physical computation, $z_{\tau'}$ is the *output*, and $M_{z_\tau \rightarrow z_{\tau'}}$ is the *logical architecture*.

Figure 3 illustrates a computation's physical implementation. SOI \mathcal{Z} is coupled to a work reservoir, depicted as a mass hanging from a string, that controls the system's Hamiltonian along a trajectory $\mathcal{H}_Z(t)$ over the computation interval $t \in [\tau, \tau']$ [37]. This is the basic definition of a *thermodynamic agent*: an evolving Hamiltonian driving a physical system to compute at the cost of work.

In a classical system, this control determines each state's energy $E(z, t)$. As a result of the control, changes in energy due to changes in the Hamiltonian correspond to work exchanges between the SOI and work reservoir. The system \mathcal{Z} follows a *state trajectory* $z_{\tau:\tau'}$ over the time interval $t \in [\tau, \tau']$, which we can write:

$$z_{\tau:\tau'} = z_\tau z_{\tau+dt} \cdots z_{\tau'-dt} z_{\tau'},$$

where z_t is the system state at time t . Here, we decomposed the trajectory into intervals of duration dt , taken short enough to yield infinitesimal changes in state probabilities and the Hamiltonian. The resulting *work production* for this trajectory is then the integrated change in energy due to the Hamiltonian's time dependence [37]:

$$W|_{z_{\tau:\tau'}} = - \int_{\tau}^{\tau'} dt \partial_t E(z, t) \big|_{z=z_t}.$$

Note that while the state trajectory $z_{\tau:\tau'}$ mirrors the time series notation used for the training data $y_{0:L} = y_0 y_1 \cdots y_{L-1}$, they are different objects and should not be conflated. On the one hand, the training data series $y_{0:L}$ is composed of realizations of L *separate* subsystems, each produced at different times $j\tau$, $j \in \{0, 1, 2, \dots, L-1\}$. y_j is

realized in the subsystem \mathcal{Y}_j , and so it can be manipulated completely separately from any other element of $y_{0:L}$ lying outside of \mathcal{Y}_j . By contrast, z_t depends dynamically on many other elements in $z_{\tau:\tau'}$, all of which lie in the *same* system, since the time series $z_{\tau:\tau'}$ represents state evolution of the single system \mathcal{Z} over time.

While the SOI exchanges work energy with the work reservoir, as Fig. 3 shows, it exchanges heat Q with the thermal reservoir. Coupling to a heat reservoir adds stochasticity to the state trajectory $z_{\tau:\tau'}$. Since the SOI computes while coupled to a thermal reservoir at temperature T , *Landauer's Principle* [19] relates a computation's logical processing to its energetics. In its contemporary form, it bounds the *average* work production $\langle W \rangle$ by a term proportional to SOI's entropy change. Taking the Shannon entropy $H[Z_t] = -\sum_z \Pr(Z_t = z) \ln \Pr(Z_t = z)$ in natural units, the Second Law of Thermodynamics implies [14]:

$$\langle W \rangle \leq k_B T (H[Z_{\tau'}] - H[Z_{\tau}]) .$$

Here, the average $\langle W \rangle$ is taken over all possible microscopic trajectories. And, the energy landscape is assumed to be flat at the computation's start and end, giving no energetic preference to a particular informational state.

IV. AGENT ENERGETICS

We now construct the theoretical framework for how agents extract work from time-series data. This involves breaking down the agent's actions into manageable elementary components—where we demonstrate their actions can be described as repeated application of sequence of computations. We then introduce tools to analyze work production within such general computations on finite data. We highlight the importance of the agent's model of the data in determining work production. This model-dependence emerges by refining the class of agents to those that execute their computation most efficiently. The results are finally combined, resulting in a closed-form expression for agent work production from time-series data.

A. Agent Architecture

Recall from Sec. II that the basic framework describes a thermodynamic agent interacting with an environment at regular time-intervals τj in state y_j . Each y_j is drawn according to a random variable Y_j , such that the sequence $Y_{0:\infty} = Y_0 Y_1 \dots$ is a semi-infinite stochastic process. The agent's task is to interact with this input string to generate useful work.

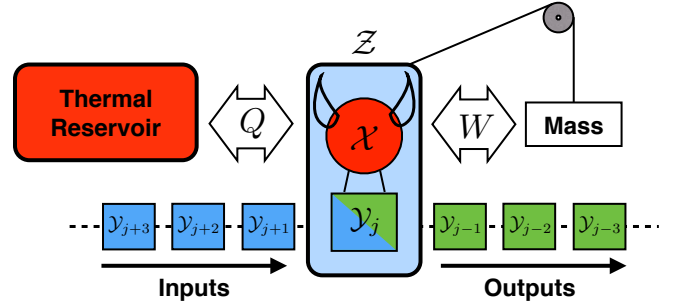


FIG. 4. Thermodynamic computing by an agent driven by an input sequence: Information bearing degrees of freedom of SOI \mathcal{Z} in the j th interaction interval split into the direct product of agent states \mathcal{X} and the j th input states \mathcal{Y}_j . Work W and heat Q are defined in the same way as in Fig. 3, with the SOI's Hamiltonian control $\mathcal{H}_{\mathcal{X} \times \mathcal{Y}_j}(t)$ explicitly decoupled from the environment's remaining subsystems $\dots \mathcal{Y}_{j-2} \mathcal{Y}_{j-1} \mathcal{Y}_{j+1} \mathcal{Y}_{j+2} \dots$, corresponding to future inputs $\mathcal{Y}_{j+1} \mathcal{Y}_{j+2} \dots$ and past outputs $\dots \mathcal{Y}_{j-2} \mathcal{Y}_{j-1}$.

For example, consider an agent charged with extracting work from an alternating process—a sequence emitted by a degenerate two-level system that alternates periodically between symbols 0 and 1. In isolation each symbol looks random and has no free energy. Thus, an agent that interacts with each symbol the same way gains no work. However, a memoryful agent can adaptively adjust its behavior, after reading the first symbol, to exactly predict succeeding symbols and, therefore, extract meaningful work. This method of harnessing *temporal correlations* is implemented by *information ratchets* [30, 31]. They combine physical inputs with additional agent memory states that store the input's temporal correlations.

As shown in Fig. 4, we describe an agent's memory via an ancillary physical system \mathcal{X} . The agent then operates cyclically with duration τ , such that the j^{th} cycle runs over the time-interval $[j\tau, (j+1)\tau)$. Each cycle involves two phases:

1. *Interaction*: Agent memory \mathcal{X} couples to and interacts with the j^{th} input system \mathcal{Y}_j that contains the j^{th} input symbol y_j . This phase has duration $\tau' < \tau$, meaning the j th interaction phase occurs over the time-interval $[j\tau, j\tau + \tau')$. At the end, the agent decouples from the system \mathcal{Y}_j , passing its new state y'_j to the environment as output or exhaust.
2. *Rest*: During time interval $[j\tau + \tau', (j+1)\tau)$, the agent's memory \mathcal{X} sits idle, waiting for the next input \mathcal{Y}_{j+1} .

In this way, the agent transforms a series of inputs $y_{0:L}$ into a series of outputs $y'_{0:L}$.

In each cycle, all nontrivial thermodynamics occur in the interaction phase, during which the SOI consists of the joint agent-input system: i.e., $\mathcal{Z} = \mathcal{X} \otimes \mathcal{Y}_j$, as shown in

Fig. 4. While the other subsystems $\cdots \mathcal{Y}_{j-2}\mathcal{Y}_{j-1}$ and $\mathcal{Y}_{j+1}\mathcal{Y}_{j+2}\cdots$ may be physically instantiated somewhere else in the environment, they do not participate in the interaction, since they are energetically decoupled from the agent in this phase. Paralleling the computation shown in Fig. 3, Hamiltonian control over the joint space $\mathcal{H}_{\mathcal{X}\times\mathcal{Y}_j}(t)$ results in a transformation of the agent's SOI that also requires the exchange of work and heat. This interaction phase updates SOI states according to a Markov transition matrix M shown in Fig. 5:

$$M_{xy \rightarrow x'y'} = \Pr(X_{j+1} = x', Y'_j = y' | X_j = x, Y_j = y), \quad (2)$$

where X_j and X_{j+1} are the random variables for the states of the agent's memory \mathcal{X} before and after the j th interaction interval, and Y_j and Y'_j are the random variables for the system \mathcal{Y}_j before and after the same interaction interval, realizing the input and output, respectively.

As Sec. III C described, M is the logical architecture of the physical computation that transforms the agent's memory and input simultaneously. It is the central element in the agent's procedure for transforming inputs $y_{0:L}$ into associated outputs $y'_{0:L}$. The key observation is that M captures all of the agent's internal logic. The logic does not change from cycle to cycle. However, the presence of persistent internal memory between cycles implies that the agent's behavior adapts to past inputs and outputs. This motivates us to define M as the *agent architecture* since it determines how an agent stores information temporally. As we will show, M is one of two essential elements in determining the work an agent produces from a time series.

Note that prior related efforts to address agent energetics focused on ensemble-average work production [29–31, 38–42]. In contrast, here we relate work production to parametric density estimation—which involves each agent being given a specific data string $y_{0:L}$ for training. To address this case, the following determines the work production for single-shot short input strings.

B. Energetics of Computational Maps

The agent architecture M specifies a physical computation as described in Sec. III C and therefore has a minimum energy cost determined by Landauer's bound. However, this is a bound on the *average* work production, which depends explicitly on the distribution of inputs. We need to determine, instead, the work produced from a *single* input y_j . To find this we return to the general case of SOI \mathcal{Z} undergoing a thermodynamic computation M . A physical operation takes the SOI from state z_τ at time τ to state $z_{\tau'}$ at time τ' . This specifies a *computational map*

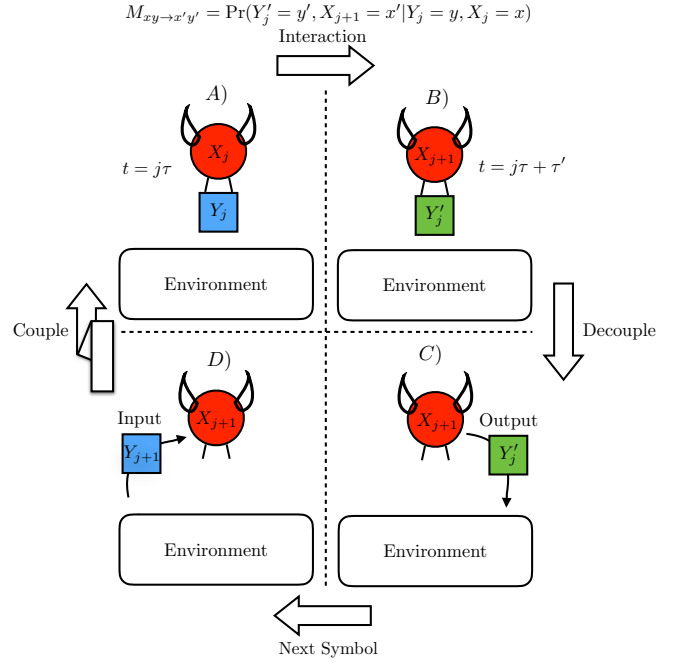


FIG. 5. Agent interacting with an environment via repeated symbol exchanges: A) At time $j\tau$ agent memory X_j begins interacting with input symbol Y_j . Transitioning from A) to B), agent memory and interaction symbol jointly evolve according to the Markov channel $M_{xy \rightarrow x'y'}$. This results in B)—the updated states of agent memory X_{j+1} and interaction symbol Y'_j at time $j\tau + \tau'$. Transitioning from B) to C), the agent memory decouples from the interaction symbol, emitting its new state to the environment. Then, transitioning from C) to D), the agent retains its memory state X_{j+1} and the environment emits the next interaction symbol Y_{j+1} . Finally, transitioning from D) to A), the agent restarts the cycle by coupling to the next input symbol.

$z_\tau \rightarrow z_{\tau'}$ that ignores intermediate states in the SOI state trajectory, as all information relevant to the computation's logical operation lies in the input and output. Thus, our attention turns to the question: What is the work production of a computational map $z_\tau \rightarrow z_{\tau'}$ performed by the computation M at temperature T ?

To determine this, we first prove a useful relation between the entropy and work production for a particular state trajectory $z_{\tau:\tau'}$. Specifically, let $W_{|z_{\tau:\tau'}}$ and $\Sigma_{|z_{\tau:\tau'}}$ denote the work and total entropy production along this trajectory, respectively. Meanwhile, let $E(z_t, t)$ denote the system energy when it is in state z_t at time t . Now, consider the *pointwise nonequilibrium free energy*:

$$\phi(z_t, t) = E(z_t, t) + k_B T \ln \Pr(Z_t = z_t). \quad (3)$$

More familiarly, note that its time-averaged quantity is the *nonequilibrium free energy* [43]:

$$F^{\text{neq}} = \langle \phi(z, t) \rangle_{\Pr(Z_t = z)}.$$

We can then show that the *entropy production* Σ can be expressed:

$$\Sigma_{|z_{\tau}, \tau'} = \frac{-W_{|z_{\tau}, \tau'} + \phi(z_{\tau}, \tau) - \phi(z_{\tau'}, \tau')}{T}. \quad (4)$$

This follows by noting that the total entropy produced from thermodynamic control is the sum of the entropy change in the system [44]:

$$\Delta S_{|z_{\tau}, \tau'}^{\mathcal{Z}} = k_B \ln \frac{\Pr(Z_{\tau} = z_{\tau})}{\Pr(Z_{\tau'} = z_{\tau'})}$$

and that of the thermal reservoir:

$$\Delta S_{|z_{\tau}, \tau'}^{\text{reservoir}} = \frac{Q_{|z_{\tau}, \tau'}}{T}.$$

Equation (4) follows by summing up these contributions to the total entropy production $\Sigma = \Delta S^{\text{reservoir}} + \Delta S^{\mathcal{Z}}$ and noting that the SOI's change in energy obeys the First Law of Thermodynamics $\Delta E^{\mathcal{Z}} = -W - Q$.

Since only the SOI's initial and final states matter to the logical operation of the computational map, we take a statistical average of all trajectories beginning in z_{τ} and ending in $z_{\tau'}$. This results in the work production:

$$\langle W_{|z_{\tau}, z_{\tau'}} \rangle = \sum_{z'_{\tau}, \tau'} W_{|z'_{\tau}, \tau'} \Pr(Z_{\tau}, \tau' = z'_{\tau}, \tau' | z_{\tau}, z_{\tau'}), \quad (5)$$

for the computational map $z_{\tau} \rightarrow z_{\tau'}$. This determines how much energy is stored in the work reservoir on average when a computation results in this particular input-output pair.

Similarly, taking the same average of the entropy production shown in Eq. (4), conditioned on inputs and outputs, gives:

$$\begin{aligned} T \langle \Sigma_{|z_{\tau}, z_{\tau'}} \rangle &= -\langle W_{|z_{\tau}, z_{\tau'}} \rangle + \phi(z_{\tau}, \tau) - \phi(z_{\tau'}, \tau'), \\ &= -\langle W_{|z_{\tau}, z_{\tau'}} \rangle - \Delta \phi_{|z_{\tau}, z_{\tau'}}. \end{aligned}$$

This suggestively relates computational-mapping work and the change in pointwise nonequilibrium free energy $\phi(z, t)$.

This relation between work and free energy simplifies for thermodynamically-efficient computations. In such scenarios, the average total entropy production over all trajectories vanishes. Appendix B shows that zero average entropy production, combined with the Crooks fluctuation theorem [45, 46], implies that entropy production along any individual trajectory z_{τ}, τ' produces zero entropy: $\Sigma_{|z_{\tau}, \tau'} = 0$. This is expected from linear response [47].

