# UC Davis

## UC Davis Previously Published Works


### Title

Thermodynamic Machine Learning through Maximum Work Production


### Permalink

https://escholarship.org/uc/item/36f3g136



### Authors

Boyd, AB
Crutchfield, JP
Gu, M



### Publication Date

2020-06-27

Peer reviewed



eScholarship.org Powered by the California Digital Library
University of California

# Thermodynamic Machine Learning through Maximum Work Production

Alexander B. Boyd*
*Complexity Institute, Nanyang Technological University, Singapore and*
*School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore*

James P. Crutchfield†
*Complexity Sciences Center and Physics Department,*
*University of California at Davis, One Shields Avenue, Davis, CA 95616*

Mile Gu‡
*Complexity Institute, Nanyang Technological University, Singapore*
*School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore and*
*Centre for Quantum Technologies, National University of Singapore, Singapore*

(Dated: June 30, 2020)

Adaptive thermodynamic systems—such as a biological organism attempting to gain survival advantage, an autonomous robot performing a functional task, or a motor protein transporting intracellular nutrients—can improve their performance by effectively modeling the regularities and stochasticity in their environments. Analogously, but in a purely computational realm, machine learning algorithms seek to estimate models that capture predictable structure and identify irrelevant noise in training data by optimizing performance measures, such as a model's log-likelihood of having generated the data. Is there a sense in which these computational models are physically preferred? For adaptive physical systems we introduce the organizing principle that thermodynamic work is the most relevant performance measure of advantageously modeling an environment. Specifically, a physical agent's model determines how much useful work it can harvest from an environment. We show that when such agents maximize work production they also maximize their environmental model's log-likelihood, establishing an equivalence between thermodynamics and learning. In this way, work maximization appears 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

arXiv:2006.15416v1 [cond-mat.stat-mech] 27 Jun 2020

## I. INTRODUCTION

A debate has carried on for the last century and a half over the relationship (if any) between abiotic physical processes and intelligence. Though taken up by many scientists and philosophers, one important thread focuses on issues that lie decidedly at the crossroads of physics and intelligence.

Perhaps unintentionally, James Clerk Maxwell laid foundations for the physics of intelligence with what Lord Kelvin (William Thomson) referred to as "intelligent demons" [1]. Maxwell in his 1857 book *Theory of Heat* had argued that a "very observant" and "neat fingered being" could subvert the Second Law of Thermodynamics [2]. 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. Converting disorganized thermal energy to organized work energy, in this way, 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 [3, 4]. 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 information 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 *model* corresponds to *density estimation* [5],

* alecboy@gmail.com
† chaos@ucdavis.edu
‡ mgu@quantumcomplexity.org

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 [6]. Thus, by addressing thermodynamic roots of model estimation, we seek a physical foundation for a wide breadth of machine learning. More to the point, we imagine a future in which the pure computation employed in a machine learning system is instantiated so that the physical properties of its implementation are essential to its functioning. And, in any case, we hope to show that this setting provides a workable, though simplified, approach to the physical and informational trade-offs facing adaptive organisms.

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 as they adapt to their environments.

The modern understanding of Maxwell's demon no longer entertains violating the Second Law of Thermodynamics. 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. They do this by exploiting an information reservoir [7–9]. That reservoir and the organization of the demon's control and measurement apparatus are how modern physics views the embodiment of its intelligence [10].

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 environmental information. Leveraging this commonality, we introduce *thermodynamic learning* as a physical process that infers optimal demons from environmental information. Thermodynamic learning selects demons that produce maximum work, paralleling parametric density estimation's selection of models with maximum likelihood. The surprising result is that these two principles of maximization are the same, when compared in a common setting.

Technically, we show that a probabilistic model of its environmental is an essential part of the construction of an intelligent work-harvesting demon. That is, the demon's work production from environmental "training data" is proportional to the log-likelihood of the demon's

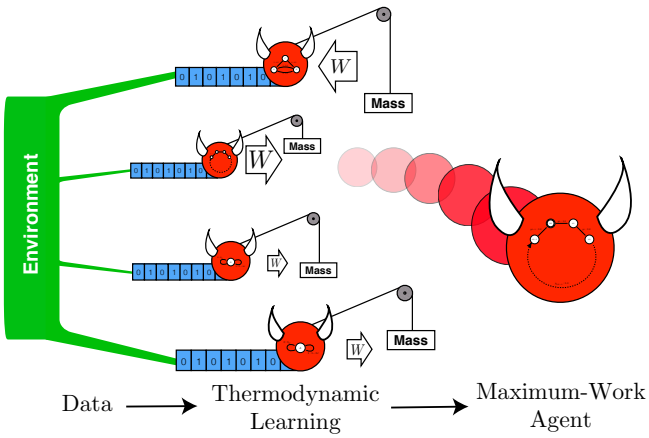


FIG. 1. Thermodynamic learning generates the maximum-work producing agent: (Left) Environment (green) behavior becomes data for agents (red). (Middle) Candidate agents each have an internal model (inscribed stochastic state-machine) that captures the environment's randomness and regularity to store work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

environment model. 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. In this way, *thermodynamic learning is machine learning for thermodynamic machines*—it 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. While it is natural to argue that learning confers benefits, our result establishes that the benefit is fundamentally rooted in the physics of energy and information.

Once these central results are presented and their interpretation explained, but before we conclude, we briefly recount the long-lived narrative of the thermodynamics of organization. This places the results in an historical setting and compares them to related works. We must first, however, explain the framework in which thermodynamic learning arises and then lay out the necessary technical background in density estimation, computational mechanics, thermodynamic computing, and thermodynamically-efficient computations. With these addressed, the use of work as a measure of learning performance is explored, ultimately deriving the equivalence between the conditions of maximum work and maximum likelihood.

## 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 energy, such as between thermal energy and work. Here, we posit a setting 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 $\tau$. 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 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 thus 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 know something about the black box's structure. This is the *principle of requisite complexity* [11]—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, which 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. Work energy is stored such that it can be retrieved later, for instance by raising a mass as depicted in Fig. 1. Then, the agent producing the most work is selected. This is "thermodynamic learning" in the sense that it selects a device based on measuring its thermodynamic performance—the amount of work it 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.

For clarity, note that thermodynamic learning differs from thermodynamic systems that, evolving in time, spontaneously adapt to their environment [12–14]. Work maximization as described here is thermodynamic in its objective, while these previous approaches are thermodynamic in their mechanism. That said, the perspectives are closely linked. In particular, it was suggested that thermodynamic systems spontaneously decrease work absorbed from driving [13]. Note that work absorbed by the system is opposite the work produced. And so, as they evolve over time, these thermodynamic systems appear to seek higher work production, paralleling how thermodynamic learning selects for the highest work production. Moreover, the adaptation by which a thermodynamic system decreases work absorption is often compared to learning [13]. Reference [14] goes further, comparing the effectiveness of thermodynamic evolution to maximum-likelihood estimation employing an autoencoder. Notably, it reports that the machine learning technique performs markedly better than the thermodynamic evolution, for the particular physical system considered there.

The framework here also compares 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. VI 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. [14], which compares thermodynamic and machine learning methods quantitatively, the framework here leads to an analytic derivation of the equivalence between thermodynamic learning and maximum-likelihood density estimation.

## III. BACKGROUND

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 more detail should refer to App. A.)

### A. Parametric Density Estimation

Parametric estimation determines, from training data, the parameters $\theta$ of a probability distribution. In the present setting, $\theta$ parametrizes a family of probabilities $\Pr(Y_{0:\infty} = y_{0:\infty}|\Theta = \theta)$ over words of arbitrary length, where $Y_j$ is the random variable for the environment's output at time $j\tau$ and $\Theta$ is the random variable for the model distribution. For convenience, we introduce the new random variables $Y_j^\theta$ that define the model:

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

With training data $y_{0:L}$, the likelihood of the model $\theta$ 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 $\mathcal{L}(\theta|y_{0:L})$ [5, 15]. However, the procedure of finding maximum-likelihood estimates usually employs the *log-likelihood* instead:

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

since it is maximized for the same models, but converges more effectively [16].

### 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 $\mathbf{\Theta} = \{\theta\}$ specify a wide range of possible distributions $\Pr(Y_{0:\infty}^\theta)$—the *semi-infinite processes*. $\epsilon$-Machines, a class of finite-state machines introduced to describe bi-infinite processes $\Pr(Y_{-\infty:\infty}^\theta)$ [17], provide a systematic means to do this. 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 explicitly constructed from the process.

A process' $\epsilon$-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 $\theta_{s\to s'}^{(y)}$ over the hidden states:

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

$\theta_{s\to s'}^{(y)}$ 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:

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

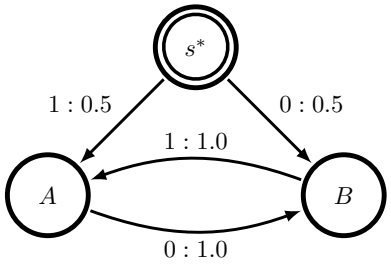


FIG. 2. $\epsilon$-Machine generating the phase-uncertain period-2 process: With probability 0.5, an initial transition is made from the start state $s^*$ to state $A$. From there is emits the sequence $1010\cdots$. And, with 0.5 probability, the start state transitions to the $B$ state and outputs the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

As an example, Fig. 2 shows an $\epsilon$-machine that generates a periodic process with initially uncertain phase.

$\epsilon$-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 function:

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

This yields a simple expression for the probability of any word in terms of the model parameters:

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

Thus, in addition to being uniquely determined by the semi-infinite process, the $\epsilon$-machine uniquely generates that same process, meaning that our model class $\Theta$ is equivalent to the class of possible distributions over time series data. Moreover, knowledge of the causal state of an $\epsilon$-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 $\epsilon$-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 it as the *system of interest*. Its physical states, denoted $\mathcal{Z} = \{z\}$, are taken as the *information bearing degrees of freedom* [8]. The system dynamic evolves the state distribution $\Pr(Z_t = z_t)$, where $Z_t$ is the random variable describing state at time $t$. Computation over the time interval $t \in [\tau, \tau']$ addresses how the dynamic maps the system 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, characterized by the conditional probability of transitioning to the final state $z_{\tau'}$ given the initial state $z_\tau$:

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

Together, these all the logical elements of the computation. In this, $z_\tau$ is the *input* to the physical computation, $z_{\tau'}$ is the *output*, and $M_{z_\tau \to z'_\tau}$ is the *logical architecture*.