Thus, substituting zero entropy production into Eq. (4), we arrive at our result: *work production for thermodynamically-efficient computations is the change*

in pointwise nonequilibrium free energy:

$$W_{|z_{\tau}, \tau'}^{\text{eff}} = -\Delta \phi_{|z_{\tau}, z_{\tau'}}.$$

Substituting Eq. (3) then gives:

$$W_{|z_{\tau}, \tau'}^{\text{eff}} = -\Delta E_{\mathcal{Z}} + k_B T \ln \frac{\Pr(Z_{\tau} = z_{\tau})}{\Pr(Z_{\tau'} = z_{\tau'})},$$

where $\Delta E_{\mathcal{Z}} = E(z_{\tau'}, \tau') - E(z_{\tau}, \tau)$. This is what we would expect in quasistatic computations, where the system energies $E(z, t)$ are varied slowly enough that the system \mathcal{Z} remains in equilibrium for the duration. We should note, though, that it is possible to implement efficient computations rapidly and out of equilibrium [48].

This also holds if we average over intermediate states of the SOI's state trajectory, yielding the work production of a computational map:

$$\langle W_{|z_{\tau}, z_{\tau'}}^{\text{eff}} \rangle = -\Delta E_{\mathcal{Z}} + k_B T \ln \frac{\Pr(Z_{\tau} = z_{\tau})}{\Pr(Z_{\tau'} = z_{\tau'})}. \quad (6)$$

The energy required to perform efficient computing is independent of intermediate properties. It depends only on the probability and energy of initial and final states. This measures the energetic gains from a single data realization as it transforms during a computation, as opposed to the ensemble average.

C. Energetics of Estimates

Thermodynamic learning concerns agents that maximize work production from their input data. As such, we now restrict our attention to agents that harness all available nonequilibrium free energy in the form of work $\langle W \rangle = -\Delta F^{\text{neq}}$. These *maximum-work agents* zero out the average entropy production $\langle \Sigma \rangle = -\langle W \rangle - \Delta F^{\text{neq}}$ and the work production of a computational map satisfies Eq. (6). From here on out, when we refer to *efficient* agents we refer to those that maximize work production from the available change in nonequilibrium free energy. SOI state probabilities feature centrally in the expression for nonequilibrium free energy and, thus, for the work production of efficient agents. However, the *actual* input distribution $\Pr(Z_{\tau})$ may vary while the agent, defined by its Hamiltonian $\mathcal{H}_{\mathcal{Z}}(t)$ over the computation interval, remains fixed. Moreover, since the work production $\langle W_{|z_{\tau}, z_{\tau'}}^{\text{eff}} \rangle$ of a computational map explicitly conditions on the initial and final SOI state, this work cannot explicitly depend on the input distribution. At first blush, this is a contradiction: work that simultaneously does and does not depend on the input distribution.

This is resolved once one recognizes the role that estimates

play in thermodynamics. As indicated in Fig. 1, we claim that an agent has an estimated model of its environment that it uses to predict the SOI. This model is, in one form or another, encoded in the evolving Hamiltonian $\mathcal{H}_{\mathcal{Z}}(t)$ that determines both the agent's energetic interactions and its logical architecture. If an agent's estimated model of SOI \mathcal{Z} is encoded as parameters θ , then the agent estimates that the SOI state z at time t has probability:

$$\Pr(Z_t^\theta = z_t) = \Pr(Z_t = z_t | \Theta = \theta) .$$

The physical relevance of the estimated distribution comes from insisting that *the agent dissipates as little work as possible from a SOI whose distribution matches its own estimate*. In essence, the initial estimated distribution $\Pr(Z_\tau^\theta)$ must be one of the distributions that minimizes the average entropy production [49]:

$$\Pr(Z_\tau^\theta) \in \arg \min_{\Pr(Z_\tau)} \langle \Sigma [\Pr(Z_\tau)] \rangle .$$

Estimated probabilities $\Pr(Z_t^\theta)$ at later times $t > \tau$ are determined by updating the initial estimate via the stochastic dynamics that result from the Hamiltonian $\mathcal{H}_{\mathcal{Z}}(t)$ interacting with the thermal bath.

Thus, since an efficient agent produces zero entropy when the SOI follows the minimum dissipation distribution $\Pr(Z_t^\theta)$, the work it produces from a computational map is:

$$\langle W_{|z_\tau, z_{\tau'}}^\theta \rangle = -\Delta E_{\mathcal{Z}} + k_B T \ln \frac{\Pr(Z_\tau^\theta = z_\tau)}{\Pr(Z_{\tau'}^\theta = z_{\tau'})} . \quad (7)$$

In this, we replaced the superscript “eff” with “ θ ” to emphasize that the agent is designed to be thermodynamically efficient for that particular estimated model. Specifying the estimated model is essential, since misestimating the input distribution leads to dissipation and entropy production [49, 50]. Returning to thermodynamic learning, this is how the model θ factors into the ratchet's operation: estimated distributions explicitly determine the work production of computational maps.

Appendix C gives a concrete quasistatic mechanism for implementing *any* computation M and achieving the work given by Eq. (7). This directly demonstrates how the model θ is built into the evolving energy landscape $\mathcal{H}_{\mathcal{Z}}(t)$ that implements M . The model θ determines the initial and final change in state energies: $\Delta E(z, \tau) = -k_B \ln \Pr(Z_\tau^\theta = z)$ and $\Delta E(z, \tau') = k_B \ln \Pr(Z_{\tau'}^\theta = z)$. This quasistatic protocol operates and produces the same work for a particular input-output pair regardless of the *actual* input distribution.

Since we focus on the energetic benefits derived from information itself rather than those from changing energy

levels, the example implementations we use also start and end with the same flat energy landscape. Restricting to such information-driven agents, we consider cases where $\Delta E_{\mathcal{Z}} = 0$, whereby:

$$\langle W_{|z_\tau, z_{\tau'}}^\theta \rangle = k_B T \ln \frac{\Pr(Z_\tau^\theta = z_\tau)}{\Pr(Z_{\tau'}^\theta = z_{\tau'})} . \quad (8)$$

This provides a direct relationship between the work production $\langle W_{|z_\tau, z_{\tau'}}^\theta \rangle$ from particular data realizations and the model θ that the agent uses, via the estimates $\Pr(Z_t^\theta)$ provided by that model. This is an essential step in determining a model through thermodynamic learning.

D. Thermally Efficient Agents

With the work production of a maximally-efficient computational map established, we are poised to determine the work production for thermodynamically-efficient agents. Specifically, consider an agent parameterized by its logical architecture M and model parameters θ . As described by the agent architecture in Sec. IV A, the agent uses its memory X_j to map inputs Y_j to outputs Y'_j and to update its memory to X_{j+1} . In stochastic mapping the SOI $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}_j$ the model parameter θ provides an estimate of the distribution over the current initial state (X_j^θ, Y_j^θ) as well as the final state $(X_{j+1}^\theta, Y_{j'}^\theta)$. Assuming the agent's logical architecture M is executed optimally, direct application of Eq. (8) then says that the expected work production of the computational map $x_j y_j \rightarrow x_{j+1} y'_j$ is:

$$\langle W_{j, x_j y_j \rightarrow x_{j+1} y'_j}^\theta \rangle \equiv k_B T \ln \frac{\Pr(X_j^\theta = x, Y_j^\theta = y)}{\Pr(X_{j+1}^\theta = x', Y_{j'}^\theta = y')} . \quad (9)$$

In this, the estimated final distribution comes from the logical architecture updating the initial distribution:

$$\begin{aligned} \Pr(X_{j+1}^\theta = x', Y_{j'}^\theta = y') \\ = \sum_{x, y} \Pr(X_j^\theta = x, Y_j^\theta = y) M_{xy \rightarrow x' y'} . \end{aligned}$$

Equation (9)'s expression for work establishes that all functional aspects (logical operation and energetics) of an efficient agent are determined by two factors:

1. The logical architecture M that specifies how the agent manipulates inputs and updates its own memory.
2. The estimated input distribution $\Pr(X_j^\theta, Y_j^\theta)$ for which the agents' execution of M is optimized to minimize dissipation.

Thus, we define a *thermally-efficient agent* by the ordered pair $\{M, \Pr(X_j^\theta, Y_j^\theta)\}$.

So defined, we can calculate the work produced when such agents act on a particular input sequence $y_{0:L}$. This is done by first considering the work production of a particular sequence of agent memory states $x_{0:L+1}$ and

outputs $y'_{0:L}$:

$$\begin{aligned} \langle W_{|y_{0:L}, y'_{0:L}, x_{0:L+1}}^\theta \rangle &= \sum_{j=0}^{L-1} \langle W_{j, x_j y_j \rightarrow x_{j+1} y'_j}^\theta \rangle \\ &= k_B T \ln \prod_{j=0}^{L-1} \frac{\Pr(X_j^\theta = x_j, Y_j^\theta = y_j)}{\Pr(X_{j+1}^\theta = x_{j+1}, Y_{j+1}^{\prime\theta} = y'_j)} . \end{aligned}$$

Then, to obtain the average work produced from a particular input sequence $y_{0:L}$, we average over all possible hidden-state sequences and $x_{0:L+1}$ and output sequences $y'_{0:L}$:

$$\begin{aligned} \langle W_{|y_{0:L}}^\theta \rangle &= k_B T \sum_{x_{0:L+1}, y'_{0:L}} \Pr(Y'_{0:L} = y'_{0:L}, X_{0:L+1} = x_{0:L+1} | Y_{0:L} = y_{0:L}) \langle W_{|y_{0:L}, y'_{0:L}, x_{0:L+1}}^\theta \rangle \\ &= k_B T \sum_{x_{0:L+1}, y'_{0:L}} \Pr(X_0 = x_0) \prod_{k=0}^{L-1} M_{x_k, y_k \rightarrow x_{k+1}, y'_k} \ln \prod_{j=0}^{L-1} \frac{\Pr(X_j^\theta = x_j, Y_j^\theta = y_j)}{\Pr(X_{j+1}^\theta = x_{j+1}, Y_{j+1}^{\prime\theta} = y'_j)} . \end{aligned} \quad (10)$$

This gives the average energy harvested by an agent that transduces inputs $y_{0:L}$ according to the logical architecture M , given that it is designed to be as efficient as possible when its model θ matches the environment.

On its own, Eq. (10)'s work production is a deeply interesting quantity. In point of fact, since our agents are stochastic Turing machines [51], this is the work production for any general form of computation that maps inputs to output distributions $\Pr(Y'_{0:L} | Y_{0:L} = y_{0:L})$ [52]. Thus, Eq. (10) determines the possible *work benefit* for *universal thermodynamic computing*.

Given this general expression for work production, one might conclude that the next step for thermodynamic learning is to search for the agent tuple $\{M, \Pr(X_j^\theta, Y_j^\theta)\}$ that maximizes the work production. However, this strategy comes with two issues. First, it requires a wider search than necessary. Second, it does not draw a direct connection to the underlying model θ . Recall that we are considering ϵ -machine models θ of the input sequence that give the probability estimate $\Pr(Y_{0:L}^\theta = y_{0:L})$ for any L .

We address both of these issues by refining the search space to agents whose anticipated inputs $\Pr(X_j^\theta, Y_j^\theta)$ are explicitly determined by their initial state distribution

$\Pr(X_0^\theta)$ and estimated input process $\Pr(Y_{0:\infty}^\theta)$:

$$\begin{aligned} \Pr(X_j^\theta = x_j, Y_j^\theta = y_j) &= \sum_{x_{0:j}, y_{0:j}, y'_{0:j}} \Pr(X_0^\theta = x_0) \Pr(Y_{0:j+1}^\theta = y_{0:j+1}) \\ &\quad \times \prod_{k=0}^{j-1} M_{x_k, y_k \rightarrow x_{k+1}, y'_k} . \end{aligned}$$

We use this estimate for the initial state in Eq. (10), since it is maximally efficient, dissipating as little as possible if the agent architecture M receives the input distribution $\Pr(Y_{0:\infty}^\theta)$. As a result of its efficiency, the resulting computation performed by the agent produces the maximum possible work, given its logical architecture. This simplifies our search for maximum-work agents by directly tying the estimated inputs to the model θ . However, it still leaves one piece of the agent undefined: its logical architecture M . Fortunately, as we discuss now, the thermodynamics of modularity further simplifies the search.

E. Thermodynamics of Modularity

An agent transduces inputs to outputs through a series of modular operations. The Hamiltonian $\mathcal{H}_{\mathcal{X} \times \mathcal{Y}_j}(t)$ that governs the evolution of the j th operation is decoupled from the other elements of the time series $\mathcal{Y}_0 \times \mathcal{Y}_1 \cdots \mathcal{Y}_{j-1} \times \mathcal{Y}_{j+1} \times \cdots$. As a result of this modular

computational architecture, the correlations lost between the agent and the rest of the information reservoir are irreversibly dissipated, producing entropy. This is an energetic cost associated directly with the agent's logical architecture, known as the *modularity dissipation* [53]. It sets a minimum for the work dissipated in a complex computation composed of many elementary steps.

To continue our pursuit of maximum-work, we must design the agent's logical architecture to minimize dissipated work. Past analysis of agents that harness energy from a pattern $\Pr(Y_{0:\infty})$ showed that the modularity dissipation is only minimized when the agent's states are *predictive* of the pattern [53]. This means that to maximize work extracted, an agent's state must contain all information about the past relevant to the future $\Pr(Y_{j+1:L}|X_j) = \Pr(Y_{j+1:L}|Y_{0:j})$. That is, for maximal work extraction agent states must be sufficient statistics for predicting the future.

Moreover, the ϵ -machines introduced in Sec. IIIB are constructed with hidden states S_j that are a *minimal* predictor of their output process. This is why they are referred to as *causal states*. And so, the ϵ -machine is a minimal sufficient statistic for prediction. Transitions among causal states trigger outputs according to:

$$\theta_{s \rightarrow s'}^{(y)} = \Pr(Y_j^\theta = y, S_{j+1}^\theta = s' | S_j^\theta = s),$$

which is the probability that an ϵ -machine in internal state s transitions to s' and emits output y . ϵ -Machines have the additional property that they are unifilar, meaning that the next causal state s' is uniquely determined by the current state s and output y via the propagator function $s' = \epsilon(s, y)$.

In short, to produce maximum work agent memory must store at least the causal states of the environment's own ϵ -machine. Appendix D describes how the logical architecture of the maximum-work minimum-memory agent is determined by its estimated ϵ -machine model θ of its inputs. The agent states are chosen to be the same as the causal states $\mathcal{X} = \mathcal{S}$, and they update according to the propagator function ϵ :

$$M_{xy \rightarrow x'y'} = \frac{1}{|\mathcal{Y}_j|} \times \begin{cases} \delta_{x', \epsilon(x, y)}, & \text{if } \sum_{x'} \theta_{x \rightarrow x'}^{(y)} \neq 0, \\ \delta_{x', x}, & \text{otherwise.} \end{cases} \quad (11)$$

This guarantees that the agent's internal states follow the causal states of its estimated input process $\Pr(Y_{0:\infty}^\theta)$. In turn, it prevents the agent from dissipating temporal correlations within that process.

F. Agent-Model Equivalence

In the pursuit of maximum work, we refined the structure of candidate agents considerably, limiting consideration to those that minimize the modularity dissipation. As a result, they store the predictive states of their estimated input and their logical architecture is explicitly determined by an ϵ -machine model. Moreover, to be efficient, the candidate agent should begin in the ϵ -machine's start state s^* such that the model θ uniquely determines the second piece of an efficient agent. This is the estimated initial distribution over its memory state and input:

$$\Pr(Y_j^\theta = y_j, X_j^\theta = s_j) = \sum_{y_{0:j}, s_{0:j}, s_{j+1}} \delta_{s_0, s^*} \prod_{k=0}^j \theta_{s_k \rightarrow s_{k+1}}^{(y_k)}.$$

Conversely, maximum-work agents, characterized by their logical architecture and anticipated input distributions $\{M, \Pr(X_j^\theta, Y_j^\theta)\}$, also specify the ϵ -machine's model of their estimated distribution:

$$\begin{aligned} \theta_{s \rightarrow s'}^{(y)} &= \Pr(Y_j^\theta = y | S_j^\theta = s) \delta_{s', \epsilon(s, y)} \\ &= \Pr(Y_j^\theta = y | X_j^\theta = s) |\mathcal{Y}_j| M_{sy \rightarrow s'y'}. \end{aligned} \quad (12)$$

Through the ϵ -machine, the agent also specifies its estimated input process. In this way, we arrive at a class of agents that are uniquely determined by their environment model.