Figure 3 illustrates a computation's physical implementation. The system of interest $\mathcal{Z}$ is coupled to a work reservoir, depicted as a mass hanging from a string, that controls the system's Hamiltonian along a trajectory $\mathcal{H}_\mathcal{Z}(t)$ over the interval of the computation $t \in [\tau, \tau']$ [18]. This is the basic definition of a thermodynamic agent.

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 system of interest and work reservoir. If the system $\mathcal{Z}$ follows the *state trajectory* over the time interval $t \in [\tau, \tau']$:

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

where $z_t$ is the system state at time $t$, then work production for that state trajectory is the integrated change in energy due to the Hamiltonian's time dependence [18]:

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

Note that this decomposes the state trajectory $z_{\tau:\tau'}$ into intervals of duration $dt$, chosen short enough to yield infinitesimal changes in state probabilities and the Hamiltonian. In this way, while the state trajectory $z_{\tau:\tau'}$ mirrors the time series notation used for our training data

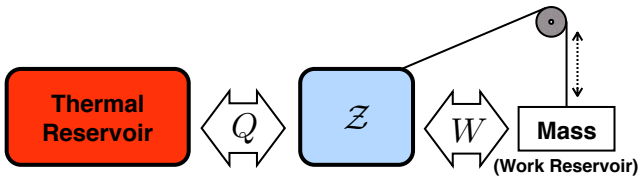


FIG. 3. Thermodynamic computing: The system of interest $\mathcal{Z}$'s physical states store information, processing it as they evolve. Work energy is supplied by the work reservoir, represented by the hanging mass. And, heat energy is supplied by the thermal reservoir.

$y_{0:L} = y_0 y_1 \cdots y_{L-1}$, it is physically different. On the one hand, the time series $z_{\tau:\tau'}$ represents a detailed tracking of states of a *single* system $\mathcal{Z}$ over time. On the other, the training data series $y_{0:L}$ is composed of realizations of $L$ *separate* subsystems of the environment, each produced at different times $j\tau$, $j \in \{0, 1, 2, \cdots\}$. As a result, elements of the series $z_{\tau:\tau'}$ necessarily depend on each other while $y_{0:L}$ has no such necessary dependency.

Assuming the system computes while coupled to a thermal reservoir at temperature $T$, Landauer's Principle [8] relates a computation's logical processing to its energetics. In its contemporary form, it bounds the *average* work production, denoted $\langle W \rangle$, by a term proportional to the system's entropy change. Setting $H[Z_t] = -\sum_z \Pr(Z_t = z) \ln \Pr(Z_t = z)$ as the Shannon entropy in natural units, the Second Law of Thermodynamics implies [4]:

$$\langle W \rangle \leq k_\mathrm{B} T \left( H[Z_{\tau'}] - H[Z_\tau] \right) \ .$$

Here, the average $\langle W \rangle$ is taken over all possible trajectories.

## IV. ENERGETICS OF COMPUTATIONAL MAPPINGS

This bound concerns the average work production over the ensemble of all possible states. However, thermodynamic learning uses the performance given a particular set of training data. Thus, the following evaluates the work production of a particular *computational mapping* $z_\tau \to z_{\tau'}$, which ignores the details of the state trajectory $z_{\tau:\tau'}$ by only tracking the input $z_\tau$ and output $z_{\tau'}$. To determine performance, we first consider an efficient mapping and then connect efficiency to maximum-likelihood.

### A. Efficient Computations

Let's estimate the maximum work production associated with a computational map $z_\tau \to z_{\tau'}$. That is, given a system following a state trajectory beginning in state $z_\tau$ and ending in $z_{\tau'}$, what is its associated work production at temperature $T$?

To do so, we first derive a useful relation between work $W_{|z_{\tau:\tau'}}$ and entropy production $\Sigma_{|z_{\tau:\tau'}}$ along a full state trajectory $z_{\tau:\tau'}$. The total entropy produced resulting from thermodynamic control is the sum of (i) the system of interest's entropy change [19]:

$$\Delta S^{\mathcal{Z}}_{|z_{\tau:\tau'}} = k_{\mathrm{B}} \ln \frac{\Pr(Z_\tau = z_\tau)}{\Pr(Z_{\tau'} = z_{\tau'})} \ ,$$

and (ii) the thermal reservoir's entropy change. The latter derives from the heat:

$$\Delta S^{\text{reservoir}}_{|z_{\tau:\tau'}} = \frac{Q_{|z_{\tau:\tau'}}}{T} \ .$$

Thus, recalling the First Law of Thermodynamics—system energy change is opposite work and heat production $\Delta E^{\mathcal{Z}} = -W - Q$—the total entropy production of a particular state trajectory can be expressed in terms of work production:

$$\begin{aligned} \Sigma_{|z_{\tau:\tau'}} &= \Delta S^{\text{reservoir}}_{|z_{\tau:\tau'}} + \Delta S^{\mathcal{Z}}_{|z_{\tau:\tau'}} \\ &= \frac{-W_{|z_{\tau:\tau'}} + \phi(z_\tau, \tau) - \phi(z_{\tau'}, \tau')}{T} \ . \end{aligned} \tag{2}$$

This collects the excess quantities into a change-of-state function, called the *pointwise nonequilibrium free energy*:

$$\phi(z_t, t) = E(z_t, t) + k_{\mathrm{B}} T \ln \Pr(Z_t = z_t) \ . \tag{3}$$

Its name derives from the averaged quantity $\langle \phi(z,t) \rangle_{\Pr(Z_t = z)} = F^{\text{neq}}(t)$, which is known as *nonequilibrium free energy* [20].

When considering computational maps, we care only about the initial and final states of the system. As such, we take a statistical average of all trajectories beginning in $z_\tau$ and ending in $z_{\tau'}$, which then defines:

$$\left\langle W_{|z_\tau, z_{\tau'}} \right\rangle = \sum_{z'_{\tau:\tau'}} W_{|z'_{\tau:\tau'}} \Pr(Z_{\tau:\tau'} = z'_{\tau:\tau'} | z_\tau, z_{\tau'}) \ , \tag{4}$$

representing the *computational mapping work production* for $z_\tau \to z_{\tau'}$. This is how much energy is stored in the work reservoir on average when the computation results in this particular input-output pair. Taking the same average conditioned on inputs and output of the entropy production in Eq (2) gives:

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

This suggests a relation between computational mapping work and the change in pointwise nonequilibrium free energy $\phi(z,t)$.

This relation becomes exact for thermodynamically-efficient computations. In such scenarios, where average total entropy production over all trajectories vanishes, App. B shows that this, combined with Crook's fluctuation theorem [21], implies that entropy production across any individual trajectory produces zero entropy: $\Sigma_{|z_{\tau:\tau'}} = 0$ for any $z_{\tau:\tau'}$. This is expected from linear response [22]. Thus, substituting zero entropy production into Eq. (2), we arrive at our result: *work production for thermodynamically-efficient computations is the change in pointwise nonequilibrium free energy*:

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

Substituting Eq. (3) then gives:

$$W^{\text{eff}}_{|z_{\tau:\tau'}} = -\Delta E_{\mathcal{Z}} + k_{\mathrm{B}} T \ln \frac{\Pr(Z_\tau = z_\tau)}{\Pr(Z_{\tau'} = z_{\tau'})} \ ,$$

where $\Delta E_{\mathcal{Z}} = E(z_{\tau'}, \tau') - E(z_\tau, \tau)$. This also holds if we average over intermediate states of the system's state trajectory, yielding the work production of a computational mapping:

$$\left\langle W^{\text{eff}}_{|z_\tau, z_{\tau'}} \right\rangle = -\Delta E_{\mathcal{Z}} + k_{\mathrm{B}} T \ln \frac{\Pr(Z_\tau = z_\tau)}{\Pr(Z_{\tau'} = z_{\tau'})} \ . \tag{5}$$

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.

### B. Thermodynamics of Misestimation

Even with perfectly-efficient thermodynamic control, misestimating the environment comes at a thermodynamic cost. If we estimate the input distribution $\Pr(Z^\theta_\tau)$ and output distribution $\Pr(Z^\theta_{\tau'})$, the natural choice is to design the computation to be efficient for those estimates. By minimizing the entropy production for the estimated distributions, we guarantee that the thermodynamic agent produces as much work as possible when it receives the estimated inputs. However, if it misestimates the input—such that over the computation interval $t \in [\tau, \tau']$ the actual and estimated input distributions differ: $\Pr(Z^\theta_t) \neq \Pr(Z_t)$—then the computation must at

least dissipate a minimum given by [23, 24]:

$$\langle \Sigma^{\min} \rangle = k_{\mathrm{B}} \left( D_{KL}(Z_\tau || Z_\tau^\theta) - D_{KL}(Z_{\tau'} || Z_{\tau'}^\theta) \right) \geq 0.$$

If Hamiltonian control misestimates inputs and outputs resulting in the lower bound on entropy being positive, then the computation is inefficient. One consequence is that the work production in Eq. (5) is no longer satisfied.

Failing Eq. (5), how can we find the work production of a protocol designed to be efficient for estimated input $\Pr(Z_\tau^\theta)$ and output $\Pr(Z_{\tau'}^\theta)$? Fortunately, the work produced by a computational mapping does not depend on the initial or final distribution. This work explicitly conditions on the initial state $z_\tau$ and final state $z_{\tau'}$. And so, it shields the resulting probability of the intervening state trajectory from the initial or final distributions. Thus, since Eq. (5) is satisfied when the input and output are the same as expected ($\Pr(Z_t^\theta) = \Pr(Z_t)$), it does not change when it receives any other input distribution. As a result, the work production of a computational mapping is entirely determined by the distributions for which the physical computation is designed to be efficient:

$$\left\langle W^{\mathrm{eff}}_{|Z_\tau^\theta = z_\tau, Z_{\tau'}^\theta = z_{\tau'}} \right\rangle = -\Delta E_{\mathcal{Z}} + k_{\mathrm{B}} T \ln \frac{\Pr(Z_\tau^\theta = z_\tau)}{\Pr(Z_{\tau'}^\theta = z_{\tau'})}. \tag{6}$$

The subscripts $Z_\tau^\theta$ and $Z_{\tau'}^\theta$ are added to the work production to indicate which distributions were anticipated by the Hamiltonian control. Equation (6) now gives an explicit relationship between useful work production $W$, finite data $z_\tau$ and $z_{\tau'}$, and the agent's model $\theta$. This is the first step in establishing work production as a thermodynamic performance metric for learning.