Figure 6 explicitly lays this out. It presents an agent that estimates a period-2 process with uncertain phase, such that $\Pr(Y_{0:\infty}^\theta = 0101\cdots) = \Pr(Y_{0:\infty}^\theta = 1010\cdots) = 0.5$. The middle column shows the ϵ -machine that is uniquely determined for that process, characterized by the model parameters $\theta_{s \rightarrow s'}^{(y)}$. The right column shows the unique minimal agent that harnesses as much work as possible from that process. All three, (i) the estimated process, (ii) the estimated ϵ -machine model, and (iii) the maximum-work agent are equivalent. And, they can be determined from one another. This holds true for *any* estimated input process $\Pr(Y_{0:\infty}^\theta)$.

Under the equivalence of model θ and agent operation, when we monitor the agent's thermodynamic performance through its work production, we also measure the predictive performance of its underlying model.

This completes the thermodynamic learning framework laid out in Fig. 1. There, the model an agent holds affects its interaction with the symbol sequence $y_{0:L}$ and, ultimately, its work production. And so, from this point forward, when discussing an estimated process or an ϵ -machine that generates that guess, we are also describing the unique thermodynamic agent designed to produce maximal work from the estimated process. We can

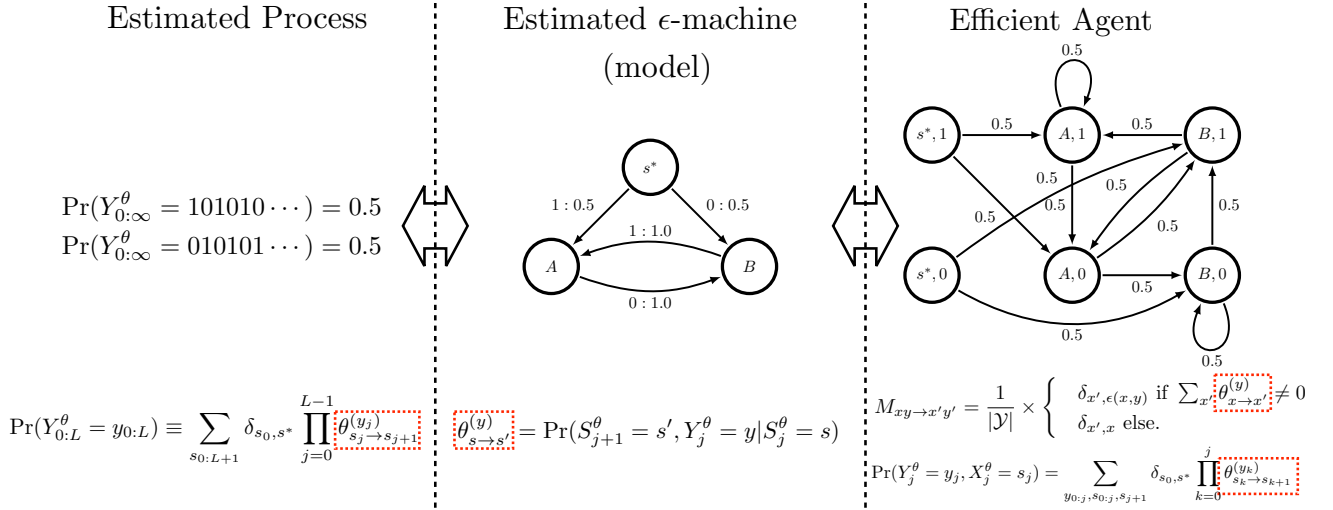


FIG. 6. Equivalence of estimated input process $\Pr(Y_{0:\infty}^\theta = y_{0:\infty})$, ϵ -machine θ , and the agent that efficiently harnesses the input process asymptotically, using logical architecture $M_{xy \rightarrow x'y'} = \Pr(X_{j+1} = x', Y_{j+1} = y' | X_j = x, Y_j = y)$ and estimated input distribution $\Pr(X_j^\theta = x, Y_j^\theta = y)$. Determining one determines the others.

now turn to explore how such agents' work production ties to their underlying models. A direct comparison to log-likelihood parametric-density estimation can now be drawn.

V. WORK-LIKELIHOOD CORRESPONDENCE FOR AGENT DESIGN

We are now ready to return to our core objective—exploring work production as a performance measure for a model estimated from a time series $y_{0:L}$. In comparison to the expression for general computing in Eq. (10), using efficiently-designed predictive agents leads to a much simpler expression for work production:

$$\langle W_{|y_{0:L}}^\theta \rangle = k_B T (\ln \Pr(Y_{0:L}^\theta = y_{0:L}) + L \ln |\mathcal{Y}|) . \quad (13)$$

The mechanism behind this vast simplification arises from unifilarity—a property of prediction machines that guarantees a single state trajectory on $x_{0:L}$ for each input string $y_{0:L}$. The details of the derivation are outlined in Appendix E.

This expression directly captures the relationship between work production and the agent's underlying model of the data. To see this, we recast it in the language of machine learning. Consider $y_{0:L}$ as training data in parametric density estimation. We are then tasked to construct a model of this data. Each candidate model is parameterized by θ , which results in an estimated process $Y_{0:L}^\theta$. Observe then that $\Pr(Y_{0:L}^\theta = y_{0:L})$ is simply the probability that the candidate model will output $y_{0:L}$. Therefore, the

log-likelihood $\ell(\theta|y_{0:L})$ of parametric estimation coincides with $\ln \Pr(Y_{0:L}^\theta = y_{0:L})$, and we can write:

$$\langle W_{|y_{0:L}}^\theta \rangle = k_B T \ell(\theta|y_{0:L}) + k_B T L \ln |\mathcal{Y}| . \quad (14)$$

One concludes that work production is maximized precisely when the log-likelihood $\ell(\theta|y_{0:L})$ is maximized. Thus, the criterion for *creating a good model of an environment* is the same as that for *extracting maximal work*.

This link is made concrete via the simple example presented in App. F. It goes through an explicit description of the Hamiltonian control required to implement a memoryless agent that harvests work from a sequence of up spins \uparrow and down spins \downarrow that compose the time series $y_{0:L}$. The agent's internal memoryless model results in Eq. (14)'s work production. And, we find that the maximum-work agent has learned about the input sequence. Specifically, the agent learns the frequency of spins \uparrow and \downarrow , confirming the basic principle of maximum-work thermodynamic learning. However, the learning presented in App. F precludes the possibility of learning temporal structure in the spin sequence, since the agents and their internal models have no memory [29]. To learn about the temporal correlations within the sequence, one must use agents with multiple memory states. We leave thermodynamic learning among memoryful agents for later investigation. Stepping back, we see the relationship between machine learning and information thermodynamics more clearly. In parametric density estimation we have:

1. Data $y_{0:L}$ that provides a window into a black box.

2. A model θ of the black box that determines an estimated distribution over the data $\Pr(Y_{0:L}^\theta)$.
3. A performance measure for the model of the data, given by the log-likelihood $\ell(\theta|y_{0:L}) = \ln \Pr(Y_{0:L}^\theta = y_{0:L})$.

The parallel in thermodynamic learning is exact, with:

1. Data $y_{0:L}$ physically stored in systems $\mathcal{Y}_0 \times \mathcal{Y}_1 \times \dots \mathcal{Y}_{L-1}$ output from the black box.
2. An agent $\{M, \Pr(X_j^\theta, Y_j^\theta)\}$ that is entirely determined by the model θ .
3. The agent's thermodynamic performance, given by its work production $\langle W_{|y_{0:L}}^\theta \rangle$, increases linearly with the log-likelihood $\ell(\theta|y_{0:L})$.

In this way, we see that thermodynamic learning through work maximization is *equivalent* to parametric density estimation.

Intuitively, the natural world is replete with complex learning systems—an observation seemingly at odds with Thermodynamics and its Second Law which dictates that order inevitably decays into disorder. However, our results are tantamount to a contravening physical principle that drives the emergence of order through learning: work maximization. We showed, in point of fact, that work maximization and learning are equivalent processes. At a larger remove, this hints of general physical principles of emergent organization.

VI. SEARCHING FOR PRINCIPLES OF ORGANIZATION

Introducing an *equivalence of maximum work production and optimal learning* comes at a late stage of a long line of inquiry into what kinds of thermodynamic constraints and laws govern the emergence of organization and, for that matter, biological life. So, let's historically place the seemingly-new principle. In fact, it enters a crowded field. Within statistical physics the paradigmatic principle of organization was found by Kirchhoff [54]: in electrical networks current distributes itself so as to dissipate the least possible heat for the given applied voltages. Generalizations, for equilibrium states, are then found in Gibbs' variational principle for entropy for heterogeneous equilibrium [55], Maxwell's principles of minimum-heat [56, pp. 407-408], and Onsager's minimizing the "rate of dissipation" [57].

Close to equilibrium, Prigogine introduced minimum entropy production [58], identifying *dissipative structures* whose maintenance requires energy [59]. However, far from equilibrium the guiding principles can be quite the

opposite. And so, the effort continues today, for example, with recent applications of nonequilibrium thermodynamics to pattern formation in chemical reactions [60]. That said, statistical physics misses at least two, related, but key components: dynamics of and information in thermal states.

Dynamical systems theory takes a decidedly mechanistic approach to the emergence of organization, analyzing the geometric structures in a system's state space that amplify fluctuations and eventually attenuate them into macroscopic behaviors and patterns. This was eventually articulated by pattern formation theory [61–63]. A canonical example is fluid turbulence [64]—a dynamical explanation for its complex organizations occupied much of the 70s and 80s. Landau's original theory of incommensurate oscillations was superseded by the mathematical discovery in the 1950s of chaotic attractors [65, 66]. This approach, too, falls short of leading to a principle of emergent organization. Patterns emerge, but what exactly are they and what complex behavior do they exhibit?

Answers to this challenge came from a decidedly different direction—Shannon's theory of noisy communication channels and his measures of information [67, 68], appropriately extended [69]. While adding an important new perspective—that organized systems store and transmit information—this, also, did not go far enough as it side-stepped the content and meaning of information [70]. Inroads to these appeared in the theory of computation inaugurated by Turing [71]. The most direct and ambitious approach to the role of information in organization, though, appeared in Wiener's cybernetics [4, 72]. While it eloquently laid out the goals to which principles should strive, it ultimately never harnessed the mathematical foundations and calculational tools needed. Likely, the earliest overt connection between statistical mechanics and information, though, appeared with Jaynes' Maximum Entropy [73] and Minimum Entropy Production Principles [74].

So, what is new today is the synthesis of statistical physics, dynamics, and information. This, finally, allows one to answer the question, How do physical systems store and process information? The answer is that they *intrinsically compute* [36]. With this, one can extract from behavior a system's information processing, even going so far as to discover the effective equations of motion [75–78]. One can now frame questions about how a physical system reacts to, controls, and adapts to its environment.

All such systems, however, are embedded in the physical world and require resources to operate. More to the point, what energetic resources underlie computation? Initiated by Brillouin [79] and Landauer and Bennett [19, 80], today there is a nascent *physics of information* [14, 81]. Resource constraints on computing by thermodynamic

systems are now expressed in a suite of new principles. For example, the *principle of requisite complexity* [29] dictates that maximally-efficient interactions require an agent’s internal organization match the environment’s organization. And, thermodynamic resource costs arise from the modularity of an agent’s architecture [53]. Pushing the search for organization further, the preceding established the thermodynamics of how a system learns, suggesting the possibility of adaptive organization.

To fully appreciate organization in natural processes, though, one must also address dynamics of agent populations, first on the time scale of agent life cycles and second on the scale of many generations. In fact, tracking the complexity of individuals reveals that selection pressures spontaneously emerge in purely-replicating populations [82] and replication itself necessarily dissipates energy [83].

As these pieces assembled, a picture has come into focus. Intelligent, adaptive systems learn to harness resources from their environment, expending energy to live and reproduce. Taken altogether, the historical perspective suggests we are moving close to realizing Wiener’s cybernetics [4].

VII. CONCLUSION

We introduced thermodynamic machine learning—a physical process that trains intelligent agents by maximizing work production from complex environmental stimuli supplied as time-series data. This involved constructing a framework to describe thermodynamics of computation at the single-shot level, enabling us to evaluate the work an agent can produce from individual data realizations. Key to the framework is its generality—applicable to agents exhibiting arbitrary adaptive input-output behavior and implemented within any physical substrate.

In the pursuit of maximum work, we refined this general class of agents to those that are best able to harness work from temporally-correlated inputs. We found that the performance of such maximum-work agents increases proportionally to the log-likelihood of the model they use for predicting their environment. As a consequence, our results show that thermodynamic learning exactly mimics parametric density estimation in machine learning. Thus, work is a thermodynamic performance measure for physically-embedded learning. This result further solidifies the connections between agency, intelligence, and the thermodynamics of information—hinting that energy harvesting and learning may be two sides of the same coin.

These connections suggest a number of exciting future directions. From the technological perspective, they

hint at a natural method for designing intelligent energy harvesters—establishing that our present tools of machine learning can be directly mapped to automated design of efficient information ratchets and pattern engines [29, 42, 84]. Meanwhile, recent results indicate that quantum systems can generate complex adaptive behaviors using fewer resources than classical counterparts [85–87]. Does this suggest there are new classes of quantum-enhanced energy harvesters and learners?

Ultimately, energy is an essential currency for life. This highlights the question, To what extent is work optimization a natural tendency of driven physical systems? Indeed, recent results indicate physical systems evolve to increase work production [32, 33], opening a fascinating possibility. Could the equivalence between work production and learning then indicate that the universe itself naturally learns? The fact that complex intelligent life emerged from the lifeless soup of the universe might be considered a continuing miracle: a string of unfathomable statistical anomalies strung together over eons. It would certainly be extraordinary if this evolution then has a physical basis—hidden laws of thermodynamic organization that guide the universe to create entities capable of extracting maximal work.

ACKNOWLEDGMENTS

The authors thank the Telluride Science Research Center for hospitality during visits and the participants of the Information Engines Workshops there. ABB thanks Wesley Boyd for useful conversations and JPC similarly thanks Adam Rupe. JPC acknowledges the kind hospitality of the Santa Fe Institute, Institute for Advanced Study at the University of Amsterdam, and California Institute of Technology for their hospitality during visits. This material is based upon work supported by, or in part by, Grant No. FQXi-RFP-IPW-1902 and FQXi-RFP-1809 from the Foundational Questions Institute and Fetzer Franklin Fund (a donor-advised fund of Silicon Valley Community Foundation), the Templeton World Charity Foundation Power of Information fellowship TWCF0337 and TWCF0560, the National Research Foundation, Singapore, under its NRFF Fellow program (Award No. NRF-NRFF2016-02), Singapore Ministry of Education Tier 1 Grants No. MOE2017-T1-002-043, and U.S. Army Research Laboratory and the U.S. Army Research Office under grants W911NF-18-1-0028 and W911NF-21-1-0048. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of National Research Foundation, Singapore.

During submission we became aware of related work: L.

Touzo, M. Marsili, N. Merhav, and E. Roldan. Optimal work extraction and the minimum description length principle. *arXiv:2006.04544*.

Appendix A: Extended Background

Developing the principle of maximum work production and calling out the physical benefits of an agent modeling its environment drew heavily from the areas of computational mechanics, nonequilibrium thermodynamics, and machine learning. Due to the variety of topics addressed, the following provides a more detailed notational and conceptual summary. This should aid the development be more self-contained, hopefully providing a common language across the areas and a foundation for further exploration. While we make suggestive comparisons by viewing foundations of each area side-by-side, it may be most appropriate for readers already familiar with them and concerned with novel results to skip, using the review to clarify unfamiliar notation.

1. Machine Learning and Generative Models

Thermodynamically, what is a good model of the data with which an agent interacts? Denote the data’s state space as $\mathcal{Z} = \{z\}$. If we have many copies of \mathcal{Z} , all initially prepared in the same way, then as we observe successive realizations $\vec{z} = \{z_0, z_1, \dots, z_N\}$ from an ensemble, the frequency of an observed state z approaches the *actual* probability distribution $\Pr(Z = z)$, where Z is the random variable that realizes states $z \in \mathcal{Z}$. However, with only a finite number N of realizations, the best that can be done is to characterize the environment with an *estimated* distribution $\Pr(Z^\theta = z)$. Estimating models that agree with finite data is the domain of statistical inference and machine learning of generative models [15, 34, 88].