Focusing on the energetic benefit deriving from the information itself, rather than the benefit of changing energy levels, we set the beginning and ending energies to be the same. Thus, $\Delta E_{\mathcal{Z}} = 0$ and the resulting work production from a computational mapping is:

$$\boxed{\left\langle W^{\mathrm{eff}}_{|Z_\tau^\theta = z_\tau, Z_{\tau'}^\theta = z_{\tau'}} \right\rangle = k_{\mathrm{B}} T \ln \frac{\Pr(Z_\tau^\theta = z_\tau)}{\Pr(Z_{\tau'}^\theta = z_{\tau'})} \ .} \tag{7}$$

This measures the energetic gains from a single data realization as it transforms during a computation, as opposed to the ensemble average. Through precise control of the energy landscape, this energetic benefit is achievable. Section VIII describes a method to extract this work from a two-level system using an alternating process of instantaneous quenching, quasistatic evolution, then quenching again. This procedure generalizes to *any* computation, as shown in App. C. The following uses this result to design efficient agents that harvest energy from a time series.

However, before exploring thermodynamic learning from time series, let's apply these results to training on data coming from the system of interest $\mathcal{Z}$ itself. What work can be produced from an input $z_\tau$, regardless of the output, and how does it depend on the agent's internal model $\theta$? In addition, can an agent maximize the work from each input?

To address these, we design the computational architecture to fully randomize the output—$M_{z \to z'} = 1/|\mathcal{Z}|$—such that the final distribution is uniform $\Pr(Z_{\tau'} = z_{\tau'}) = 1/|\mathcal{Z}|$. In this setting, an agent extracts the maximum possible energy by expanding into the state space as much as possible. The resulting work produced on average from a particular input $z_\tau$ is:

$$\left\langle W^{\mathrm{eff}}_{|Z_\tau^\theta = z_\tau} \right\rangle = k_{\mathrm{B}} T \left( \ln \Pr(Z_\tau^\theta = z_\tau) + \ln |\mathcal{Z}| \right) \ . \tag{8}$$

Thus, an agent whose model maximizes the probability of particular input data produces the most work from that data.

From the machine learning perspective, the work production of an efficient agent operating on a single system increases proportionally to the log-likelihood (Eq. (1)) of the model $\theta$ given the input data:

$$\boxed{\left\langle W^{\mathrm{eff}}_{|Z_\tau^\theta = z_\tau} \right\rangle = k_{\mathrm{B}} T \left( \ell(\theta | z_\tau) + \ln |\mathcal{Z}| \right) \ .} \tag{9}$$

That is, thermodynamic learning leads to a work-maximizing agent that also maximizes the likelihood of its model of the given input. Thus, by maximizing work production, a designer builds thermodynamic agents that, as in machine learning, employ maximum-likelihood models.

## V. WORK PRODUCTION FROM A TIME SERIES

To determine the work production, as just set up, of time series of separate inputs $y_{0:L}$, it is tempting to take as our controlled system of interest $\mathcal{Z}$ the joint variable of all inputs $\mathcal{Y}^L$ and then apply a quasistatic channel that simultaneously controls the energy of all inputs simultaneously. However, this violates *temporal modularity* in the inputs—the fact that each must be interacted with separately. Such a strategy requires a global energy landscape $E(y_{0:L}, t)$. This is nonsensical, since each $y_j$ is made available at a different time $j\tau$. However, this does not mean that correlations between inputs cannot be addressed.

To harness the *temporal correlations* in a time series, we turn to *information ratchets* [25, 26]. These generalize

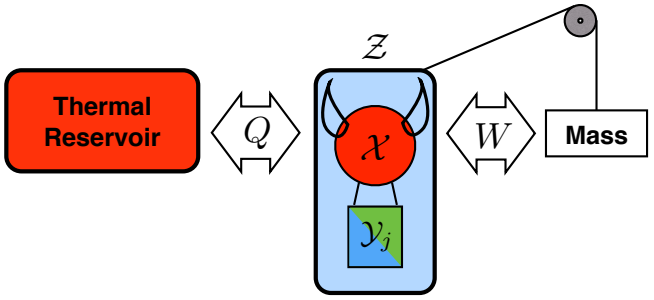


FIG. 4. Thermodynamic computing by an agent subject to an input: Information bearing degrees of freedom $\mathcal{Z}$ split into the direct product of agent states $\mathcal{X}$ and the $j$th input states $\mathcal{Y}_j$. Work and heat are defined correspondingly.

Maxwell's demon to harvest work from a series of inputs. Combining physical inputs with additional agent memory states that store the input's temporal correlations, we find the work production for single-shot short input strings. This contrasts with prior results that focused, instead, on ensemble-average work production [11, 25–31].

Given a sequence $y_{0:L}$ of inputs, with $y_j$ being the input provided at time $t = j\tau$, the information ratchet strategy for extracting work is to allow each input to interact as part of an autonomous agent that stores memory of past inputs. This modifies our notion of thermodynamic computing only slightly. As Fig. 4 illustrates, over the $j$th time step the information-bearing system of interest becomes the joint system $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}_j$, consisting of agent state and the $j$th interaction symbol. In an information ratchet, each symbol $\mathcal{Y}_j$ interacts with the agent's memory $\mathcal{X}$ over the interaction interval $[j\tau, j\tau + \tau']$, transforming the symbol stored in $\mathcal{Y}_j$ from input to output. Then, the agent's memory decouples and couples to the next symbol over the interval $[j\tau + \tau', (j+1)\tau]$ while its state preserves memory of past interactions. In this way, the agent uses its memory to transform a series of inputs $y_{0:L}$ to a series of outputs $y'_{0:L}$.

The functional and energetic components of this procedure occur during the *interaction interval* $[j\tau, j\tau + \tau']$, with the time interval $[j\tau + \tau', (j+1)\tau]$ serving simply as buffer time between the $j$th and $(j+1)$th interaction. $\mathcal{Y}_j$ is the *interaction-symbol* subsystem during the interaction interval and, along with the agent state, it evolves according to the framework for thermodynamic computing laid out in Sec. III C. Specifically, $j\tau$ takes the place of the initial time $\tau$ and $j\tau + \tau'$ takes the place of the final time $\tau'$. The Hamiltonian control over the joint space $\mathcal{H}_{\mathcal{X}\times\mathcal{Y}_j}(t)$ updates states according to a Markov transition matrix in the same way:

$$M_{xy\to x'y'} = \Pr(Z_{j\tau+\tau'} = x' \times y' | Z_{j\tau} = x \times y) \ .$$

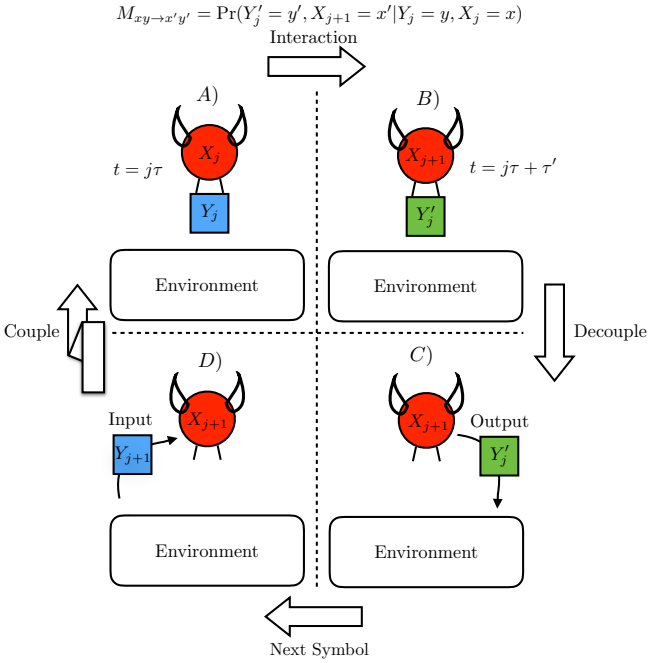


FIG. 5. Agent interacting with an environment via repeated symbol exchanges: A) At time $j\tau$ agent memory $X_j$ begins interacting with input symbol $Y_j$. Transitioning from A) to B), agent memory and interaction symbol jointly evolve according to the Markov channel $M_{xy\to x'y'}$. This results 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 $Y_{i+1}$. Finally, transitioning from D) to A), the agent restarts the cycle by coupling to the next input symbol.

This specifies the logical architecture of the ratchet's operation at each time step.

For convenience, we factor the joint system of interest random variable at the start of the interaction interval into separate components $Z_{j\tau} = X_j \times Y_j$. This gives a shorthand for the $j$th state of the agent $X_j$ and the $j$th input $Y_j$. We also factor the system into component variables at the end of the interaction interval $Z_{j\tau+\tau'} = X_{j+1} \times Y'_j$, with $Y'_j$ representing the symbol emitted back to the environment and $X_{j+1}$ representing the agent memory after the $j$th interaction, which is preserved for the next input. Figure 5 illustrates how this results in a general representation of thermodynamic *information transduction* [32]. Expressed in terms of the newly defined variables for inputs, outputs, and agent memory states, we have the agent's logical architecture:

$$M_{xy\to x'y'} = \Pr(X_{j+1} = x', Y'_j = y' | X_j = x, Y_j = y). \quad (10)$$

The results on efficient thermodynamic control above

now apply to this joint system. If the agent has a model $\Pr(X_j^\theta = x, Y_j^\theta = y)$ of its inputs at the beginning of each interaction interval, then it also has estimates of the output at the end of the interaction interval:

$$\Pr(X_{j+1}^\theta = x', Y_j^{'P} = y') = \sum_{x,y} \Pr(X_i^\theta = x, Y_i^\theta = y) M_{xy \to x'y'}.$$

With the computation designed to be efficient, the estimates determine the work production for a transition:

$$\begin{aligned} \left\langle W^{\text{eff}}_{j,x_j y_j \to x_{j+1} y'_j} \right\rangle &\equiv \langle W^{\text{eff}}_{|X_j^\theta = x, Y_j^\theta = y, X_{j+1}^\theta = x', Y_j^{'\theta} = y'} \rangle \\ &= k_{\text{B}} T \ln \frac{\Pr(X_j^\theta = x, Y_j^\theta = y)}{\Pr(X_{j+1}^\theta = x', Y_j^{'\theta} = y')} \ . \end{aligned} \tag{11}$$

When the estimated input distribution matches the actual distribution $\Pr(X_j = x, Y_j = y)$, the average work production takes on a familiar form [30, 33]:

$$\langle W^{\text{eff}}_{|X_j, Y_j, X_{j+1}, Y'_j} \rangle = k_{\text{B}} T (H[X_{j+1}, Y'_j] - H[X_j, Y_j]). \tag{12}$$

More to the point, Eq. (11)'s computational-mapping work production now allows us to calculate the work produced for particular input sequences.

We do this by first considering the work production of a particular sequence of agent memory states $x_{0:L+1}$ and outputs $y'_{0:L}$:

$$\begin{aligned} \left\langle W^{\text{eff}}_{|y_{0:L}, y'_{0:L}, x_{0:L+1}} \right\rangle &= \sum_{j=0}^{L-1} \langle W^{\text{eff}}_{j,x_j y_j \to x_{j+1} y'_j} \rangle \\ &= k_{\text{B}} T \ln \prod_{j=0}^{L-1} \frac{\Pr(X_j^\theta = x_j, Y_j^\theta = y_j)}{\Pr(X_{j+1}^\theta = x_{j+1}, Y_j^{'\theta} = y'_j)}. \end{aligned}$$

If the agent is designed to start in a distribution $\Pr(X_0^\theta)$ uncorrelated with its estimated input distribution $\Pr(Y_{0:L}^\theta)$, then it anticipates the distribution over the sequence of inputs, outputs, and agent states:

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

This gives the estimated distribution over the agent and input at time $j\tau$ from the marginal:

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

Though challenging to calculate, the next section illustrates that there are particular agents that have an efficient logical architecture $M$ for which this is straightforward to evaluate.

Before exploring the simplifications that arise for efficient agents, it is worth finding a general expression for the average work production resulting from a particular input string $y_{0:L}$. We do this by summing over all possible agent-state trajectories $x_{0:L+1}$ and output series $y'_{0:L}$. Using both actual and estimated initial distributions over agent states $\{\Pr(X_0), \Pr(X_0^\theta)\}$, the agent's logical architecture $M_{xy \to x'y'}$, and the actual and estimated distribution over inputs $\{\Pr(Y_{0:L}), \Pr(Y_{0:L}^\theta))\}$, the average work, in its full detail, is:

$$\begin{aligned} \left\langle W^{\text{eff}}_{|Y_{0:L}^\theta = y_{0:L}} \right\rangle &= k_{\text{B}} T \sum_{x_{0:L+1}, y'_{0:L}} \Pr(Y'_{0:L} = y'_{0:L}, X_{0:L+1} = x_{0:L+1} | Y_{0:L} = y_{0:L}) \ln \prod_{j=0}^{L-1} \frac{\Pr(X_j^\theta = x_j, Y_j^\theta = y_j)}{\Pr(X_{j+1}^\theta = x_{j+1}, Y_j^{'\theta} = y'_j)} \\ &= k_{\text{B}} T \sum_{x_{0:L+1}, y'_{0:L}} \Pr(X_0 = x_0) \prod_{k=0}^{L-1} M_{x_k, y_k \to x_{k+1}, y'_k} \\ &\quad \times \ln \prod_{j=0}^{L-1} \frac{\sum_{x'_{0:j}, y''_{0:j}, y'''_{0:j}} \Pr(X_0^\theta = x'_0) \Pr(Y_{0:j}^\theta = y''_{0:j}, Y_j^\theta = y_j) M_{x''_{j-1}, y''_{j-1} \to x_j, y'''_{j-1}} \prod_{l=0}^{j-2} M_{x'_l, y''_l \to x'_{l+1}, y'''_l}}{\sum_{x''_j, y''_j} M_{x''_j, y''_j \to x_{j+1}, y'_j} \sum_{x''_{0:j}, y''''_{0:j}, y'''''_{0:j}} \Pr(X_0^\theta = x''_0) \Pr(Y_{0:j+1}^\theta = y''''_{0:j+1}) \prod_{m=0}^{j-1} M_{x''_m, y''''_m \to x''_{m+1}, y'''''_m}}. \end{aligned} \tag{13}$$

This is 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 for input $\Pr(Y_{0:L}^\theta)$. This is a far cry from the simple expression for the work production of a single-step computation, expressed as a difference of log-likelihoods. Without simplifications, the difficulty of calculating this quantity scales poorly as the length $L$ of the input in-

creases. We include the expression here primarily to reinforce the challenge of calculating work production from the estimated input distribution $\Pr(Y^\theta_{0:\infty})$ and agent distribution $\Pr(X^\theta_0)$. Baring simplifications, it seems quite challenging to maximize an agent's work production for long input strings.

Despite unwieldiness, Eq. (13)'s work production is a deeply interesting quantity. In point of fact, since transducers are stochastic Turing machines [34], this is the work production for any general form of computation that maps inputs to output distributions $\Pr(Y'_{0:L}|Y_{0:L} = y_{0:L})$ [32]. Thus, Eq. (13) determines the work benefit possible for universal thermodynamic computing. Prior analyses showed that these ratchet's have thermodynamic functionality including, but not limited to, (i) expending work to generate patterns [26, 31] and (ii) harnessing temporal correlations to extract work [11, 27]. The following turns to focus on the latter, specifically restricting attention to information engines designed to produce maximum work from correlated inputs. These are the *information extractors* of Refs. [31, 33]. This leads to considerable simplifications.

## VI. DESIGNING EFFICIENT AGENTS

As Sec. III B noted, our estimated semi-infinite input process $\Pr(Y^\theta_{0:\infty})$ has a unique minimal model—the $\epsilon$-machine:

$$\theta^{(y)}_{s\to s'} = \Pr(Y^\theta_j = y, S^\theta_{j+1} = s' | S^\theta_j = s) \ .$$

What does the estimated model $\theta$ tell us about an agent that effectively transforms one of those inputs into useful work? The answer is found in the tuple $\{M, \Pr(X^\theta_j, Y^\theta_j)\}$, which characterizes the agent. Recall that $M$ is the agent's logical architecture and $\Pr(X^\theta_j, Y^\theta_j)$ is the estimated input distribution at time-step $j$. Together, these fully determine the work production of an efficient agent. For an information extractor to avoid the thermodynamic cost of modularity, the logical architecture $M_{xy\to x'y'}$ must be constructed such that the memory-state variables $X_j$ are predictive of the input [33]. Thus, the $\epsilon$-machine generator provides a prescription for the transitions between the states of the joint input-and-memory variable.

Appendix E shows that the $\epsilon$-machine specifies exactly how to construct an agent $\{M, \Pr(X^\theta_j, Y^\theta_j)\}$ that efficiently harvests work from the input $\Pr(Y^\theta_{0:\infty})$. The logical architecture $M$ is given by the $\epsilon$-machine's $\epsilon$-function that maps histories to causal states:

$$M_{xy\to x'y'} = \frac{1}{|\mathcal{Y}_j|} \times \begin{cases} \delta_{x',\epsilon(x,y)}, \text{ if } \sum_{x'} \theta^{(y)}_{x\to x'} \neq 0, \\ \delta_{x',x}, \text{ otherwise.} \end{cases} \quad (14)$$

The estimated input is given by the $\epsilon$-machine's hidden Markov model:

$$\Pr(Y^\theta_j = y_j, X^\theta_j = s_j) = \sum_{y_{0:j}, s_{0:j}, s_{j+1}} \delta_{s_0,s^*} \prod_{k=0}^{j} \theta^{(y_k)}_{s_k\to s_{k+1}} \ .$$

Conversely, the efficient agent, characterized by its logical architecture and anticipated input distributions $\{M, \Pr(X^\theta_j, Y^\theta_j)\}$, also specifies the $\epsilon$-machine's model of the estimated distribution:

$$\begin{aligned} \theta^{(y)}_{s\to s'} &= \Pr(Y^\theta_j = y | S^\theta_j = s)\delta_{s',\epsilon(s,y)} \\ &= \Pr(Y^\theta_j = y | X^\theta_j = s)|\mathcal{Y}_j| M_{sy\to s'y'} \ . \end{aligned} \quad (15)$$

Through the $\epsilon$-machine, the agent also specifies its estimated input process.

Figure 6 illustrates an example in which an initially-uncertain-phase process (left) drives an agent with a matching internal model (middle). The result is a thermodynamically-efficient agent (far right). This demonstrates the bijection between (i) the anticipated input distribution, (ii) the $\epsilon$-machine, which is the estimated model, and (iii) the agent that is designed to efficiently harness that input. In this way, the agent's effectiveness at harnessing work from finite data is directly associated with the model that underlies that agent's architecture. And so, from this point forward, when discussing an estimated process or an $\epsilon$-machine that generates that guess, we are also describing the unique thermodynamic agent designed to produce maximal work from the estimated process.

## VII. WORK AS A PERFORMANCE MEASURE

Recall that our goal is to explore work production as a performance measure for a model estimated from a time series $y_{0:L}$. Section V calculated the work production from a time series for an agent. The result, though, was an unwieldy expression with no seeming connection to the agent's underlying model of the data. However, using predictive models $\theta$, Sec. VI conveniently provided the design of agents that efficiently harness work energy with the model $\theta$ built in. Appendix D shows that using efficiently-designed predictive agents leads to a consider-

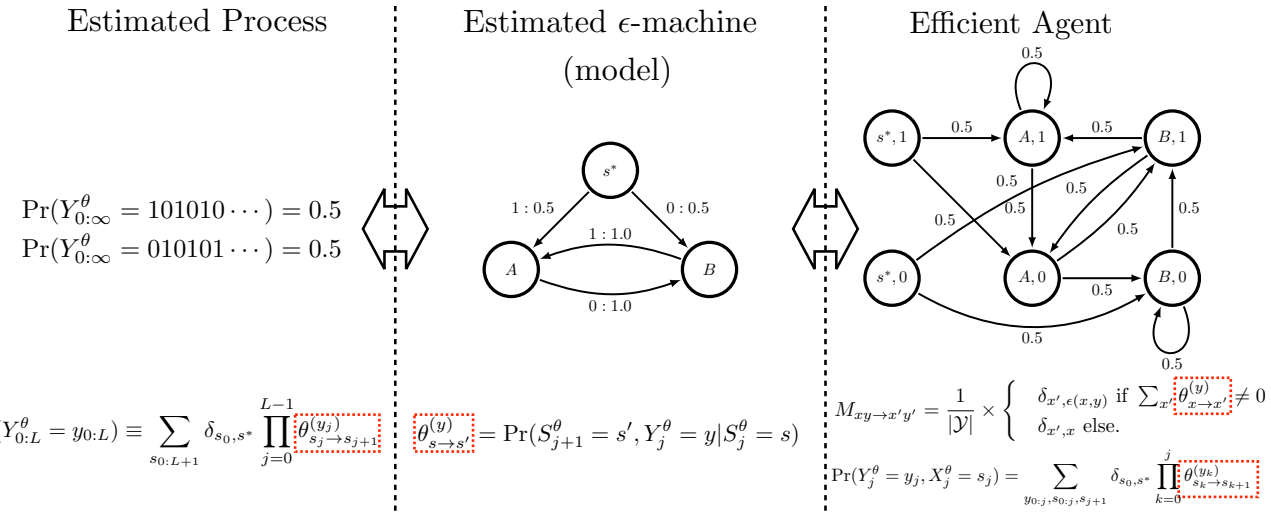


FIG. 6. Estimated input process $\Pr(Y^\theta_{0:\infty} = y_{0:\infty})$, $\epsilon$-machine $\theta$, and agent that efficiently harnesses the input process asymptotically, using logical architecture $M_{xy\to x'y'} = \Pr(X_{j+1} = x', Y_{j+1} = y' | X_j = x, Y_j = y)$ and estimated input distribution $\Pr(X^\theta_j = x, Y^\theta_j = y)$, are equivalent. Determining one determines the others.