At first blush, estimating a probability distribution appears distinct from familiar machine learning challenges, such as image classification and the inverse problem of artificially generating exemplar images from given data. However, both classification and prediction can be achieved through a form of unsupervised learning [18]. For instance, if the system is a joint variable over both the pixel images and the corresponding label $\mathcal{Z} = \text{pixels} \times \{\text{cat}, \text{dog}\}$, then our estimated distribution $\Pr(Z^\theta = z)$ gives both a means of choosing a label for an image $\Pr(\text{label}^\theta = \text{cat} | \text{pixels}^\theta = \text{image})$ and a means of choosing an image for a label $\Pr(\text{pixels}^\theta = \text{image} | \text{label}^\theta = \text{cat})$.

A generative model is specified by a set of parameters θ from which the model produces the estimated distribution

$\Pr(Z^\theta = z) = \Pr(Z = z | \Theta = \theta)$. The procedure of arriving at this estimated model is parametric density estimation [15, 88]. However, we take the random variable Z^θ for the estimated distribution to denote the model for notational and conceptual convenience. The Shannon entropy [68]:

$$H[Z] \equiv - \sum_z \Pr(Z = z) \ln \Pr(Z = z)$$

measures uncertainty in nats, a “natural” unit for thermodynamic entropies. The Shannon entropy easily extends to joint probabilities and all information measures that come from their composition (conditional and mutual informations). For instance, if the environment is composed of two correlated subcomponents $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, the probability and entropy are expressed:

$$\begin{aligned} \Pr(Z = z) &= \Pr(X = x, Y = y) \\ &\neq \Pr(X = x) \Pr(Y = Y) \text{ and} \\ H[Z] &= H[X, Y] \\ &\neq H[X] + H[Y] , \end{aligned}$$

respectively.

While there are many other ways to create parametric models θ —from polynomial functions with a small number of parameters to neural networks with thousands [15]—the goal is to match as well as possible the estimated distribution $\Pr(Z^\theta)$ to the actual distribution $\Pr(Z)$.

One measure of success in this is the probability that the model generated the data—the *likelihood*. The likelihood of the model θ given a data point z is the same as the likelihood of Z^θ :

$$\begin{aligned} \mathcal{L}(\theta|z) &= \Pr(Z = z | \Theta = \theta) \\ &= \Pr(Z^\theta = z) \\ &= \mathcal{L}(Z^\theta|z) . \end{aligned}$$

Given a set $\vec{z} = \{z_1, z_2, \dots, z_N\}$ of training data and assuming independent samples, then the likelihood of the model is the product:

$$\mathcal{L}(Z^\theta|\vec{z}) = \prod_{i=1}^N \mathcal{L}(Z^\theta|z_i) . \quad (\text{A1})$$

This is a commonly used performance measure in machine learning, where algorithms search for models with maximum likelihood [15]. However, it is common to use the *log-likelihood* instead, which is maximized for the same

models:

$$\begin{aligned}\ell(\theta|\vec{z}) &= \ln \mathcal{L}(\theta|\vec{z}) \\ &= \sum_{i=1}^N \ln \Pr(Z^\theta = z_i) .\end{aligned}\quad (\text{A2})$$

If the model Z^θ were specified by a neural network, the log-likelihood could be determined through stochastic gradient descent back-propagation [34, 89], for instance. The intention is that the procedure will converge on a network model that produces the data with high probability.

2. Thermodynamics of Information

Learning from data translates information in an environment into a useful model. What makes that model useful? In a physical setting, recalling from Landauer that “information is physical” [90], the usefulness one can extract from thermodynamic processes is work. Figure 3 shows a basic implementation for physical computation. Such an information-storing physical system $\mathcal{Z} = \{z\}$, in contact with a thermal reservoir, can execute useful computations by drawing energy from a work reservoir. Energy flowing from the system \mathcal{Z} into the thermal reservoir is positive heat Q . When energy flows from the system \mathcal{Z} to the work reservoir, it is positive work W production. Work production quantifies the amount of energy that is stored in the work reservoir available for later use. And so, in this telling, it represents a natural and physically-motivated measure of thermodynamic performance. In the framework for thermodynamic computation of Fig. 3, work is extracted via controlling the system’s Hamiltonian.

Specifically, the system’s informational states are controlled via a time-dependent Hamiltonian—energy $E(z, t)$ of state z at time t . For state trajectory $z_{\tau:\tau'} = z_\tau z_{\tau+dt} \cdots z_{\tau'-dt} z_{\tau'}$ over time interval $t \in [\tau, \tau']$, the resulting work extracted by Hamiltonian control is the temporally-integrated change in energy [37]:

$$W|_{z_{\tau:\tau'}} = - \int_{\tau}^{\tau'} dt \partial_t E(z, t)|_{z=z_t} .$$

Heat $Q|_{z_{\tau:\tau'}} = E(z_\tau, \tau) - E(z_{\tau'}, \tau') - W|_{z_{\tau:\tau'}}$ flows into the thermal reservoir, increasing its entropy:

$$\Delta S|_{z_{\tau:\tau'}}^{\text{reservoir}} = \frac{Q|_{z_{\tau:\tau'}}}{T} . \quad (\text{A3})$$

where the thermal reservoir is at temperature T . The Second Law of Thermodynamics states that, on average, any processing on the informational states can only yield nonnegative entropy production of the universe (reservoir

and system \mathcal{Z}):

$$\begin{aligned}\langle \Sigma \rangle &= \langle \Delta S^{\text{reservoir}} \rangle + \langle \Delta S^{\mathcal{Z}} \rangle \\ &\geq 0 .\end{aligned}\quad (\text{A4})$$

This constrains the energetic cost of computations performed within the system \mathcal{Z} .

A computation over time interval $t \in [\tau, \tau']$ has two components:

1. An initial distribution over states $\Pr(Z_\tau = z_\tau)$, where Z_t is the random variable of system \mathcal{Z} at time t .
2. A Markov channel that transforms it, specified by the conditional probability of the final state $z_{\tau'}$ given the initial input z_τ :

$$M_{z_\tau \rightarrow z_{\tau'}} = \Pr(Z_{\tau'} = z_{\tau'} | Z_\tau = z_\tau) .$$

This specifies, in turn, the final distribution $\Pr(Z_{\tau'} = z_{\tau'})$ that allows direct calculation of the system-entropy change [44]:

$$\Delta S|_{z_{\tau:\tau'}}^{\mathcal{Z}} = k_B \ln \frac{\Pr(Z_\tau = z_\tau)}{\Pr(Z_{\tau'} = z_{\tau'})} .$$

Adding this to the information reservoir’s entropy change yields the entropy production of the universe. This can also be expressed in terms of the work production:

$$\begin{aligned}\Sigma|_{z_{\tau:\tau'}} &\equiv \Delta S|_{z_{\tau:\tau'}}^{\text{reservoir}} + \Delta S|_{z_{\tau:\tau'}}^{\mathcal{Z}} \\ &= \frac{-W|_{z_{\tau:\tau'}} + \phi(z_\tau, \tau) - \phi(z_{\tau'}, \tau')}{T} .\end{aligned}$$

Here, $\phi(z, t) = E(z, t) + k_B T \ln \Pr(Z_t = z)$ is the *pointwise nonequilibrium free energy*, which becomes the *nonequilibrium free energy* when averaged: $\langle \phi(z, t) \rangle_{\Pr(Z_t = z)} = F^{\text{neq}}(t)$ [43].

Note that the entropy production is also proportional to the additional work that could have been extracted if the computation was efficient. This is referred to as the *dissipated work*:

$$W|_{z_{\tau:\tau'}}^{\text{diss}} = T \Sigma|_{z_{\tau:\tau'}} . \quad (\text{A5})$$

Turning back to the Second Law of Thermodynamics, we see that the *average* work extracted is bounded by the change in nonequilibrium free energy:

$$\langle W \rangle \geq F^{\text{neq}}(\tau') - F^{\text{neq}}(\tau) .$$

When the system starts and ends as an information reservoir, with equal energies for all states $E(z, \tau) = E(z', \tau')$ [37], this reduces to Landauer’s familiar principle for era-

sure [19]—work production must not exceed the change in state uncertainty:

$$\langle W \rangle \leq k_B T (H[Z_{\tau'}] - H[Z_{\tau}]) ,$$

where $H[Z_t] = -\sum_{z \in \mathcal{Z}} \Pr(Z_t = z) \ln \Pr(Z_t = z)$ is the Shannon entropy of the system at time t measured in nats. This is the starting point for determining the work production that agents can extract from data.

3. Computational Mechanics

When describing thermodynamics and machine learning, data was taken from the state space \mathcal{Z} all at once. However, what if we consider a state space composed of L identical components $\mathcal{Z} = \mathcal{Y}^L$ that are received in sequence. Our model of the time series $y_{0:L}$ of realizations is described by an estimated distribution $\Pr(Y_{0:L}^\theta = y_{0:L})$. However, for L large enough, this object becomes impossible to store, due to the exponential increase in the number of sequences. Fortunately, there are ways to generally characterize an arbitrarily long time-series distribution using a finite model.

a. Generative Machines

A hidden Markov model (HMM) is described by a set of hidden states \mathcal{S} , a set of output states \mathcal{Y} , a conditional output-labeled matrix that gives the transition probabilities between the states:

$$\theta_{s \rightarrow s'}^{(y)} = \Pr(S_{j+1}^\theta = s', Y_j^\theta = y | S_j^\theta = s)$$

for all j , and a start state $s^* \in \mathcal{S}$. (Generally, one also specifies an initial state distribution, with selecting a start state being a special case.) We label the transition probabilities with the model parameter θ , since these are the actual parameters that must be stored to generate probabilities of time series. For instance, Fig. 2 shows an HMM that generates a periodic process with uncertain phase. Edges between hidden states s and s' are labeled $y : \theta_{s \rightarrow s'}^{(y)}$, where y is the output symbol and $\theta_{s \rightarrow s'}^{(y)}$ is the probability of emitting that symbol on that transition. If $\{\theta_{s \rightarrow s'}^{(y)}\}$ is the model for the estimated input $\Pr(Y^\theta = y_{0:L})$, then the probability of any word is calculated by taking the product of transition matrices and summing over internal states:

$$\Pr(Y_{0:L}^\theta = y_{0:L}) = \sum_{s_{0:L+1}} \delta_{s_0, s^*} \prod_{j=0}^{L-1} \theta_{s_j \rightarrow s_{j+1}}^{(y_j)} .$$

Beyond generating length- L symbol strings, these HMMs

generate distributions over semi-infinite strings $\Pr(Y_{0:\infty}^\theta = y_{0:\infty})$. As such, they allow us to anticipate more than just the first L symbols from the same source. Once we have a model $\theta = \{(\theta_{s \rightarrow s'}^{(y)}, s, s', y)\}_{s, s', y}$ from our training data $y_{0:L}$, we can calculate probabilities of longer words $\Pr(Y_{0:L'}^\theta = y'_{0:L'})$ and, thus, the probability of symbols following the training data $\Pr(Y_{L:L'}^\theta = y'_{L:L'} | Y_{0:L}^\theta = y_{0:L})$. Distributions over semi-infinite strings $\Pr(Y_{0:\infty}^\theta = y_{0:\infty})$ are similar to processes, which are distributions over bi-infinite strings $\Pr(Y_{-\infty:\infty}^\theta = y_{-\infty:\infty})$. While not insisting on stationarity, and so allowing for flexibility in using subwords of length L , we can mirror computational mechanics' construction of unifilar HMMs from time series, where the hidden states S_j^θ are minimal sufficient statistics of the past $Y_{0:j}^\theta$ about the future $Y_{j:\infty}^\theta$ [91]. In other words, the hidden states are perfect predictors.

Given a semi-infinite process $\Pr(Y_{0:\infty}^\theta = y_{0:\infty})$, we construct a minimal predictor through a causal equivalence relation $y_{0:k} \sim y'_{0:j}$ that says two histories $y_{0:k}$ and $y'_{0:j}$ are members of the same equivalence class if and only if they have the same semi-infinite future distribution:

$$\Pr(Y_{j:\infty}^\theta = y''_{0:\infty} | Y_{0:j}^\theta = y'_{0:j}) = \Pr(Y_{k:\infty}^\theta = y''_{0:\infty} | Y_{0:k}^\theta = y_{0:k}) .$$

An equivalence class of histories is a *causal state*. Causal states also induce a map $\epsilon(\cdot)$ from histories $y_{0:k}$ to states s_i :

$$\begin{aligned} s_i &= \{y'_{0:j} | y_{0:k} \sim y'_{0:j}\} \\ &\equiv \epsilon(y_{0:k}) . \end{aligned}$$

This guarantees that a causal state is a sufficient statistic of the past about the future, such that we can track it as a perfect predictor:

$$\Pr(Y_{k:\infty}^\theta | Y_{0:k}^\theta = y_{0:k}) = \Pr(Y_{k:\infty}^\theta | S_k^\theta = \epsilon(y_{0:k})) .$$

In fact, the causal states are minimal sufficient statistics. Constructing causal states reveals a number of properties of stochastic processes and models. One of these is *unifilarity*, which means that if the current causal state $s_k = \epsilon(y_{0:k})$ is followed by any sequence $y_{k:j}$, then the resulting causal state $s_k = \epsilon(y_{0:k})$ is uniquely determined. And, we can expand our use of the ϵ function to include updating a causal state:

$$\begin{aligned} s_j &= \epsilon(s_k, y_{k:j}) \\ &\equiv \{y'_{0:l} | \exists y_{0:k} \ni s_k = \epsilon(y_{0:k}) \text{ and } y'_{0:l} \sim y_{0:j}\} . \end{aligned}$$

This is the set of histories $y'_{0:l}$ that predict the same future as a history $y_{0:j}$ which leads to causal state s_k via the initial sequence $y_{0:k}$ and then follows with the sequence $y_{k:j}$.

Unifilarity is key to deducing several useful properties of the HMM $\theta_{s \rightarrow s'}^{(y)}$, which we will refer to as a *nonstationary* ϵ -machine. First, unifilarity implies that for any causal state s followed by a symbol y , there is a unique next state $s' = \epsilon(s, y)$, meaning that the symbol-labeled transition matrix can be written:

$$\theta_{s \rightarrow s'}^{(y)} = \theta_{s \rightarrow \epsilon(s, y)}^{(y)} \delta_{s', \epsilon(s, y)} .$$

Moreover, the ϵ -machine's form is uniquely determined from the semi-infinite process:

$$\theta_{s \rightarrow s'}^{(y)} = \Pr(Y_j^\theta = y | S_j^\theta = s) \delta_{s', \epsilon(s, y)} ,$$

where the conditional probability is determined from the process:

$$\Pr(Y_j^\theta = y | S_j^\theta = \epsilon(y_{0:j})) = \Pr(Y_j^\theta = y | Y_{0:j}^\theta = y_{0:j}) .$$

Once constructed, the ϵ -machine allows us to reconstruct word probabilities via the simple product:

$$\Pr(Y_{0:L}^\theta = y_{0:L}) = \prod_{j=0}^{L-1} \theta_{\epsilon(s^*, y_{0:j}) \rightarrow \epsilon(s^*, y_{0:j+1})}^{(y_j)} ,$$

where $y_{0:0}$ denotes the null word, taking a causal state to itself under the causal update $\epsilon(s, y_{0:0}) = s$.

Allowing for arbitrarily-many causal states, our class of models (nonstationary ϵ -machines) is so general that it can represent *any* semi-infinite process and, thus, any distribution over sequences \mathcal{Y}^L . One concludes that computational mechanics provides an ideal class of generative models to fit to data $y_{0:L}$. *Bayesian structural inference* implements just this [92].

In these ways, computational mechanics already had solved (and several decades prior) the unsupervised learning challenge recently posed by Ref. [93] to create an ‘‘AI Physicist’’: a machine that learns regularities in time series to make predictions of the future from the past [94].

b. Input-Output Machines

This way one constructs a predictive HMM that generates a desired semi-infinite process $\Pr(Y_{0:L}^\theta = y_{0:L})$. The generalization to ϵ -transducers allows for an *input* as well as an *output* process—a transformation between processes [52]. The transducer at the i th time step is described by transitions among the hidden states $X_i \rightarrow X_{i+1}$, that are conditioned on the input Y_i , emitting the output Y'_i :

$$M_{x \rightarrow x'}^{(y'|y)} = \Pr(Y'_i = y', X_{i+1} = x' | Y_i = y, X_i = x) .$$

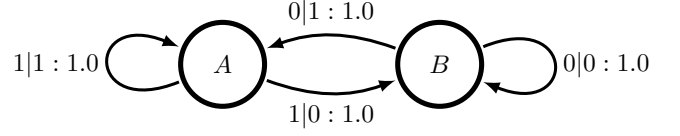


FIG. 7. The delay channel ϵ -transducer: The last input symbol is stored in its memory (states). If the last symbol was 1, then the corresponding transitions, labeled $y'|1 : 1.0$, update the hidden state to A. Then, all outputs from A are symbol 1. Similarly, input 0 leads to state B, whose corresponding outputs are all 0. In this way, the delay channel outputs the previous input symbol.

ϵ -Transducer state-transition diagrams label the edges of transitions between hidden states $y'|y : M_{x \rightarrow x'}^{(y'|y)}$. As Fig. 7 shows this is to be read as the probability $M_{x \rightarrow x'}^{(y'|y)}$ of output y' and next hidden state x' given input y and current hidden state x .

These devices are memoryful channels, with their memory encoded in the hidden states X_i . They implement a wide variety of functional operations. Figure 7 shows the *delay channel*. With sufficient memory, though, an ϵ -transducer can implement a universal Turing machine [51]. Moreover, if the input and output alphabets are the same, then they represent the form of a physical information ratchet, which have energetic requirements that arise from the thermodynamics of their operation [30, 31]. Since these physically-implementable information processors are so general in their ability to compute, they represent a very broad class of physical agents. As such, we use the framework of information ratchets to explore the functionality of agents that process information as a fuel.

Appendix B: Proof of Zero Entropy Production of Trajectories

Perfectly-efficient agents dissipate zero work and generate zero entropy $\langle \Sigma \rangle = 0$. The Crooks fluctuation theorem and other detailed fluctuation theorems say that entropy production is proportional to the log-ratio of probabilities [45, 46]:

$$\Sigma_{|z_{\tau:\tau'}} = k_B \ln \frac{\rho_F(\Sigma_{|z_{\tau:\tau'}})}{\rho_R(-\Sigma_{|z_{\tau:\tau'}})} ,$$

where $\rho_F(\Sigma_{|z_{\tau:\tau'}})$ is the probability of the entropy production under the protocol that controls the ratchet and $\rho_R(-\Sigma_{|z_{\tau:\tau'}})$ is the probability of minus that same entropy production if the control protocol is reversed. Thus, the average entropy production is proportional to the relative

entropy between these two distributions [68]:

$$\begin{aligned} \langle \Sigma \rangle &= k_B \sum_{z_\tau, \tau'} \rho_F(\Sigma|_{z_\tau, \tau'}) \ln \frac{\rho_F(\Sigma|_{z_\tau, \tau'})}{\rho_R(-\Sigma|_{z_\tau, \tau'})} \\ &\equiv k_B D_{KL}(\rho_F(\Sigma|_{z_\tau, \tau'}) || \rho_R(-\Sigma|_{z_\tau, \tau'})) . \end{aligned}$$

If the control is thermodynamically efficient, this relative entropy vanishes [95], implying the necessary and sufficient condition that $\rho_F(\Sigma) = \rho_R(-\Sigma)$. This then implies that all paths produce zero entropy:

$$\Sigma|_{z_\tau, \tau'} = 0.$$

Entropy fluctuations vanish as the entropy production goes to zero.

Appendix C: Thermodynamically Efficient Markov Channels

Given a physical system $\mathcal{Z} = \{z\}$, a computation on its states is given by a Markov channel $M_{z \rightarrow z'} = \Pr(Z_{\tau'} = z' | Z_\tau = z)$ and an input distribution $\Pr(Z_\tau = z)$. The following describes a quasistatic thermodynamic control that implements this computation efficiently if the input distribution matches the estimated distribution $\Pr(Z^\theta = z)$. This means that the work production is equal to the change in pointwise nonequilibrium free energy:

$$\begin{aligned} \langle W_{|z_\tau, z_{\tau'}}^\theta \rangle &= \phi(z_{\tau'}, \tau') - \phi(z_\tau, \tau) \\ &= \Delta E_{\mathcal{Z}} + k_B T \ln \frac{\Pr(Z_{\tau'}^\theta = z_{\tau'})}{\Pr(Z_\tau^\theta = z_\tau)} . \end{aligned} \quad (\text{C1})$$

Note that, while $\Pr(Z_\tau^\theta = z)$ is the input distribution for which the computation is efficient, it is possible that other input distributions $\Pr(Z_\tau = z)$ yield zero entropy production as well. They are only required to minimize $D_{KL}(Z_\tau || Z_\tau^\theta) - D_{KL}(Z_{\tau'} || Z_{\tau'}^\theta) = 0$ [49].

The physical setting that we take for thermodynamic control is overdamped Brownian motion with a controllable energy landscape. This is described by detailed-balanced rate equations. However, if our physical state-space is limited to \mathcal{Z} , then not all channels can be implemented with continuous-time rate equations [48]. Fortunately, this can be circumvented by additional ancillary or hidden states [48, 96]. And so, to implement any possible channel, we add an ancillary copy of our original system \mathcal{Z}' , such that our entire physical system is $\mathcal{Z}_{\text{total}} = \mathcal{Z} \times \mathcal{Z}'$.

Prescriptions have been given that efficiently implement any computation, specified by a Markov channel $M_{z_\tau \rightarrow z_{\tau'}} = \Pr(Z_{\tau'} = z_{\tau'} | Z_\tau = z_\tau)$, using quasistatic manipulation of the \mathcal{Z} 's energy levels and an ancillary copy \mathcal{Z}' [42, 53]. However, these did not determine the work

production for individual computational maps $z_\tau \rightarrow z_{\tau'}$ during the computation interval (τ, τ') .

The following implements an analogous form of quasistatic computation that allows us to easily calculate the energy associated with implementing the computation $M_{z_\tau \rightarrow z_{\tau'}}$, assuming the subsystem \mathcal{Z} started in z_τ and ends in $z_{\tau'}$. Due to detailed balance, the rate equation dynamics over the computational system and its ancillary copy $\mathcal{Z}_{\text{total}} = \mathcal{Z} \times \mathcal{Z}'$ are partially specified by the energy $E(z, z', t)$ of system state z and ancillary state z' at time t . This also uniquely specifies the equilibrium distribution:

$$\Pr(Z_t^{\text{eq}} = z, Z_t'^{\text{eq}} = z') = \frac{e^{-E(z, z', t)/k_B T}}{\sum_{z, z'} e^{-E(z, z', t)/k_B T}} .$$

The normalization constant $\sum_{z, z'} e^{-E(z, z', t)/k_B T}$ is the partition function that determines the equilibrium free energy:

$$F^{\text{eq}}(t) = -k_B T \ln \left(\sum_{z, z'} e^{-E(z, z', t)/k_B T} \right) .$$

The equilibrium free energy adds to the system energy. It is constant over the states:

$$E(z, z', t) = F^{\text{eq}}(t) - k_B T \ln \Pr(Z_t^{\text{eq}} = z, Z_t'^{\text{eq}} = z') .$$

We leverage the relationship between energy and equilibrium probability to design a protocol that achieves the work production given by Eq. (C2) for a Markov channel M . The estimated distribution over the whole space assumes that the initial distribution of the ancillary variable is uncorrelated and uniformly distributed:

$$\Pr(Z_\tau^\theta, Z_\tau') = \frac{\Pr(Z_\tau^\theta)}{|\mathcal{Z}|} .$$

Assuming the default energy landscape is constant initially and finally— $E(z, z', \tau) = E(z, z', \tau') = \xi$ —the maximally efficient protocol over the interval $[\tau, \tau']$ decomposes into five epochs, see Fig. 8:

1. Quench: $[\tau, \tau^+]$,
2. Quasistatically evolve: $(\tau, \tau_1]$,
3. Swap: $(\tau_1, \tau_2]$,
4. Quasistatically evolve: (τ_2, τ') , and
5. Reset: $[\tau'^-, \tau']$.

For all protocol epochs, except for Epoch 3 during which the two subsystems are swapped, \mathcal{Z} is held fixed while the ancillary system \mathcal{Z}' follows the local equilibrium distribution. Let's detail these in turn.

1. *Quench*: Instantaneously quench the energy from $E(z, z', \tau) = \xi$ to $E(z, z', \tau^+) = k_B T \ln(|\mathcal{Z}| / \Pr(Z_\tau = z))$

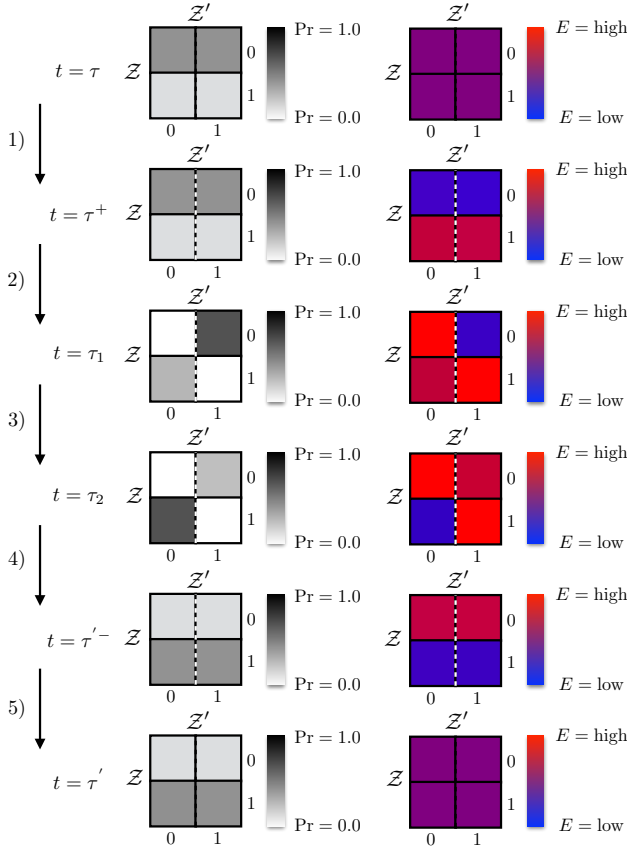


FIG. 8. Quasistatic agent implementing the Markov chain $M_{z_\tau \rightarrow z'_\tau}$ in the system \mathcal{Z} over the time interval $[\tau, \tau']$ using ancillary copy \mathcal{Z}' in five steps: Epoch 1: Energy landscape is instantaneously brought into equilibrium with the distribution over the joint system. Epoch 2: Probability flows in the ancillary system \mathcal{Z}' as the energy landscape quasistatically changes to make the conditional probability distribution in \mathcal{Z}' reflect the Markov channel $\Pr(Z'_{\tau_1} = z' | Z_{\tau_1} = z) = M_{z \rightarrow z'}$. Epoch 3: Systems \mathcal{Z} and \mathcal{Z}' are swapped. Epoch 4: Ancillary system quasistatically reset to the uniform distribution. Epoch 5: Energy landscape instantaneously reset to uniform.

over the infinitesimal time interval $[\tau, \tau^+]$ such that, if the distribution was as we expect, it would be in equilibrium $\Pr(Z_\tau^{\text{eq}}, Z_\tau^{\text{eq}}) = \Pr(Z_\tau^\theta) / |\mathcal{Z}|$. If the system started in z_τ , then the associated work produced is opposite the energy change:

$$\begin{aligned} \langle W_{|z_\tau, z_{\tau'}}^{\theta, 1} \rangle &= E(z_\tau, z', \tau) - E(z_\tau, z', \tau^+) \\ &= \xi + k_B T \ln \frac{\Pr(Z_\tau^\theta = z_\tau)}{|\mathcal{Z}|} . \end{aligned}$$

$\langle W_{|z_\tau, z_{\tau'}}^{\theta, 1} \rangle$ denotes that the work is produced in Epoch 1, conditioned on the estimated distributions Z_τ^θ and $Z_{\tau'}^\theta$, and initial and final states z_τ and $z_{\tau'}$. Note that we also condition on $Z_{\tau'}^\theta = z_{\tau'}$, since work production in this phase is unaffected by the computation's end state.

2. *Quasistatically evolve:* Quasistatically evolve the energy landscape over a third of total time interval $(\tau, \tau_1]$ such that the joint system remains in equilibrium and the ancillary system \mathcal{Z}' is determined by the Markov channel M applied to the system \mathcal{Z} :

$$\begin{aligned} \Pr(Z_{\tau_1} = z, Z'_{\tau_1} = z') &= \Pr(Z_\tau = z) M_{z \rightarrow z'} \\ E(z, z', \tau_1) &= -k_B T \ln \Pr(Z_\tau^\theta = z) M_{z \rightarrow z'} . \end{aligned}$$

Also, hold the energy barriers between states in \mathcal{Z} high, preventing probability flow between states and preserving the distribution $\Pr(Z_t) = \Pr(Z_\tau)$ for all $t \in (\tau, \tau_1]$.

Given that the system started in $Z_\tau = z_\tau$, the work production during this epoch corresponds to the average change in energy:

$$\begin{aligned} \langle W_{|z_\tau, z_{\tau'}}^{\theta, 2} \rangle &= - \sum_{z, z'} \int_{\tau^+}^{\tau_1} dt \Pr(Z'_t = z', Z_t = z | Z_\tau = z_\tau) \partial_t E(z, z', t) . \end{aligned}$$

As the system \mathcal{Z} remains in z_τ over the interval:

$$\Pr(Z'_t = z', Z_t = z | Z_\tau = z_\tau) = \Pr(Z'_t = z' | Z_t = z) \delta_{z, z_\tau}$$

and the work production simplifies to:

$$\begin{aligned} \langle W_{|z_\tau, z_{\tau'}}^{\theta, 2} \rangle &= - \sum_{z'} \int_{\tau^+}^{\tau_1} dt \Pr(Z'_t = z' | Z_t = z_\tau) \partial_t E(z_\tau, z', t) . \end{aligned}$$

We can express the energy in terms of the estimated equilibrium probability distribution:

$$E(z_\tau, z', t) = -k_B T \ln \Pr(Z'_t = z' | Z_t = z_\tau) \Pr(Z_t^\theta = z_\tau) .$$

And, since the distribution over the system \mathcal{Z} is fixed during this interval:

$$\begin{aligned} \Pr(Z'_t = z' | Z_t = z_\tau) \Pr(Z_t^\theta = z_\tau) &= \Pr(Z'_t = z' | Z_t = z_\tau) \Pr(Z_\tau^\theta = z_\tau) . \end{aligned}$$

Plugging these into the expression for the work production, we find that the evolution happens without energy

exchange:

$$\begin{aligned}
& \left\langle W_{|z_\tau, z_{\tau'}}^{\theta, 2} \right\rangle \\
&= -k_B T \int_{\tau^+}^{\tau_1} dt \sum_{z'} \Pr(Z'_t = z' | Z_t = z_\tau) \\
&\quad \times \partial_t \ln \Pr(Z'_t = z' | Z_t = z_\tau) \Pr(Z_\tau^\theta = z_\tau) \\
&= -k_B T \int_{\tau^+}^{\tau_1} dt \sum_{z'} \Pr(Z'_t = z' | Z_t = z_\tau) \\
&\quad \times \frac{\Pr(Z_\tau^\theta = z_\tau) \partial_t \Pr(Z'_t = z' | Z_t = z_\tau)}{\Pr(Z'_t = z' | Z_t = z_\tau) \Pr(Z_\tau^\theta = z_\tau)} \\
&= -k_B T \int_{\tau^+}^{\tau_1} dt \sum_{z'} \partial_t \Pr(Z'_t = z' | Z_t = z_\tau) \\
&= -k_B T \int_{\tau^+}^{\tau_1} dt \partial_t \sum_{z'} \Pr(Z'_t = z' | Z_t = z_\tau) \\
&= -k_B T \int_{\tau^+}^{\tau_1} dt \partial_t 1 \\
&= 0 .
\end{aligned}$$