able simplification of the sequence work production:

$$\boxed{\left\langle W^{\text{eff}}_{|Y^\theta_{0:L}=y_{0:L}} \right\rangle = k_{\text{B}}T(\ln \Pr(Y^\theta_{0:L} = y_{0:L}) + L \ln |\mathcal{Y}|).} \quad (16)$$

This vastly reduces the complexity of the work-production expression to a difference of the log-probabilities between the input distribution $\Pr(Y^\theta_{0:L} = y_{0:L})$ and output distribution $\Pr(Y'^P_{0:L} = y'_{0:L}) = 1/|\mathcal{Y}|^L$. This form is familiar, nearly exactly reproducing that of Eq. (8), which determines the work harvested by an efficient agent extracting energy from a single realization of a physical system $Z_\tau$. However, it should be emphasized that achieving this work production relies on including the additional agent memory $\mathcal{X}$. That memory allows us to account for the temporal modularity of the input by storing temporal correlations [33].

Thus, thermodynamically-efficient pattern extractors offer a substantial simplification when calculating agent work production. Much of the advantage derives from unifilarity, which guarantees a single hidden state trajectory $x_{0:L+1}$ for an input $y_{0:L}$. Even calculating the probability of the output reduces to tracking the terms for a particular causal-state trajectory $s_{0:L+1}$, where $s_j = \epsilon(s^*, y_{0:j})$:

$$\begin{aligned} \Pr(Y^\theta_{0:L} = y_{0:L}) &\equiv \sum_{s_{0:L+1}} \delta_{s_0,s^*} \prod_{j=0}^{L-1} \theta^{(y_j)}_{s_j \to s_{j+1}} \\ &= \prod_{i=0}^{L-1} \theta^{(y_j)}_{\epsilon(s^*,y_{0:j}) \to \epsilon(s^*,y_{0:j+1})} \ . \end{aligned}$$

Moreover, the model-dependent term in thermodynamically-efficient agent work production is familiar and interpretable—the model $\theta$'s log-likelihood:

$$\begin{aligned} \boxed{\left\langle W^{\text{eff}}_{|Y^\theta_{0:L}=y_{0:L}} \right\rangle} &= k_{\text{B}}T\ell(Y^\theta_{0:L}|y_{0:L}) + k_{\text{B}}TL \ln |\mathcal{Y}| \\ &= k_{\text{B}}T\ell(\theta|y_{0:L}) + k_{\text{B}}TL \ln |\mathcal{Y}|. \end{aligned} \quad (17)$$

Since efficient agents are characterized by the model of their environment—the $\epsilon$-machine—Eq. (17) suggests a parallel between machine learning and the thermodynamic processes that harness work from a finite string. If we treat $y_{0:L}$ as training data for the model, then the log-likelihood is maximized when agent's model anticipates the input with highest probability. This is the same condition for the thermodynamic agent extracting maximum work. Thus, the criterion for *creating a good model of an environment is the same as that for extracting maximal work.*

## VIII. TRAINING SIMPLE AGENTS

We now outline a simple version 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 last section. Lastly, we find the agent selected by thermodynamic learning along with it's corresponding model. As expected, we find that this maximum-work producing agent learns the features of its environment.

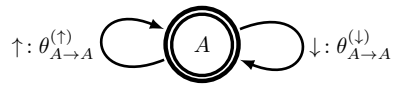


FIG. 7. Memoryless model of binary data consisting of a single state $A$ and the probability of outputting a $\uparrow$ and a $\downarrow$, denoted $\theta^{(\uparrow)}_{A\to A}$ and $\theta^{(\downarrow)}_{A\to A}$, respectively.

### A. 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' internal models correspond to memoryless $\epsilon$-machines, as shown in Fig. 7. The model's parameters are the probabilities of emitting $\uparrow$ and $\downarrow$, denoted $\theta^{(\uparrow)}_{A\to A}$ and $\theta^{(\downarrow)}_{A\to A}$, 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 $\mathcal{H}_{\mathcal{Z}}(t) = \mathcal{H}_{\mathcal{X}\times\mathcal{Y}_j}(t)$ 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. 8. The thermodynamic evolution there instantaneously quenches the energy landscape into equilibrium with the estimated distribution at the beginning of the interaction interval $\Pr(Z^\theta_{j\tau})$, then quasistatically evolves the system in equilibrium to the estimated final distribution $\Pr(Z^\theta_{j\tau+\tau'})$, and, finally, quenches back to the default energy landscape. In Fig. 8, the system undergoes a cycle, starting and ending with the same flat energy landscape, such that $\Delta E_{\mathcal{Z}} = 0$. This cycle evolves the distribution over the joint states $A\times\uparrow$ and $A\times\downarrow$ from $\Pr(Z^\theta_{j\tau} = \{A\times\uparrow, A\times\downarrow\}) = \{0.8, 0.2\}$ to $\Pr(Z^\theta_{j\tau+\tau'} = \{A\times\uparrow, A\times\downarrow\}) = \{0.4, 0.6\}$. 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 $\tau'$. 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:

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

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

$$E(z,t) = -k_{\mathrm{B}}T \ln \Pr(Z^\theta_t = z) \ .$$

The resulting process produces zero work:

$$\begin{aligned} W^{\text{quasistatic}} &= -\int_{z=j\tau}^{j\tau+\tau'} dt \sum_z \Pr(Z^\theta_t = z)\partial_t E(Z_t = z) \\ &= 0 \end{aligned}$$

and maps $\Pr(Z_{j\tau})$ to $\Pr(Z_{j\tau+\tau'})$ 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 to the initial quench:

$$E(x\times y, j\tau) - E(x\times y, j\tau^+) = k_{\mathrm{B}}T \ln \Pr(Z^\theta_{j\tau} = x\times y)$$

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

$$\begin{aligned} E(x'\times y', j\tau+\tau'^-) - E(x'\times y', j\tau+\tau') \\ = -k_{\mathrm{B}}T \ln \Pr(Z^\theta_{j\tau+\tau'} = x'\times y') \ . \end{aligned}$$

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

$$\left\langle W_{|x\times y, x'\times y'} \right\rangle = k_{\mathrm{B}}T \ln \frac{\Pr(Z^\theta_{j\tau} = x\times y)}{\Pr(Z^\theta_{j\tau+\tau'} = x'\times y')} \ . \quad (18)$$

Recall from Sec. V that the ratchet system variable $Z^\theta_{j\tau} = X^\theta_j \times Y^\theta_j$ is split into the random variable for the $j$th agent memory state and the $j$th input. Similarly, $Z^\theta_{j\tau+\tau'} = X^\theta_{j+1} \times Y^{'\theta}_j$ is split into the $(j+1)$th agent memory state and $j$th output. This work production achieves the efficient value described in Eq. (7).

Appendix C generalizes the thermodynamic operation above to *any* computation $M_{z_\tau \to z_{\tau'}}$. While it requires an ancillary copy of the system $\mathcal{Z}$ to execute the conditional dependencies in the computation, it is conceptually identical in that it uses a sequence of quenching, evolving quasistatically, and then quenching again. This appendix extends the strategies outlined in Refs. [31, 33] to computational-mapping work calculations.

### B. Efficient Information Ratchets

With the method for efficiently mapping inputs to outputs in hand, we can design a series of such computations

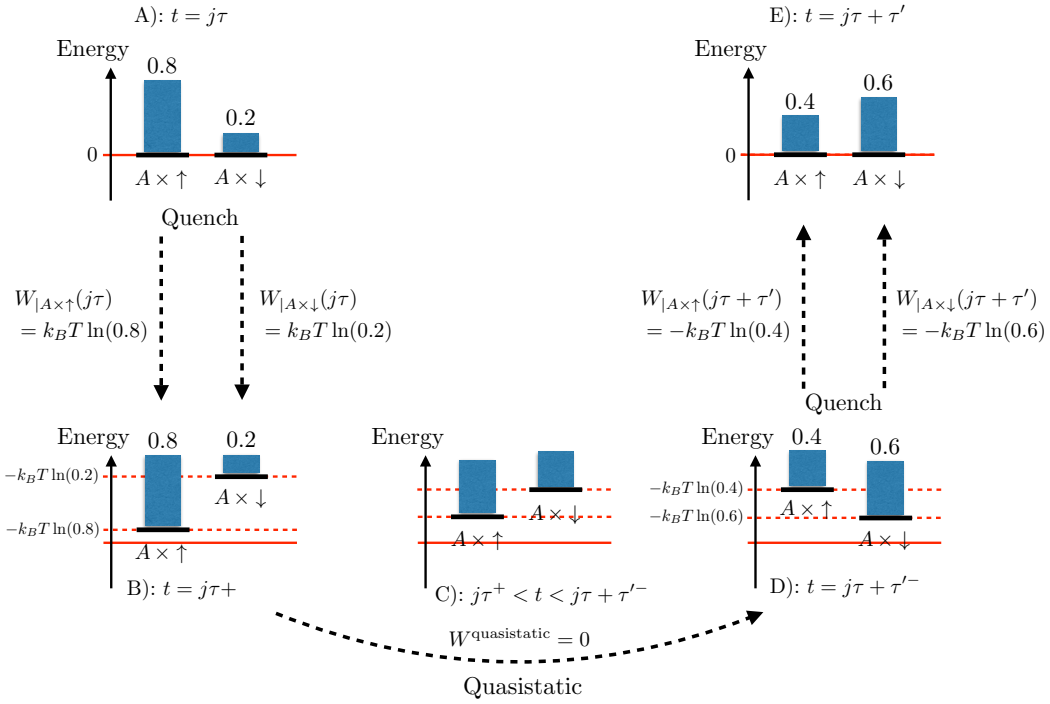


FIG. 8. Joint two-level system $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}_j = \{A \times \uparrow, A \times \downarrow\}$ undergoing perfectly-efficient computation when it receives its estimated input through a series of operations. The computation occurs over the time interval $t \in (j\tau, j\tau + \tau')$. 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 $\Pr(Z^\theta_{j\tau} = \{A \times \uparrow, A \times \downarrow\}) = \{0.8, 0.2\}$. 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 $\Pr(Z^\theta_{j\tau+\tau'} = \{A \times \uparrow, A \times \downarrow\}) = \{0.4, 0.6\}$, 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 $A \times y \to A \times y'$ is given by the work from the initial quench $W_{|A\times y}(j\tau)$ plus the work from the final quench $W_{|A\times y'}(j\tau + \tau')$.