After this state, at time τ_1 the resulting joint distribution over the ancillary and primary system matches the desired computation:

$$\Pr(Z_{\tau_1} = z, Z'_{\tau_1} = z') = \Pr(Z_\tau = z) M_{z \rightarrow z'} . \quad (\text{C2})$$

3. *Swap*: Over time interval $(\tau_1, \tau_2]$, efficiently swap the two systems $\mathcal{Z} \leftrightarrow \mathcal{Z}'$, such that $\Pr(Z_{\tau_1} = z, Z'_{\tau_1} = z') = \Pr(Z_{\tau_2} = z', Z'_{\tau_2} = z)$ and $E(z, z', \tau_1) = E(z', z, \tau_2)$. This operation requires zero work as well, as it is reversible, regardless of where the system starts or ends:

$$\left\langle W_{|z_\tau, z_{\tau'}}^{\theta, 3} \right\rangle = 0 .$$

Such an efficient swap operation has been demonstrated in finite time [48]. The resulting joint distribution over the ancillary and primary system matches a flip of the desired computation:

$$\Pr(Z_{\tau_2} = z, Z'_{\tau_2} = z') = \Pr(Z_\tau = z') M_{z' \rightarrow z} .$$

4. *Quasistatically evolve*: Over time interval (τ_2, τ') , quasistatically evolve the energy landscape from:

$$E(z, z', \tau_2) = -k_B T \ln \Pr(Z_\tau^\theta = z') M_{z' \rightarrow z}$$

to

$$\begin{aligned}
E(z, z', \tau') &= -k_B T \ln \frac{\sum_{z''} \Pr(Z_\tau^\theta = z'') M_{z'' \rightarrow z}}{|\mathcal{Z}|} \\
&= -k_B T \ln \frac{\Pr(Z_\tau^\theta = z)}{|\mathcal{Z}|} .
\end{aligned}$$

We keep the primary system \mathcal{Z} fixed as in Epoch 2. And, as in Epoch 2, there is zero work production:

$$\left\langle W_{|z_\tau, z_{\tau'}}^{\theta, 4} \right\rangle = 0 .$$

The result is that the primary system is in the desired final distribution:

$$\Pr(Z_{\tau'} = z) \equiv \sum_{z'} \Pr(Z_\tau = z') M_{z' \rightarrow z} ,$$

having undergone a mapping from its original state at time τ , while the ancillary system has returned to an uncorrelated uniform distribution.

5. *Reset*: Finally, over time interval $[\tau', \tau']$ instantaneously change the energy to the default flat landscape $E(z, z', \tau') = \xi$. Given that the system ends in state $z_{\tau'}$, the associated work production is:

$$\begin{aligned}
\left\langle W_{|z_\tau, z_{\tau'}}^{\theta, 5} \right\rangle &= E(z'_\tau, z', \tau'^-) - E(z'_\tau, z', \tau') \\
&= -\xi - k_B T \ln \frac{\Pr(Z_{\tau'}^\theta = z_{\tau'})}{|\mathcal{Z}|} .
\end{aligned}$$

The net work production given the initial state z_τ and final state $z_{\tau'}$ is then:

$$\begin{aligned}
\left\langle W_{|z_\tau, z_{\tau'}}^\theta \right\rangle &= \sum_{i=1}^5 \left\langle W_{|z_\tau, z_{\tau'}}^{\theta, i} \right\rangle \\
&= k_B T \ln \frac{\Pr(Z_\tau^\theta = z_\tau)}{\Pr(Z_{\tau'}^\theta = z_{\tau'})} .
\end{aligned}$$

Thus, when we average over all possible inputs and outputs and the estimated and actual distributions are the same $Z_\tau^\theta = Z_\tau$, we see that this protocol achieves the thermodynamic Landauer's bound:

$$\begin{aligned}
\langle W^\theta \rangle &= \sum_{z_\tau, z_{\tau'}} \Pr(Z_\tau = z_\tau, Z_{\tau'} = z_{\tau'}) \left\langle W_{|z_\tau, z_{\tau'}}^\theta \right\rangle \\
&= k_B T \ln 2 (H[Z_{\tau'}] - H[Z_\tau]) .
\end{aligned}$$

One concludes that this is a thermodynamically-efficient method for computing any Markov channel.

Appendix D: Designing Agents

To design a predictive thermodynamic agent, its hidden states must match the states of the ϵ -machine at every time step $X_i = S_i^\theta$. To do this, the agent states and causal states occupy the same space $\mathcal{X} = \mathcal{S}$, and the transitions within the agent M are directly drawn from

causal equivalence relation:

$$M_{xy \rightarrow x'y'} = \frac{1}{|\mathcal{Y}|} \times \begin{cases} \delta_{x', \epsilon(x,y)} & \text{if } \sum_{x'} \theta_{x \rightarrow x'}^{(y)} \neq 0, \\ \delta_{x', x} & \text{otherwise.} \end{cases}$$

The factor $1/|\mathcal{Y}|$ maximizes work production by mapping to uniform outputs.

The second term on the right is the probability of the next agent state given the current input and current hidden state $\Pr(X_{i+1} = x' | Y_i = y, X_i = x)$. The top case $\delta_{x', \epsilon(x,y)}$ gives the probability that the next *causal* state is $S_{i+1}^\theta = x'$ given that the current causal state is $S_i^\theta = x$ and output of the ϵ -machine is $Y_i = y$. This is contingent on the probability of seeing y given causal state x being nonzero. If it is, then the transitions among the agent's hidden states match the transitions of the ϵ -machine's causal states.

In this way, if $y_{0:L}$ is a sequence that could be produced by the ϵ -machine, we have designed the agent to stay synchronized to the causal state of the input $X_i = S_i^\theta$, so that the ratchet is predictive of the process $\Pr(Y_{0:\infty}^\theta)$ and produces maximal work by fully randomizing the outputs:

$$\Pr(Y'_i = y'_i | \cdot) = \frac{1}{|\mathcal{Y}|}.$$

It can be the case that the ϵ -machine cannot produce y from the causal state x . This corresponds to a disallowed transition of our model $\theta_{x \rightarrow x'}^{(y)} = 0$. In these cases, we arbitrarily choose the next state to be the same $\delta_{x, x'}$. There are many possible choices, though—such as resetting to the start state s^* . However, it is physically irrelevant, since these transitions correspond zero estimated probability and, thus, to infinite work dissipation, drowning out all other details of the model. However, this particular choice for when y cannot be generated from the causal state x preserves unifilarity and allows the agent to wait in its current state until it receives an input that it can accept.

Modulo disallowed, infinitely-dissipative transitions, we now have a direct mapping between our estimated input process $\Pr(Y_{0:\infty}^\theta)$ and its ϵ -machine θ to the logical architecture M of a maximum work-producing agent.

As yet, this does not fully specify agent behavior, since it leaves out the estimated input distribution $\Pr(Y_i^\theta = y, X_i^\theta = x)$. This distribution must match the actual distribution $\Pr(Y_i = y, X_i = x)$ for the agent to be locally efficient, not accounting for temporal correlations. Fortunately, since agent states are designed to match the ϵ -machine's causal states, we know that the agent state

distribution matches the causal-state distribution and inputs:

$$\Pr(Y_i^\theta = y_i, X_i^\theta = s_i) = \Pr(Y_i^\theta = y_i, S_i^\theta = s_i),$$

if the ratchet is driven by the estimated input. The joint distribution over causal states and inputs is also determined by the ϵ -machine, since the construction assumes starting in the state $s_0 = s^*$. To start, note that the joint probability trajectory distribution is given by:

$$\begin{aligned} \Pr(Y_{0:i+1}^\theta = y_{0:i+1}, S_{0:i+2}^\theta = s_{0:i+2}) \\ = \delta_{s_0, s^*} \prod_{j=0}^i \theta_{s_j \rightarrow s_{j+1}}^{(y_j)}. \end{aligned}$$

Summing over the variables besides Y_i^θ and S_i^θ , we obtain an expression for the estimated agent distribution in terms of just the ϵ -machine's HMM:

$$\Pr(Y_i^\theta = y_i, X_i^\theta = s_i) = \sum_{y_{0:i}, s_{0:i}, s_{i+1}} \delta_{s_0, s^*} \prod_{j=0}^i \theta_{s_j \rightarrow s_{j+1}}^{(y_j)}.$$

Thus, an agent $\{M, \Pr(X_i^\theta, Y_i^\theta)\}$ designed to be *globally efficient* for the estimated input process $\Pr(Y_{0:L}^\theta)$ can be derived from the estimated input process through its ϵ -machine $\theta_{s \rightarrow s'}^{(y)}$.

Appendix E: Work Production of Optimal Transducers

The work production of an arbitrary transducer M driven by an input $y_{0:L}$ can be difficult to calculate, as shown in Eq. (10). However, when the transducer is designed to harness an input process with ϵ -machine T , such that:

$$M_{xy \rightarrow x'y'} = \frac{1}{|\mathcal{Y}|} \times \begin{cases} \delta_{x', \epsilon(x,y)} & \text{if } \sum_{x'} \theta_{x \rightarrow x'}^{(y)} \neq 0, \\ \delta_{x', x} & \text{else,} \end{cases}$$

the work production simplifies. To see this, we express Eq. (10) in terms of the estimated distribution $\Pr(Y_{0:L}^\theta)$, ratchet M , and input $y_{0:L}$, assuming that the word $y_{0:L}$ can be produced from every initial hidden state with nonzero probability $\Pr(Y_{0:L}^\theta = y_{0:L} | S_0 = s_0) \Pr(X_0 = s_0) \neq 0$, which guarantees that $\sum_{x_{i+1}} \theta_{x_i \rightarrow x_{i+1}}^{(y_i)} \neq 0$. If this constraint is not satisfied, the agent will dissipate infinite work, as it implies $\Pr(X_i^\theta = x_i, Y_i^\theta = x_i) = 0$ for some i . Thus, we use the expression $M_{xy \rightarrow x'y'} = \delta_{x', \epsilon(x,y)} / |\mathcal{Y}|$ in the work production:

$$\begin{aligned}
& \langle W_{|y_{0:L}}^\theta \rangle \\
&= k_B T \sum_{x_{0:L+1}, y'_{0:L}} \Pr(X_0 = x_0) \prod_{j=0}^{L-1} M_{x_j, y_j \rightarrow x_{j+1}, y'_j} \ln \prod_{i=0}^{L-1} \frac{\Pr(X_i^\theta = x_i, Y_i^\theta = y_i)}{\Pr(X_{i+1}^\theta = x_{i+1}, Y_i^{\prime\theta} = y'_i)} \\
&= k_B T \sum_{x_{0:L+1}, y'_{0:L}} \Pr(X_0 = x_0) \prod_{j=0}^{L-1} \frac{\delta_{x_{j+1}, \epsilon(x_j, y_j)}}{|\mathcal{Y}|} \ln \prod_{i=0}^{L-1} \frac{\Pr(X_i^\theta = x_i, Y_i^\theta = y_i)}{\Pr(X_{i+1}^\theta = x_{i+1})/|\mathcal{Y}|} \\
&= k_B T \ln |\mathcal{Y}| + k_B T \sum_{x_{0:L+1}} \Pr(X_0 = x_0) \prod_{j=0}^{L-1} \delta_{x_{j+1}, \epsilon(x_j, y_j)} \left(\ln \prod_{i=0}^{L-1} \Pr(Y_i^\theta = y_i | X_i^\theta = x_i) + \ln \prod_{i=0}^{L-1} \frac{\Pr(X_i^\theta = x_i)}{\Pr(X_{i+1}^\theta = x_{i+1})} \right).
\end{aligned}$$

Note that $\prod_{j=0}^{L-1} \delta_{x_{j+1}, \epsilon(x_j, y_j)}$ vanishes unless each element of the hidden state trajectory x_i corresponds to the resulting state of the ϵ -machine when the initial state x_0 is driven by the first i inputs $y_{0:i}$, which is $\epsilon(x_0, y_{0:i})$. The fact that the agent is driven into a unique state is guaranteed by the ϵ -machine's unifilarity. Thus, we rewrite:

$$\prod_{j=0}^{L-1} \delta_{x_{j+1}, \epsilon(x_j, y_j)} = \prod_{j=1}^L \delta_{x_j, \epsilon(x_0, y_{0:j})}.$$

This engenders a simplification of the work production:

$$\begin{aligned}
\langle W_{|y_{0:L}}^\theta \rangle &= k_B T \ln |\mathcal{Y}| + k_B T \sum_{x_{0:L+1}} \Pr(X_0 = x_0) \prod_{j=1}^L \delta_{x_j, \epsilon(x_0, y_{0:j})} \ln \prod_{i=0}^{L-1} \frac{\Pr(Y_i^\theta = y_i | X_i^\theta = x_i) \Pr(X_i^\theta = x_i)}{\Pr(X_{i+1}^\theta = x_{i+1})} \\
&= k_B T \ln |\mathcal{Y}| + k_B T \sum_{x_0} \Pr(X_0 = x_0) \ln \prod_{i=0}^{L-1} \Pr(Y_i^\theta = y_i | X_i^\theta = \epsilon(x_0, y_{0:i})) \\
&\quad + k_B T \sum_{x_0} \Pr(X_0 = x_0) \ln \frac{\Pr(X_0^\theta = x_0)}{\Pr(X_L^\theta = \epsilon(x_0, y_{0:L}))}, \tag{E1}
\end{aligned}$$

where $y_{0:0}$ denotes the null input, which leaves the state fixed under the ϵ -map $\epsilon(x_0, y_{0:0}) = x_0$.

This brings us to an easily calculable work production, especially when the system is initialized in the start state s^* . Recognizing that if we initiate the ϵ -machine in its start state s^* , such that $\Pr(X_0 = x_0) = \delta_{x_0, s^*}$, then $X_0 = S_0$ is predictive. By extension, every following agent state is predictive and equivalent to the causal state $X_i = S_i$ yielding:

$$\begin{aligned}
\Pr(Y_i = y_i | X_i = \epsilon(s^*, y_{0:i})) &= \Pr(Y_i = y_i | S_i = \epsilon(s^*, y_{0:i})) \\
&= \Pr(Y_i = y_i | Y_{0:i} = y_{0:i}).
\end{aligned}$$

Thus, the work production simplifies a sum of terms that

includes the log-likelihood of the finite input:

$$\begin{aligned}
& \langle W_{|y_{0:L}}^\theta \rangle \\
&= k_B T \sum_{x_0} \delta_{x_0, s^*} \ln \prod_{i=0}^{L-1} \Pr(Y_i^\theta = y_i | Y_{0:i}^\theta = y_{0:i}) \\
&\quad + k_B T L \ln |\mathcal{Y}| + k_B T \sum_{x_0} \delta_{x_0, s^*} \ln \frac{\delta_{x_0, s^*}}{\Pr(X_L^\theta = \epsilon(x_0, y_{0:L}))} \\
&= k_B T (\ln \Pr(Y_{0:L}^\theta = y_{0:L}) + L \ln |\mathcal{Y}| \\
&\quad - \ln \Pr(X_L^\theta = \epsilon(s^*, y_{0:L}))) .
\end{aligned}$$

The first $\log(\cdot)$ in the last line is the log-likelihood of the model generating $y_{0:L}$ —a common performance measure for machine learning algorithms. If an input has zero probability this leads to $-\infty$ work production, and all other features are drowned out by the log-likelihood term. Thus, the additional terms that come into play when the input probability vanishes become physically irrelevant: the agent is characterized by the ϵ -machine. From a ma-

chine learning perspective, the model is also characterized by the ϵ -machine $\theta_{s \rightarrow s'}^{(y)}$ for the process $\Pr(Y_{0:\infty}^\theta)$. The additional term $k_B T L \ln |\mathcal{Y}|$ is the work production that comes from exhausting fully randomized outputs and does not change depending on the underlying model.