to implement a simple information ratchet that produces work from a series $y_{0:L}$. As prescribed in Eq. (14) of Sec. VI, to produce the most work from estimated model $\theta$, the agent's logical architecture should randomly map every state to all others:

$$M_{xy\to x'y'} = \frac{1}{|\mathcal{Y}_j|} = \frac{1}{2} ,$$

since there is only one causal state $A$. In conjunction with Eq. (15), 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:

$$\begin{aligned}\Pr(Z^\theta_{j\tau} = x \times y) &= \Pr(X^\theta_j = x, Y^\theta_j = y) \\ &= \Pr(Y^\theta_j = y | X^\theta_j = A) \Pr(X^\theta_j = A) \\ &= \theta^{(y)}_{A\to A} ,\end{aligned}$$

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

$$\begin{aligned}\Pr(Z^\theta_{j\tau+\tau'} = x' \times y') &= \sum_{xy} \Pr(X^\theta_j = x, Y^\theta_j = y) M_{xy\to x'y'} \\ &= \frac{1}{2} .\end{aligned}$$

Thus, assuming the agent has model $\theta$ built-in, then Eq. (18) determines that the work production for mapping $A \times y$ to output $A \times y'$ for a particular symbol $y$ is:

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

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:

$$\left\langle W_{|y} \right\rangle = \left\langle W_{|A\times y, A\times y'} \right\rangle . \tag{19}$$

With the work production expressed for a single input $y_j$, we can now consider how much work our designed agent harvests from the binary training data $y_{0:L}$. Summing the work production of each input yields a simple expression in terms of the model $\theta$:

$$\begin{aligned}\left\langle W_{|y_{0:L}} \right\rangle &= \sum_{j=0}^{L-1} \left\langle W_{|y_j} \right\rangle \\ &= \sum_{j=0}^{L-1} k_{\mathrm{B}} T \left( \ln 2 + \ln \theta^{(y_j)}_{A \to A} \right) \\ &= k_{\mathrm{B}} T \left( L \ln 2 + \ln \prod_{j=0}^{L-1} \theta^{(y_j)}_{A \to A} \right) \ .\end{aligned}$$

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

$$\begin{aligned}\langle W_{|y_{0:L}} \rangle &= k_{\mathrm{B}} T \left( L \ln 2 + \ell(\theta | y_{0:L}) \right) \\ &= \langle W^{\mathrm{eff}}_{|Y^{\theta}_{0:L} = y_{0:L}} \rangle \ ,\end{aligned}$$

again, achieving efficient work production, as expected.

### C. Inferring 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:

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

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

$$\begin{aligned}\left\langle W^{\mathrm{eff}}_{|Y^{\theta}_{0:L} = y_{0:L}} \right\rangle &= k_{\mathrm{B}} T \ell(\theta | y_{0:L}) + k_{\mathrm{B}} T L \ln 2 \\ &= k_{\mathrm{B}} T L \left( \ln 2 + f(\uparrow) \ln \theta^{(\uparrow)}_{A \to A} + f(\downarrow) \ln \theta^{(\downarrow)}_{A \to A} \right) .\end{aligned}$$

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

$$\left\langle W^{\mathrm{eff}}_{|Y^{\theta}_{0:L} = y_{0:L}} \right\rangle = k_{\mathrm{B}} T L (\ln 2 - H[f(\uparrow)]) \ ,$$

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 learning. It 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 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. To illustrate how to extract this additional energy, we design and analyze memoryful agents [27] in a sequel.

## IX. SEARCHING FOR PRINCIPLES OF ORGANIZATION

Introducing a *principle of maximum work production* 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 was found by Kirchhoff [35]: 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 [36], Maxwell's principles of minimum-heat [37, pp. 407-408], and Onsager's minimizing the "rate of dissipation" [38].

Close to equilibrium Prigogine introduced minimum entropy production [39], identifying *dissipative structures* whose maintenance requires energy [40]. 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 [41]. 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 [42–44]. A canonical example is fluid turbulence [45]—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 [46, 47]. 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 [48, 49], appropriately extended [50]. While adding an important new perspective—that organized systems store and transmit information—this, also, did not go far enough as it sidestepped the content and meaning of information [51]. Inroads to these appeared in the theory of computation inaugurated by Turing [52]. The most direct and ambitious approach to the role of information in organization, though, appeared in Wiener's cybernetics [53, 54]. 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 [55] and Minimum Entropy Production Principles [56]—a link that in many ways is responsible for modern machine learning.

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* [57]. With this, one can extract from behavior a system's information processing, even going so far as to discover the effective equations of motion [58–61]. 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 [62] and Landauer and Bennett [8, 63], today there is a nascent *physics of information* [4, 64]. Resource constraints on computing by thermodynamic systems are now expressed in a suite of new principles. For example, the *information processing Second Law* [26] places a lower bound on the work required to perform a given amount of information processing. The *principle of requisite complexity* [11] 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 [33].

To fully appreciate organization in life processes one must also address dynamics of agent populations, first on the time scale of agent life cycles and second on the scale of generational reproduction. In fact, tracking the complexity of individuals reveals that selection pressures spontaneously emerge in purely-replicating populations [65] and replication itself necessarily dissipates energy [66].

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 [53].

## X. CONCLUSION

Coming in this historical setting, our main development introduced thermodynamic machine learning—a means for training intelligent agents to maximize 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 platform. We found that the performance of such agents increases proportionally to the log-likelihood of the model they used 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, 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, these results hint at a natural method for design of intelligent energy harvesters—establishing that our present tools of machine learning can be directly mapped towards automated design of more efficient information ratchets and pattern engines [11, 31, 67]. Meanwhile, recent results hint that quantum systems are capable to generating certain complex adaptive behavior with less resources than

classical counterparts [68–70]. The challenge now is to explore how the principle of maximum work production generalizes to quantum agents—Does this lead to new classes of quantum-enhanced energy harvesters or learners?

Ultimately, energy is an essential currency for life. This elevates 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 [13, 14], 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 wondrous if this evolution then has a physical basis—a hidden Fourth Law of Thermodynamics that guides the universe to creative 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. 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, 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 contracts W911NF-13-1-0390 and W911NF-18-1-0028. 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

Our development of the principle of maximum work production and understanding the physical benefits of an agent modeling its environment drew heavily from the fields of computational mechanics, nonequilibrium thermodynamics, and machine learning. Due to the variety of topics addressed, the following provides a more detailed summary of the notation and concepts. Hopefully, this will help the development be somewhat more self-contained. Our fondest hope is that it provide a common language between oft-segregated disciplines and a foundation for better appreciating the principle. While we make suggestive comparisons by viewing foundations of each field side-by-side, it may be appropriate for a reader who is familiar with the fields and most concerned with novel results to skip this, using it only to clarify any notation that unfamiliar in the main text.

### 1. Machine Learning and Generative Models

Thermodynamically, what is a good model of the environment with which an agent interacts? Denote the environment'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 $\vec{z} = \{z_0, z_1, \cdots, z_N\}$ 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 [5, 15, 71].

At first blush, estimating a probability distribution appears distinct from familiar machine learning challenges, such as image classification and 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 [6]. 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 $\theta$ 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 [5, 71]. However, we take the random variable for the estimated distribution $Z^\theta$ to denote the model for notational and conceptual convenience.

The Shannon entropy [49]:

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

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:

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

respectively.

While there are many other ways to create parametric models $\theta$—from polynomial functions with a small number of parameters to neural networks with thousands [5]—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 $\theta$ given a data point $z$ is the same as the likelihood of $Z^\theta$:

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

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

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

This is a commonly used performance measure in machine learning, where algorithms search for models with maximum likelihood [5]. However, it is common to use the *log-likelihood* instead, which is maximized for the same models:

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

If the model $Z^\theta$ were specified by a neural network, the log-likelihood could be determined through stochastic gradient descent back-propagation [15, 72], 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" [73], 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 [18]:

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

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:

$$\Delta S^{\text{reservoir}}_{|z_{\tau:\tau'}} = \frac{Q_{|z_{\tau:\tau'}}}{T} \ . \tag{A3}$$

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 $\mathcal{Z}$):

$$\langle \Sigma \rangle = \langle \Delta S^{\text{reservoir}} \rangle + \langle \Delta S^{\mathcal{Z}} \rangle \geq 0 \ . \tag{A4}$$

This constrains the energetic cost of computations per-

formed 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 the 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 $z_\tau$:

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

This specifies, in turn, the final distribution $\Pr(Z_{\tau'} = z_{\tau'})$ that allows direct calculation of the system-entropy change [19]:

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

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:

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

Here, $\phi(z,t) = E(z,t) + k_{\mathrm{B}} T \ln \Pr(Z_t = z)$ is the pointwise nonequilibrium free energy, which becomes the nonequilibrium free energy when averaged $\langle \phi(z,t) \rangle_{\Pr(Z_t = z)} = F^{\text{neq}}(t)$ [20].

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*:

$$W^{\text{diss}}_{|z_\tau : z_{\tau'}} = T \Sigma_{|z_\tau : z_{\tau'}} \ . \tag{A5}$$

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 $E(z,\tau) = E(z',\tau')$ [18], this reduces to Landauer's familiar principle for erasure [8]—work production must not exceed the change in state uncertainty:

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

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 $\Pr(Y^\theta_{0:L} = y_{0:L})$. 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:

$$\theta^{(y)}_{s \to s'} = \Pr(S^\theta_{j+1} = s', Y^\theta_j = y | S^\theta_j = s) \text{ for all } j \ ,$$

and a start state $s^* \in \mathcal{S}$. We label the transition probabilities with the model parameter $\theta$, 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 $y : \theta^{(y)}_{s \to s'}$, where $y$ is the output symbol and $\theta^{(y)}_{s \to s'}$ is the probability of emitting that symbol on that transition.

If $\{\theta^{(y)}_{s \to s'}\}$ 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:

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

Beyond generating length-$L$ symbol strings, these HMMs generate distributions over semi-infinite strings $\Pr(Y^\theta_{0:\infty} = y_{0:\infty})$. As such, they allow us to anticipate more than just the first $L$ symbols from the same source. Once we have a model $\theta = \{(\theta^{(y)}_{s \to s'}, s, s', y)\}_{s,s',y}$ from our training data $y_{0:L}$, we can calculate probabilities of longer words $\Pr(Y^\theta_{0:L'} = y'_{0:L'})$ and, thus, the probability of symbols following the training data $\Pr(Y^\theta_{L:L'} = y'_{L:L'} | Y^\theta_{0:L} = y_{0:L})$.

Distributions over semi-infinite strings $\Pr(Y^\theta_{0:\infty} = y_{0:\infty})$ are similar to processes, which are distributions over bi-infinite strings $\Pr(Y^\theta_{-\infty:\infty} = y_{-\infty:\infty})$. 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 $S^\theta_j$ are minimal sufficient statistics of the past $Y^\theta_{0:j}$ about the future $Y^\theta_{j:\infty}$ [17]. In other words, the hidden states are perfect predictors.

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

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

An equivalence class of histories is a *causal state*. Causal states also induced a map $\epsilon(\cdot)$ from histories to states:

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

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:

$$\Pr(Y^\theta_{k:\infty}|Y^\theta_{0:k} = y_{0:k}) = \Pr(Y^\theta_{k:\infty}|S^\theta_k = \epsilon(y_{0:k})) \ .$$

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 $\epsilon$ 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 $\theta^{(y)}_{s\to s'}$, which we will refer to as a *nonstationary* $\epsilon$-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:

$$\theta^{(y)}_{s\to s'} = \theta^{(y)}_{s\to\epsilon(s,y)}\delta_{s',\epsilon(s,y)} \ .$$

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

$$\theta^{(y)}_{s\to s'} = \Pr(Y^\theta_j = y|S^\theta_j = s)\delta_{s',\epsilon(s,y)} \ ,$$

where the conditional probability is determined from the process:

$$\Pr(Y^\theta_j = y|S^\theta_j = \epsilon(y_{0:i})) = \Pr(Y^\theta_j = y|Y^\theta_{0:j} = y_{0:j}) \ .$$

Once constructed, the $\epsilon$-machine allows us to reconstruct word probabilities with the simple product:

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

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 $\epsilon$-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 [74].

In these ways, computational mechanics had already solved (and several decades prior) the unsupervised learning challenge recently posed by Ref. [75] to create an "AI Physicist": a machine that learns regularities in time series to make predictions of the future from the past [76].

### b. *Input-Output Machines*

In this way one constructs a predictive HMM that generates a desired semi-infinite process $\Pr(Y^\theta_{0:L} = y_{0:L})$. A generalization, called an *$\epsilon$-transducer* [32] allows for an *input* as well as an output process. The transducer at the $i$th time step is described by transitions among the hidden states $X_i \to X_{i+1}$, which are conditioned on the input $Y_i$, yielding the output $Y'_i$:

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

$\epsilon$-Transducer state-transition diagrams label the edges of transitions between hidden states $y'|y : M^{(y'|y)}_{x\to x'}$ as shown in Fig. 9, to be read as the probability of output $y'$ and hidden state $x'$ given input $y$ and hidden state $x'$ is $M^{(y'|y)}_{x\to x'}$.

These devices are memoryful channels, with their memory encoded in the hidden states $X_i$. They implement a wide variety of functional operations. Figure 9 shows the *delay channel*. With sufficient memory,

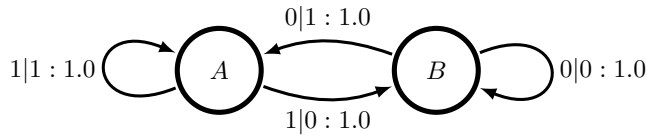


FIG. 9. The delay channel $\epsilon$-transducer: The last input symbol is stored in its memory (states). If the last symbol was 1, then the corresponding transitions, labeled $y'|1:1.0$, update the hidden state to $A$. Then, all outputs from $A$ are symbol 1. Similarly, input 0 leads to state $B$, whose corresponding outputs are all 0. In this way, the delay channel outputs the previous input symbol.

though, an $\epsilon$-transducer can implement a universal Turing machine [34]. 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 [25, 26]. 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 $\langle \Sigma \rangle = 0$. Crooks' fluctuation theorem says that entropy production is proportional to the log-ratio of probabilities [21]:

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

where $\rho_F(\Sigma_{|z_{\tau:\tau'}})$ is the probability of the entropy production under the control protocol 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 [49]:

$$\begin{aligned} \langle \Sigma \rangle &= k_{\mathrm{B}} \sum_{z_{\tau:\tau'}} \rho_F(\Sigma_{|z_{\tau:\tau'}}) \ln \frac{\rho_F(\Sigma_{|z_{\tau:\tau'}})}{\rho_R(-\Sigma_{|z_{\tau:\tau'}})} \\ &\equiv k_{\mathrm{B}} D_{KL}(\rho_F(\Sigma_{|z_{\tau:\tau'}}) || \rho_R(-\Sigma_{|z_{\tau:\tau'}})) \ . \end{aligned}$$

If the control is thermodynamically efficient, this relative entropy vanishes [77], 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 $M_{z \to z'} = \Pr(Z_{\tau'} = z' | Z_\tau = z)$ 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:

$$\begin{aligned} \left\langle W^{\mathrm{eff}}_{|Z^\theta_\tau = z_\tau, Z^\theta_{\tau'} = z_{\tau'}} \right\rangle &= \phi(z_{\tau'}, \tau') - \phi(z_\tau, \tau) \\ &= \Delta E_{\mathcal{Z}} + k_{\mathrm{B}} T \ln \frac{\Pr(Z^\theta_{\tau'} = z_{\tau'})}{\Pr(Z^\theta_\tau = z_\tau)} \ . \end{aligned} \tag{C1}$$

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

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. Fortunately, this can be circumvented by additional ancillary or hidden states [78]. 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 $\mathcal{Z}_{\mathrm{total}} = \mathcal{Z} \times \mathcal{Z}'$.

Prescriptions have been given that efficiently implement any computation, specified by a Markov channel $M_{z_\tau \to z_{\tau'}} = \Pr(Z_{\tau'} = z_{\tau'} | Z_\tau = z_\tau)$, using quasistatic manipulation of the $\mathcal{Z}$'s energy levels and an ancillary copy $\mathcal{Z}'$ [31, 33]. However, these did not determine the work production for individual logical trajectories over $z_\tau \to z'_\tau$ during the computation interval $(\tau, \tau')$.

The following implements an analogous form of quasistatic computation that allows us to easily calculate the energy associated with implementing the computation $M_{z_\tau \to z_{\tau'}}$, assuming the system started in $z_\tau$ and ends in $z_{\tau'}$. This also requires an ancillary copy of $\mathcal{Z}$, denoted $\mathcal{Z}'$, to allow for the full range of logical opera-

tions. (Note that some are impossible if restricted to a single copy of $\mathcal{Z}$ and continuous-time rate-equation evolution.) Due to detailed balance, the rate equation dynamics 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:

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

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

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

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

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

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:

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

See Fig. 10. 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 $E(z, z', \tau^+) = k_{\text{B}}T \ln(|\mathcal{Z}|/\Pr(Z_\tau = z))$ 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 $z_\tau$, then the associated work

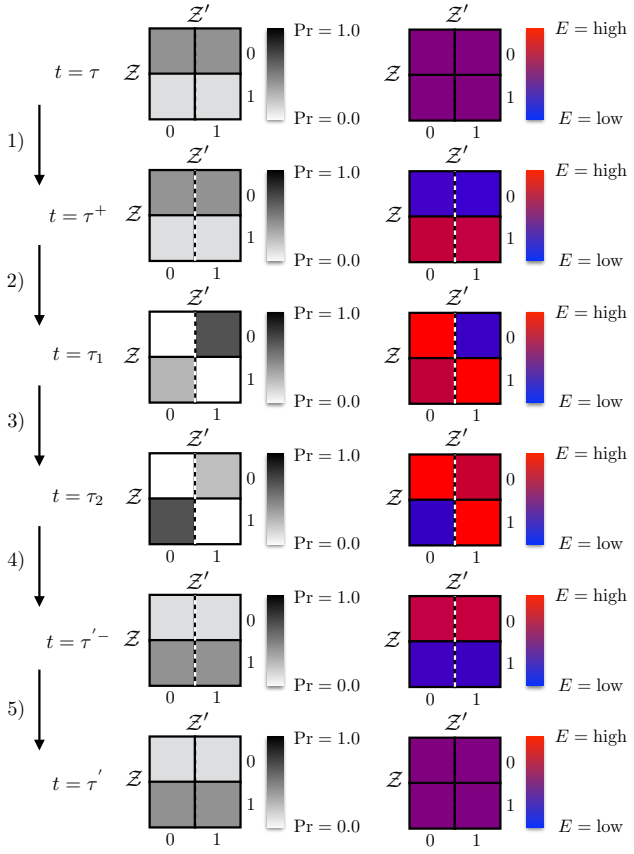


FIG. 10. Quasistatic agent implementing the Markov chain $M_{z_\tau \to z'_\tau}$ in the system $\mathcal{Z}$ over the time interval $[\tau, \tau']$ using ancillary copy $\mathcal{Z}'$ in five steps: Epoch 1: Energy landscape is instantaneously brought into equilibrium with the distribution over the 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 $\Pr(Z'_{\tau_1} = z' | Z_{\tau_1} = z) = M_{z \to z'}$. 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.

produced is opposite the energy change:

$$\begin{aligned} \left\langle W^{\text{prod},1}_{|Z_\tau^\theta = z_\tau, Z_{\tau'}^\theta = z_{\tau'}} \right\rangle &= E(z_\tau, z', \tau) - E(z_\tau, z', \tau^+) \\ &= \xi + k_{\text{B}}T \ln \frac{\Pr(Z_\tau^\theta = z_\tau)}{|\mathcal{Z}|} \ . \end{aligned}$$

$\left\langle W^{\text{prod},1}_{|Z_\tau^\theta = z_\tau, Z_{\tau'}^\theta = z_{\tau'}} \right\rangle$ denotes that the work is produced in the 1st stage, conditioned on the estimated distributions $Z_\tau^\theta$ and $Z_{\tau'}^\theta$, and initial and final states $z_\tau$ and $z_{\tau'}$. Note that we also condition on $Z_{\tau'}^\theta = z_{\tau'}$, since work production in this phase is unaffected by the end state of the computation.