The final term $-k_B T \ln \Pr(X_L^\theta = \epsilon(s^*, y_{0:L}))$ does directly depend on the model. $\Pr(X_L^\theta = x)$ is the distribution over agent states \mathcal{X} at time $L\tau$ if the agent is driven by the estimated input distribution $Y_{0:L}^\theta$. This component of the work production is larger, on average, for agents with high state uncertainty, since this leads, on-average, to smaller values of $\Pr(X_L^\theta)$. This contribution to the work production comes from the state space expanding from the start state s^* to the larger (recurrent) subset of agent states, and so it provides additional work. This indicates that we are neglecting the cost of resetting to the start state while harnessing the energetic benefit of starting in it.

If the machine is designed to efficiently harness inputs again after it operates on one string, it must be reset to the start state s^* . This can be implemented with an efficient channel that anticipates the input distribution $\Pr(X_L^\theta = x)$, outputs the distribution $\Pr(X_{L+1}^\theta = x) = \delta_{x,s^*}$, and so costs:

$$W_{|y_{0:L}}^{\theta, \text{reset}} = k_B T \ln \Pr(X_L^\theta = \epsilon(s^*, y_{0:L})) .$$

Thus, when we add the cost of resetting the agent to the start state at X_{L+1} , the work production is dominated by the log-likelihood:

$$\langle W_{|y_{0:L}}^\theta \rangle = k_B T (\ln \Pr(Y_{0:L}^\theta = y_{0:L}) + L \ln |\mathcal{Y}|) . \quad (\text{E2})$$

Appendix F: Training Simple Agents

We now outline a case study of thermodynamic learning that is experimentally implementable using a controllable two-level system. We first introduce a straightforward method to implement the simplest possible efficient agent. Second, we show that this physical process achieves the general maximum-likelihood result arrived at in the main development. Last, we find the agent selected by thermodynamic learning along with its corresponding model. As expected, we find that this maximum-work producing agent learns its environment's predictive features.

1. Efficient Computational Trajectories

The simplest possible information ratchets have only a single internal state A and receive binary data y_j from a series of two-level systems $\mathcal{Y}_j = \{\uparrow, \downarrow\}$. These agents'



FIG. 9. Memoryless model of binary data consisting of a single state A and the probability of outputting a \uparrow and a \downarrow , denoted $\theta_{A \rightarrow A}^{(\uparrow)}$ and $\theta_{A \rightarrow A}^{(\downarrow)}$, respectively.

internal models correspond to memoryless ϵ -machine s , as shown in Fig. 9. The model's parameters are the probabilities of emitting \uparrow and \downarrow , denoted $\theta_{A \rightarrow A}^{(\uparrow)}$ and $\theta_{A \rightarrow A}^{(\downarrow)}$, respectively.

Our first step is to design an efficient computation that maps an input distribution $\Pr(Z_{j\tau})$ to an output distribution $\Pr(Z_{j\tau+\tau'})$ over the j th interaction interval $[j\tau, j\tau + \tau']$. The agent corresponds to the Hamiltonian evolution $\mathcal{H}_Z(t) = \mathcal{H}_{\mathcal{X} \times \mathcal{Y}_j}(t)$ over the joint space of the agent memory and j th input symbol. The resulting energy landscape $E(z, t)$ is entirely specified by the energy of the two input states $E(A \times \uparrow, t)$ and $E(A \times \downarrow, t)$.

Appropriately designing this energy landscape allows us to implement the efficient computation shown in Fig. 10. The thermodynamic evolution there instantaneously quenches the energy landscape into equilibrium with the estimated distribution at the beginning of the interaction interval $\Pr(Z_{j\tau}^\theta)$, then quasistatically evolves the system in equilibrium to the estimated final distribution $\Pr(Z_{j\tau+\tau'}^\theta)$, and, finally, quenches back to the default energy landscape. In Fig. 10, the system undergoes a cycle, starting and ending with the same flat energy landscape, such that $\Delta E_Z = 0$. This cycle evolves the distribution over the joint states $A \times \uparrow$ and $A \times \downarrow$ from $\Pr(Z_{j\tau}^\theta = \{A \times \uparrow, A \times \downarrow\}) = \{0.8, 0.2\}$ to $\Pr(Z_{j\tau+\tau'}^\theta = \{A \times \uparrow, A \times \downarrow\}) = \{0.4, 0.6\}$. Note that this strategy can be used to evolve between any initial and final distributions.

We control the transformation over time interval $t \in (j\tau, j\tau + \tau')$ such that the time scale of equilibration in the system of interest is much shorter than the interval length τ' . This slow-moving quasistatic control means that the states are in equilibrium with the energy landscape over the interval. In this case, the state distribution becomes the Boltzmann distribution:

$$\Pr(Z_t = z) = e^{(F^{\text{EQ}}(t) - E(z, t))/k_B T} .$$

To minimize dissipation for the estimated distribution, the state distribution must be the estimated distribution $\Pr(Z_t = z) = \Pr(Z_t^\theta = z)$. And so, we set the two-level-system energies to be in equilibrium with the estimates:

$$E(z, t) = -k_B T \ln \Pr(Z_t^\theta = z) .$$

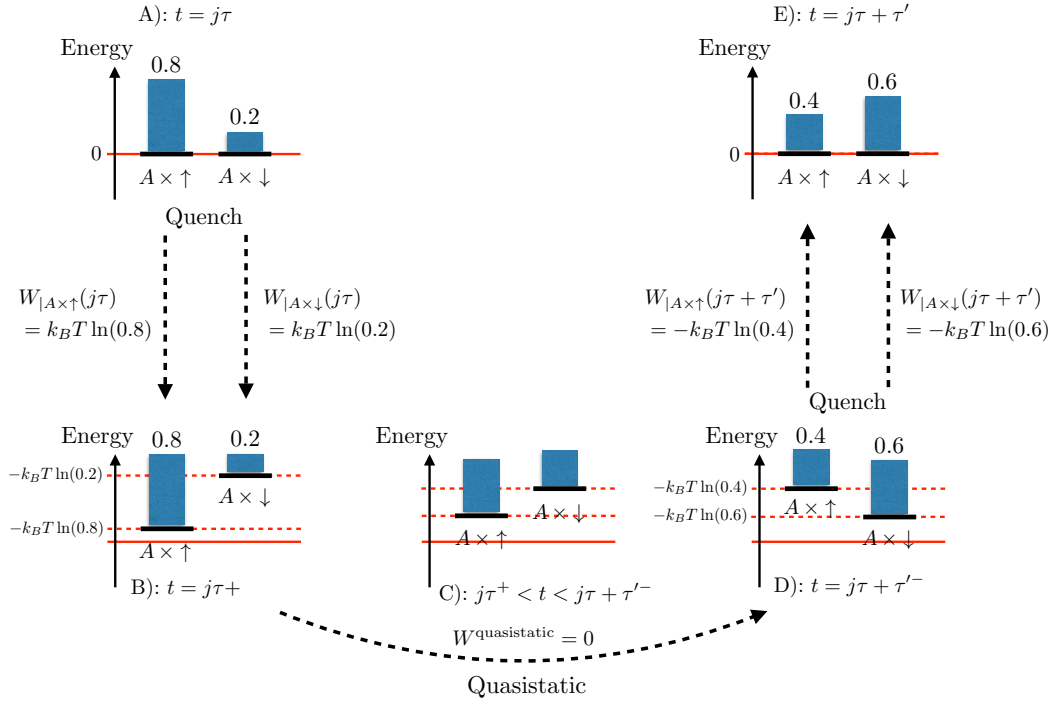


FIG. 10. Joint two-level system $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}_j = \{A \times \uparrow, A \times \downarrow\}$ undergoing perfectly-efficient computation when it receives its estimated input through a series of operations. The computation occurs over the time interval $t \in (j\tau, j\tau + \tau')$. At panel A) $t = j\tau$ and the system has a default flat energy landscape energy $E(z, j\tau) = E(x \times y, j\tau) = 0$. However, it is out of equilibrium, since it is in the distribution $\Pr(Z_{j\tau}^\theta = \{A \times \uparrow, A \times \downarrow\}) = \{0.8, 0.2\}$. The first operation is a quench, which instantaneously sets the energies be in equilibrium with the initial distribution, as shown in panel B). The associated energy change is work. Then, a quasistatic operation slowly evolves the system in equilibrium, through panel C), to the final desired distribution $\Pr(Z_{j\tau+\tau'}^\theta = \{A \times \uparrow, A \times \downarrow\}) = \{0.4, 0.6\}$, shown in panel D). This requires no work. Then, the final operation is another quench, in which the energies are reset to the default energy landscape $E(z, j\tau + \tau') = 0$, leaving the system as shown in panel E). Again, the change in energy corresponds to work invested through control. The total work production for a particular computational mapping $A \times y \rightarrow A \times y'$ is given by the work from the initial quench $W_{|A \times y}(j\tau)$ plus the work from the final quench $W_{|A \times y'}(j\tau + \tau')$.

The resulting process produces zero work:

$$W^{\text{quasistatic}} = - \int_{z=j\tau}^{j\tau+\tau'} dt \sum_z \Pr(Z_t^\theta = z) \partial_t E(z, t) = 0$$

and maps $\Pr(Z_{j\tau}^\theta)$ to $\Pr(Z_{j\tau+\tau'}^\theta)$ without dissipation. With the quasistatic transformation producing zero work, the total work produced from the initial joint state $x \times y$ is exactly opposite the change in energy during the initial quench:

$$E(x \times y, j\tau) - E(x \times y, j\tau^+) = k_B T \ln \Pr(Z_{j\tau}^\theta = x \times y)$$

minus the change in energy of the final joint state $x' \times y'$ during the final quench:

$$E(x' \times y', j\tau + \tau'^-) - E(x' \times y', j\tau + \tau') = -k_B T \ln \Pr(Z_{j\tau+\tau'}^\theta = x' \times y').$$

The two-level system's state is fixed during the instantaneous energy changes. Thus, if the joint state follows the computational mapping $x \times y \rightarrow x' \times y'$ the work production is, as expected, directly connected to the estimated distributions:

$$\langle W_{|x \times y, x' \times y'} \rangle = k_B T \ln \frac{\Pr(Z_{j\tau}^\theta = x \times y)}{\Pr(Z_{j\tau+\tau'}^\theta = x' \times y')}. \quad (\text{F1})$$

Recall from Sec. IV A that the ratchet system variable $Z_{j\tau}^\theta = X_j^\theta \times Y_j^\theta$ splits into the random variable X_j^θ for the j th agent memory state and the j th input Y_j^θ . Similarly, $Z_{j\tau+\tau'}^\theta = X_{j+1}^\theta \times Y_j^\theta$ splits into the $(j+1)$ th agent memory state X_{j+1}^θ and j th output Y_j^θ . This work production achieves the efficient limit for a model θ ($\langle W_{|x \times y, x' \times y'} \rangle = \langle W_{|x \times y, x' \times y'}^\theta \rangle$) described in Eq. (8). Appendix C generalized the thermodynamic operation above to *any* computation $M_{z_\tau \rightarrow z_{\tau'}}$. While it requires an ancillary copy of the system \mathcal{Z} to execute the conditional dependencies in the computation, it is concep-

tually identical in that it uses a sequence of quenching, evolving quasistatically, and then quenching again. This appendix extends the strategies outlined in Refs. [42, 53] to computational-mapping work calculations.

2. Efficient Information Ratchets

With the method for efficiently mapping inputs to outputs in hand, we can design a series of such computations to implement a simple information ratchet that produces work from a series $y_{0:L}$. As prescribed in Eq. (11) of Sec. V, to produce the most work from estimated model θ , the agent's logical architecture should randomly map every state to all others:

$$\begin{aligned} M_{xy \rightarrow x'y'} &= \frac{1}{|\mathcal{Y}_j|} \\ &= \frac{1}{2}, \end{aligned}$$

since there is only one causal state A . In conjunction with Eq. (12), we find that the estimated joint distribution of the agent and interaction symbol at the start of the interaction is equivalent to the parameters of the model:

$$\begin{aligned} \Pr(Z_{j\tau}^\theta = x \times y) &= \Pr(X_j^\theta = x, Y_j^\theta = y) \\ &= \Pr(Y_j^\theta = y | X_j^\theta = A) \Pr(X_j^\theta = A) \\ &= \theta_{A \rightarrow A}^{(y)}, \end{aligned}$$

where we again used the fact that A is the only causal state. In turn, the estimated distribution after the interaction is:

$$\begin{aligned} \Pr(Z_{j\tau+\tau'}^\theta = x' \times y') &= \sum_{xy} \Pr(X_j^\theta = x, Y_j^\theta = y) M_{xy \rightarrow x'y'} \\ &= \frac{1}{2}. \end{aligned}$$

Thus, assuming the agent has model θ built-in, then Eq. (F1) determines that the work production for mapping $A \times y$ to output $A \times y'$ for a particular symbol y is:

$$\langle W_{|A \times y, A \times y'} \rangle = k_B T \left(\ln 2 + \ln \theta_{A \rightarrow A}^{(y)} \right).$$

Since A is the only memory state and work does not depend on the output symbol y' , the average work produced from an input y is:

$$\langle W_{|y} \rangle = \langle W_{|A \times y, A \times y'} \rangle. \quad (\text{F2})$$

With the work production expressed for a single input y_j , we can now consider how much work our designed agent harvests from the training data $y_{0:L}$. Summing the work

production of each input yields a simple expression in terms of the model θ :

$$\begin{aligned} \langle W_{|y_{0:L}} \rangle &= \sum_{j=0}^{L-1} \langle W_{|y_j} \rangle \\ &= \sum_{j=0}^{L-1} k_B T \left(\ln 2 + \ln \theta_{A \rightarrow A}^{(y_j)} \right) \\ &= k_B T \left(L \ln 2 + \ln \prod_{j=0}^{L-1} \theta_{A \rightarrow A}^{(y_j)} \right). \end{aligned}$$

Due to the single causal state, the product within the logarithm simplifies to the probability of the word given the model $\prod_{j=0}^{L-1} \theta_{A \rightarrow A}^{(y_j)} = \Pr(Y_{0:L}^\theta = y_{0:L})$. So, the resulting work production depends on the familiar log-likelihood:

$$\begin{aligned} \langle W_{|y_{0:L}} \rangle &= k_B T (L \ln 2 + \ell(\theta|y_{0:L})) \\ &= \langle W_{|y_{0:L}}^\theta \rangle, \end{aligned}$$

again, achieving efficient work production, as expected.

3. Maximizing Work for Memoryless Models

Leveraging the explicit construction for efficient information ratchets, we can search for the agent that maximizes work from the input string $y_{0:L}$. To infer a model through work maximization, we label the frequency of \uparrow states in this sequence with $f(\uparrow)$ and the frequency of \downarrow with $f(\downarrow)$. The corresponding log-likelihood of the model is:

$$\begin{aligned} \ell(\theta|y_{0:L}) &= \ln \left(\theta_{A \rightarrow A}^{(\uparrow)} \right)^{L f(\uparrow)} \left(\theta_{A \rightarrow A}^{(\downarrow)} \right)^{L f(\downarrow)} \\ &= L f(\uparrow) \ln \left(\theta_{A \rightarrow A}^{(\uparrow)} \right) + L f(\downarrow) \ln \left(\theta_{A \rightarrow A}^{(\downarrow)} \right). \end{aligned}$$

Thus, for the corresponding agent, the work production is:

$$\begin{aligned} \langle W_{|y_{0:L}}^\theta \rangle &= k_B T \ell(\theta|y_{0:L}) + k_B T L \ln 2 \\ &= k_B T L \left(\ln 2 + f(\uparrow) \ln \theta_{A \rightarrow A}^{(\uparrow)} + f(\downarrow) \ln \theta_{A \rightarrow A}^{(\downarrow)} \right). \end{aligned}$$

Selecting from all possible memoryless agents, the model parameters θ maximizing work production are given by the frequency of symbols in the input: $f(\uparrow) = \theta_{A \rightarrow A}^{(\uparrow)}$ and $f(\downarrow) = \theta_{A \rightarrow A}^{(\downarrow)}$. The resulting work production is:

$$\langle W_{|y_{0:L}}^\theta \rangle = k_B T L (\ln 2 - H[f(\uparrow)]),$$

where $H[f(\uparrow)]$ is the Shannon entropy of binary variable Y with $\Pr(Y = \uparrow) = f(\uparrow)$ measured in nats.

This simple example of learning statistical bias serves to

explicitly lay out the stages of thermodynamic machine learning. The class of models is too simple, though, to illustrate the full power of the new learning method. That said, it does confirm that thermodynamic work maximization leads to useful models of data in the simplest case. As one would expect, the simple agent found by thermodynamic machine learning discovers the frequency of zeros in the input and, thus, it learns about its environment. The

corresponding work production is the same as energetic gain of randomizing L bits distributed according to the frequency $f(\uparrow)$.

However, this neglects the substantial thermodynamic benefits possible with temporally-correlated environments [38]. To illustrate how to extract this additional energy, a sequel designs and analyzes memoryful agents.

-
- [1] (Baron) G. Cuvier. *Essay on the Theory of the Earth*. Kirk and Mercein, New York, 1818. 1
 - [2] H. Bergson. *Creative Evolution*. Henry Holt and Company, New York, New York, 1907. 1
 - [3] D. W. Thompson. *On Growth and Form*. Cambridge University Press, Cambridge, 1917. 1
 - [4] N. Wiener. *Cybernetics: Or Control and Communication in the Animal and the Machine*. MIT, Cambridge, 1948. 1, 13, 14
 - [5] N. Wiener. *The Human Use Of Human Beings: Cybernetics And Society*. Da Capo Press, Cambridge, 1988. 1
 - [6] D. C. Dennett. *Consciousness Explained*. Little, Brown and Co., New York, New York, 1991. 1
 - [7] S. J. Gould and R. Lewontin. The spandrels of san marco and the panglossian paradigm: A critique of the adaptationist programme. *Proc. Roy. Soc. Lond. B*, 205(1161):581–598, 1979. 1
 - [8] D. C. Dennett. *Darwin’s Dangerous Idea: Evolution and the Meanings of Life*. Simon and Schuster, New York, New York, 1995.
 - [9] J. Maynard-Smith and E. Szathmary. *The Major Transitions in Evolution*. Oxford University Press, Oxford, reprint edition, 1998.
 - [10] G. P. Wagner. *Homology, Genes, and Evolutionary Innovation*. Princeton University Press, Princeton, New Jersey, 2014. 1
 - [11] W. Thomson. Kinetic theory of the dissipation of energy. *Nature*, pages 441–444, 9 April 1874. 1
 - [12] J. C. Maxwell. *Theory of Heat*. Longmans, Green and Co., London, United Kingdom, ninth edition, 1888. 1
 - [13] T. Sagawa. Thermodynamics of information processing in small systems. *Prog. Theo. Phys.*, 127(1):1–56, 2012. 1
 - [14] J. M. R. Parrondo, J. M. Horowitz, and T. Sagawa. Physics of information. *Nature Physics*, 11(2):131, 2015. 1, 6, 13
 - [15] S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014. 2, 4, 15
 - [16] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer, New York, New York, second edition, 1818.
 - [17] P. Mehta, M. Bukov, C.-H. Wang, A. G. R. Day, C. Richardson, C. K. Fisher, and D. J. Schwab. A high-bias, low-variance introduction to machine learning for physicists. *Physics Reports*, 810(1-124), 2019. 2, 4
 - [18] H. W. Lin, M. Tegmark, and D. Rolnick. Why does deep and cheap learning work so well? *J. Stat. Phys.*, 168(6):1223–1247, 2017. 2, 15
 - [19] R. Landauer. Irreversibility and heat generation in the computing process. *IBM J. Res. Develop.*, 5(3):183–191, 1961. 2, 5, 6, 13, 17
 - [20] C. H. Bennett. Demons, engines and the Second Law. *Sci. Am.*, 257(5):108–116, 1987.
 - [21] L. Szilard. On the decrease of entropy in a thermodynamic system by the intervention of intelligent beings. *Z. Phys.*, 53:840–856, 1929. 2
 - [22] T. L. H. Watkins, A. Rau, and M. Biehl. The statistical mechanics of learning a rule. *Rev. Mod. Phys.*, 65:499 – 556, 1993. 2
 - [23] A. Engel and C. Van den Broeck. *Statistical Mechanics of Learning*. Cambridge University Press, 2001.
 - [24] A. Bell. Learning out of equilibrium. In J. P. Crutchfield and J. Machta, editors, *Santa Fe Institute Workshop on Randomness, Structure, and Causality, 9-13 January 2011*, volume 21, 2011.
 - [25] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *arXiv:1503.03585*, 2015.
 - [26] S. Goldt and U. Seifert. Stochastic thermodynamics of learning. *Phys. Rev. Lett.*, 118(010601), 2017. 3
 - [27] A. A. Alemi and I. Fischer. TherML: Thermodynamics of machine learning. *arXiv:1807.04162*, 2018.
 - [28] Y. Bahri, J. Kadmon, J. Pennington, S. S. Schoenholz, J. Sohl-Dickstein, and S. Ganguli. Statistical mechanics of deep learning. *Ann. Rev. Cond. Matter Physics*, 11:501–528, 2020. 2
 - [29] A. B. Boyd, D. Mandal, and J. P. Crutchfield. Leveraging environmental correlations: The thermodynamics of requisite variety. *J. Stat. Phys.*, 167(6):1555–1585, 2016. 3, 7, 12, 14
 - [30] D. Mandal and C. Jarzynski. Work and information processing in a solvable model of Maxwell’s demon. *Proc. Natl. Acad. Sci. USA*, 109(29):11641–11645, 2012. 3, 6, 18
 - [31] A. B. Boyd, D. Mandal, and J. P. Crutchfield. Identifying functional thermodynamics in autonomous Maxwellian ratchets. *New J. Physics*, 18:023049, 2016. 3, 6, 7, 18
 - [32] J. M. Gold and J. L. England. Self-organized novelty detection in driven spin glasses. *arXiv:1911.07216*, 2019. 3, 14

- [33] W. Zhong, J. M. Gold, S. Marzen, J. L. England, and N. Y. Halpern. Learning about learning by many-body systems. *arXiv:2004.03604 [cond-mat.stat-mech]*, 2020. 3, 4, 14
- [34] D. Jimenez Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014. 4, 15, 16
- [35] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Stat. Soc. Series B*, 39(1):1–22, 1977. 4
- [36] J. P. Crutchfield. Between order and chaos. *Nature Physics*, 8(January):17–24, 2012. 4, 13
- [37] S. Deffner and C. Jarzynski. Information processing and the second law of thermodynamics: An inclusive, Hamiltonian approach. *Phys. Rev. X*, 3:041003, 2013. 5, 16
- [38] A. B. Boyd, D. Mandal, and J. P. Crutchfield. Correlation-powered information engines and the thermodynamics of self-correction. *Phys. Rev. E*, 95(1):012152, 2017. 7, 27
- [39] A. B. Boyd, D. Mandal, P. M. Riechers, and J. P. Crutchfield. Transient dissipation and structural costs of physical information transduction. *Phys. Rev. Lett.*, 118:220602, 2017.
- [40] N. Merhav. Sequence complexity and work extraction. *J. Stat. Mech.*, page P06037, 2015.
- [41] N. Merhav. Relations between work and entropy production for general information-driven, finite-state engines. *J. Stat. Mech.: Th. Expt.*, 2017:1–20, 2017.
- [42] A. J. P. Garner, J. Thompson, V. Vedral, and M. Gu. Thermodynamics of complexity and pattern manipulation. *Phys. Rev. E*, 95(042140), 2017. 7, 14, 19, 26
- [43] S. Deffner and E. Lutz. Information free energy for nonequilibrium states. *arXiv:1201.3888*, 2012. 7, 16
- [44] M. Esposito and C. van den Broeck. Second law and Landauer principle far from equilibrium. *Europhys. Lett*, 95:40004, 2011. 8, 16
- [45] G. E. Crooks. Entropy production fluctuation theorem and the nonequilibrium work relation for free energy differences. *Phys. Rev. E*, 60(3), 1999. 8, 18
- [46] C. Jarzynski. Hamiltonian derivation of a detailed fluctuation theorem. *J. Stat. Phys.*, 98(77-102), 2000. 8, 18
- [47] T. Speck and U. Seifert. Distribution of work in isothermal nonequilibrium processes. *Phys. Rev. E*, 70(066112), 2004. 8
- [48] K. J. Ray, G. W. Wimsatt, A. B. Boyd, and J. P. Crutchfield. Non-markovian momentum computing: Universal and efficient. 2020. arxiv:2010.01152. 8, 19, 21
- [49] A. Kolchinsky and D. H. Wolpert. Dependence of dissipation on the initial distribution over states. *J. Stat. Mech.: Th. Expt.*, page 083202, 2017. 9, 19
- [50] P. M. Riechers and M. Gu. Initial-state dependence of thermodynamic dissipation for any quantum process. *arXiv:2002.11425*, 2020. 9
- [51] J. G. Brookshear. *Theory of computation: Formal languages, automata, and complexity*. Benjamin/Cummings, Redwood City, California, 1989. 10, 18
- [52] N. Barnett and J. P. Crutchfield. Computational mechanics of input-output processes: Structured transformations and the ϵ -transducer. *J. Stat. Phys.*, 161(2):404–451, 2015. 10, 18
- [53] A. B. Boyd, D. Mandal, and J. P. Crutchfield. Thermodynamics of modularity: Structural costs beyond the Landauer bound. *Phys. Rev. X*, 8(031036), 2018. 11, 14, 19, 26
- [54] G. R. Kirchhoff. *Ann. Phys.*, 75:1891, 1848. 13
- [55] J. W. Gibbs. *The Scientific Papers of J. Willard Gibbs*. Longmans, Green, New York, New York, 1906. 13
- [56] J. C. Maxwell. *A Treatise on Electricity and Magnetism, vol. I and II*. Dover Publications, Inc., New York, New York, third edition, 1954. 13
- [57] L. Onsager. Reciprocal relations in irreversible processes, I. *Phys. Rev.*, 37(4):405–426, 1931. 13
- [58] I. Prigogine. Modération et transformations irréversibles des systèmes ouverts. *Bulletin de la Classe des Sciences, Académie Royale de Belgique*, 31:600–606, 1945. 13
- [59] I. Prigogine and P. Glansdorff. *Thermodynamic Theory of Structure, Stability and Fluctuations*. Wiley-Interscience, London, 1971. 13
- [60] G. Falasco, R. Rao, and M. Esposito. Information thermodynamics of Turing patterns. *Phys. Rev. Lett.*, 121:108301, 2018. 13
- [61] A. M. Turing. The chemical basis of morphogenesis. *Trans. Roy. Soc., Series B*, 237:5, 1952. 13
- [62] R. Hoyle. *Pattern Formation: An Introduction to Methods*. Cambridge University Press, New York, 2006.
- [63] M. Cross and H. Greenside. *Pattern Formation and Dynamics in Nonequilibrium Systems*. Cambridge University Press, Cambridge, United Kingdom, 2009. 13
- [64] W. Heisenberg. Nonlinear problems in physics. *Physics Today*, 20:23–33, 1967. 13
- [65] D. Ruelle and F. Takens. On the nature of turbulence. *Comm. Math. Phys.*, 20:167–192, 1971. 13
- [66] A. Brandstater, J. Swift, Harry L. Swinney, A. Wolf, J. D. Farmer, E. Jen, and J. P. Crutchfield. Low-dimensional chaos in a hydrodynamic system. *Phys. Rev. Lett.*, 51:1442, 1983. 13
- [67] C. E. Shannon. A mathematical theory of communication. *Bell Sys. Tech. J.*, 27:379–423, 623–656, 1948. 13
- [68] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, New York, second edition, 2006. 13, 15, 19
- [69] R. G. James, C. J. Ellison, and J. P. Crutchfield. Anatomy of a bit: Information in a time series observation. *CHAOS*, 21(3):037109, 2011. 13
- [70] C. E. Shannon. The bandwagon. *IEEE Trans. Info. Th.*, 2(3):3, 1956. 13
- [71] A. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proc. Lond. Math. Soc.*, 42, 43:230–265, 544–546, 1937. 13
- [72] W. Ross Ashby. *An Introduction to Cybernetics*. John Wiley and Sons, New York, second edition, 1960. 13
- [73] E. T. Jaynes. Information theory and statistical mechanics. *Phys. Rev.*, 106(4):620–630, 1957. 13
- [74] E. T. Jaynes. The minimum entropy production principle. *Annu. Rev. Phys. Chem.*, 31:579–601, 1980. 13

- [75] A. N. Kolmogorov. A new metric invariant of transient dynamical systems and automorphisms in Lebesgue spaces. *Dokl. Akad. Nauk. SSSR*, 119:861, 1958. (Russian) Math. Rev. vol. 21, no. 2035a. [13](#)
- [76] N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. Shaw. Geometry from a time series. *Phys. Rev. Lett.*, 45:712, 1980.
- [77] F. Takens. Detecting strange attractors in fluid turbulence. In D. A. Rand and L. S. Young, editors, *Symposium on Dynamical Systems and Turbulence*, volume 898, page 366, Berlin, 1981. Springer-Verlag.
- [78] J. P. Crutchfield and B. S. McNamara. Equations of motion from a data series. *Complex Systems*, 1:417 – 452, 1987. [13](#)
- [79] L. Brillouin. *Science and Information Theory*. Academic Press, New York, second edition, 1962. [13](#)
- [80] C. H. Bennett. Thermodynamics of computation—a review. *Intl. J. Theo. Phys.*, 21:905, 1982. [13](#)
- [81] T. Sagawa and M. Ueda. Information thermodynamics: Maxwell’s demon in nonequilibrium dynamics. *Nonequilibrium Stat. Phys. Small Syst. Fluct. Relations Beyond*, pages 181–211, nov 2013. [13](#)
- [82] J. P. Crutchfield and O. Gernerup. Objects that make objects: The population dynamics of structural complexity. *J. Roy. Soc. Interface*, 3:345–349, 2006. [14](#)
- [83] J. England. Statistical physics of self-replication. *J. Chem. Phys.*, 139(121923), 2013. [14](#)
- [84] V. Serreli, C.-F. Lee, E. R. Kay, and D. A. Leigh. A molecular information ratchet. *Nature*, 445(7127):523–537, 2007. [14](#)
- [85] J. Thompson, A. J. P. Garner, V. Vedral, and M. Gu. Using quantum theory to simplify input–output processes. *njp Quant. Info.*, 3(6), 2017. [14](#)
- [86] S. Loomis and J. P. Crutchfield. Thermal efficiency of quantum memory compression. *Phys. Rev. Lett.*, 125:020601, 2020.
- [87] M. P. Woods, R. Silva, G. Putz, S. Stupar, and R. Renner. Quantum clocks are more accurate than classical ones. *arXiv preprint arXiv:1806.00491*, 2018. [14](#)
- [88] P. Dayan, G. E. Hinton, R. M. Neal, and R. S. Zemel. The Helmholtz machine. *Neural computation*, 7(5):889–904, 1995. [15](#)
- [89] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521:436–444, May 2015. [16](#)
- [90] R. Landauer. Information is physical. *Physics Today*, pages 23–29, May 1991. [16](#)
- [91] J. P. Crutchfield and D. P. Feldman. Regularities unseen, randomness observed: Levels of entropy convergence. *CHAOS*, 13(1):25–54, 2003. [17](#)
- [92] C. C. Strelhoff and J. P. Crutchfield. Bayesian structural inference for hidden processes. *Phys. Rev. E*, 89:042119, 2014. [18](#)
- [93] T. Wu and M. Tegmark. Toward an artificial intelligence physicist for unsupervised learning. *Phys. Rev. E*, 100(3):033311, 2019. [18](#)
- [94] J. P. Crutchfield. The calculi of emergence: Computation, dynamics, and induction. *Physica D*, 75:11–54, 1994. [18](#)
- [95] C. Jarzynski. Equalities and inequalities: Irreversibility and the second law of thermodynamics at the nanoscale. *Annu. Rev. Cond. Matt. Phys.*, 2(1), 2011. [19](#)
- [96] J. A. Owen, A. Kolchinsky, and D. H. Wolpert. Number of hidden states needed to physically implement a given conditional distribution. *New J. Phys.*, 21(013022), 2019. [19](#)