*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 $\mathcal{Z}$:

$$\Pr(Z_{\tau_1} = z, Z'_{\tau_1} = z') = \Pr(Z_\tau = z) M_{z \to z'}$$
$$E(z, z', \tau_1) = -k_{\mathrm{B}} T \ln \Pr(Z^\theta_\tau = z) M_{z \to z'} \ .$$

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:

$$\left\langle W^{\mathrm{prod},1}_{|Z^\theta_\tau = z_\tau, Z^\theta_{\tau'} = z_{\tau'}} \right\rangle$$
$$= -\sum_{z,z'} \int_{\tau^+}^{\tau_1} dt \Pr(Z'_t = z', Z_t = z | Z_\tau = z_\tau) \partial_t E(z, z', t) \ .$$

As the system $\mathcal{Z}$ remains in $z_\tau$ over the interval:

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

the work production simplifies to:

$$\left\langle W^{\mathrm{prod},1}_{|Z^\theta_\tau = z_\tau, Z^\theta_{\tau'} = z_{\tau'}} \right\rangle$$
$$= -\sum_{z'} \int_{\tau^+}^{\tau_1} dt \Pr(Z'_t = z' | Z_t = z_\tau) \partial_t E(z_\tau, z', t) \ .$$

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

$$E(z_\tau, z', t) = -k_{\mathrm{B}} T \ln \Pr(Z'_t = z' | Z_t = z_\tau) \Pr(Z^\theta_t = z_\tau) \ .$$

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

$$\Pr(Z'_t = z' | Z_t = z_\tau) \Pr(Z^\theta_t = z_\tau)$$
$$= \Pr(Z'_t = z' | Z_t = z_\tau) \Pr(Z^\theta_\tau = z_\tau) \ .$$

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

$$\left\langle W^{\mathrm{prod},2}_{|Z^\theta_\tau = z_\tau, Z^\theta_{\tau'} = z_{\tau'}} \right\rangle$$
$$= -k_{\mathrm{B}} T \int_{\tau^+}^{\tau_1} dt \sum_{z'} \Pr(Z'_t = z' | Z_t = z_\tau)$$
$$\times \partial_t \ln \Pr(Z'_t = z' | Z_t = z_\tau) \Pr(Z^\theta_\tau = z_\tau)$$
$$= -k_{\mathrm{B}} T \int_{\tau^+}^{\tau_1} dt \sum_{z'} \Pr(Z'_t = z' | Z_t = z_\tau)$$
$$\times \frac{\Pr(Z^\theta_\tau = z_\tau) \partial_t \Pr(Z'_t = z' | Z_t = z_\tau)}{\Pr(Z'_t = z' | Z_t = z_\tau) \Pr(Z^\theta_\tau = z_\tau)}$$
$$= -k_{\mathrm{B}} T \int_{\tau^+}^{\tau_1} dt \sum_{z'} \partial_t \Pr(Z'_t = z' | Z_t = z_\tau)$$
$$= -k_{\mathrm{B}} T \int_{\tau^+}^{\tau_1} dt \, \partial_t \sum_{z'} \Pr(Z'_t = z' | Z_t = z_\tau)$$
$$= -k_{\mathrm{B}} T \int_{\tau^+}^{\tau_1} dt \, \partial_t 1$$
$$= 0 \ .$$

The resulting joint distribution over the ancillary and primary system matches the desired computation:

$$\Pr(Z = z, Z' = z') = \Pr(Z_\tau = z) M_{z \to z'} \ . \tag{C2}$$

*3. Swap:* Over the time interval $(\tau_1, \tau_2]$, slowly 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:

$$\left\langle W^{\mathrm{prod},3}_{|Z^\theta_\tau = z_\tau, Z^\theta_{\tau'} = z_{\tau'}} \right\rangle = 0 \ .$$

The resulting joint distribution over the ancillary and primary system matches a flip of the desired computation:

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

*4. Quasistatically evolve:* Over time interval $(\tau_2, \tau')$, quasistatically evolve the energy landscape from:

$$E(z, z', \tau_2) = -k_{\mathrm{B}} T \ln \Pr(Z^\theta_\tau = z') M_{z' \to z}$$

to

$$E(z, z', \tau'^{-}) = -k_{\mathrm{B}} T \ln \frac{\sum_{z''} \Pr(Z^\theta_\tau = z'') M_{z'' \to z}}{|\mathcal{Z}|}$$
$$= -k_{\mathrm{B}} T \ln \frac{\Pr(Z^\theta_{\tau'} = z)}{|\mathcal{Z}|} \ .$$

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

$$\left\langle W^{\text{prod},4}_{|Z^\theta_\tau = z_\tau, Z^\theta_{\tau'} = z_{\tau'}} \right\rangle = 0 \ .$$

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

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

having undergone a mapping from its original state at time $\tau$, 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 $E(z, z', \tau') = \xi$. The associated work production, given that the system ends in the state $z_{\tau'}$, is:

$$\begin{aligned}\left\langle W^{\text{prod},5}_{|Z^\theta_\tau = z_\tau, Z^\theta_{\tau'} = z_{\tau'}} \right\rangle &= E(z'_\tau, z', \tau^{'-}) - E(z'_\tau, z', \tau') \\ &= -\xi - k_{\text{B}} T \ln \frac{\Pr(Z^\theta_{\tau'} = z_{\tau'})}{|\mathcal{Z}|} \ .\end{aligned}$$

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

$$\begin{aligned}\left\langle W^{\text{eff}}_{|Z^\theta_\tau = z_\tau, Z^\theta_{\tau'} = z_{\tau'}} \right\rangle &= \sum_{\text{i}=1}^{5} \left\langle W^{\text{prod,i}}_{|Z^\theta_\tau = z_\tau, Z^\theta_{\tau'} = z_{\tau'}} \right\rangle \\ &= k_{\text{B}} T \ln \frac{\Pr(Z^\theta_\tau = z_\tau)}{\Pr(Z^\theta_{\tau'} = z_{\tau'})}.\end{aligned}$$

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

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

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

## Appendix D: 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. (13). However, when the transducer is designed to harness an input process with $\epsilon$-machine $T$, such that:

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

the work production simplifies. To see this, we express Eq. (13) in terms of the estimated distribution $\Pr(Y^\theta_{0:L})$, 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 $\Pr(Y^\theta_{0:L} = y_{0:L} | S_0 = s_0) \Pr(X_0 = s_0) \neq 0$, which guarantees that $\sum_{x_{i+1}} \theta^{(y_i)}_{x_i \to x_{i+1}} \neq 0$. If this constraint is not satisfied, the agent will dissipate infinite work, as it implies $\Pr(X^\theta_i = x_i, Y^\theta_i = x_i) = 0$ for some $i$. Thus, we use the expression $M_{xy \to x'y'} = \delta_{x', \epsilon(x,y)} / |\mathcal{Y}|$ in the work production:

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

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 $\epsilon$-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 $\epsilon$-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:

$$\left\langle W^{\text{eff}}_{|Y^\theta_{0:L}=y_{0:L}} \right\rangle = k_{\text{B}}T \ln|\mathcal{Y}| + k_{\text{B}}T \sum_{x_{0:L+1}} \Pr(X_0 = x_0) \prod_{j=1}^{L} \delta_{x_j,\epsilon(x_0,y_{0:j})} \ln \prod_{i=0}^{L-1} \frac{\Pr(Y^\theta_i = y_i | X^\theta_i = x_i) \Pr(X^\theta_i = x_i)}{\Pr(X^\theta_{i+1} = x_{i+1})} \quad \text{(D1)}$$

$$= k_{\text{B}}T \ln|\mathcal{Y}| + k_{\text{B}}T \sum_{x_0} \Pr(X_0 = x_0) \ln \prod_{i=0}^{L-1} \Pr(Y^\theta_i = y_i | X^\theta_i = \epsilon(x_0, y_{0:i})) \quad \text{(D2)}$$

$$+ k_{\text{B}}T \sum_{x_0} \Pr(X_0 = x_0) \ln \frac{\Pr(X^\theta_0 = x_0)}{\Pr(X^\theta_L = \epsilon(x_0, y_{0:L}))} \ , \quad \text{(D3)}$$

where $y_{0:0}$ denotes the null input, which leaves the state fixed under the $\epsilon$-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 $s^*$. If we recognize that if we initiate the $\epsilon$-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:

$$\begin{aligned}&\left\langle W^{\text{eff}}_{|Y^\theta_{0:L}=y_{0:L}} \right\rangle \\ &= k_{\text{B}}T \sum_{x_0} \delta_{x_0,s^*} \ln \prod_{i=0}^{L-1} \Pr(Y^\theta_i = y_i | Y^\theta_{0:i} = y_{0:i}) \\ &\quad + k_{\text{B}}TL \ln|\mathcal{Y}| + k_{\text{B}}T \sum_{x_0} \delta_{x_0,x^*} \ln \frac{\delta_{x_0,s^*}}{\Pr(X^\theta_L = \epsilon(x_0, y_{0:L}))} \\ &= k_{\text{B}}T (\ln \Pr(Y^\theta_{0:L} = y_{0:L}) + L \ln|\mathcal{Y}| \\ &\quad - \ln \Pr(X^\theta_L = \epsilon(s^*, y_{0:L}))) \ .\end{aligned}$$

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 $\epsilon$-machine. From a machine learning perspective, the model is also characterized by the $\epsilon$-machine $\theta^{(y)}_{s \to s'}$ for the process $\Pr(Y^\theta_{0:\infty})$. The additional term $k_{\text{B}}TL \ln|\mathcal{Y}|$ is the work production that comes from exhausting fully randomized outputs and does not change depending on the underlying model.

The final term $-k_{\text{B}}T \ln \Pr(X^\theta_L = \epsilon(s^*, y_{0:L})))$ does directly depend on the model. $\Pr(X^\theta_L = x)$ is the distribution over agent states $\mathcal{X}$ at time $L\tau$ if the demon is driven by a the estimated input distribution $Y^\theta_{0:L}$. 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 $\Pr(X^\theta_L)$. 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 it so 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 $\Pr(X^\theta_L = x)$, outputs the distribution $\Pr(X^\theta_{L+1} = x) = \delta_{x,s^*}$, and so costs:

$$W^{\text{reset}}_{|Y^\theta_{0:L}=y_{0:L}} = k_{\text{B}}T \ln \Pr(X^\theta_L = \epsilon(s^*, y_{0:L})) \ .$$

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:

$$\left\langle W^{\text{eff}}_{|Y^\theta_{0:L}=y_{0:L}} \right\rangle = k_{\text{B}}T (\ln \Pr(Y^\theta_{0:L} = y_{0:L}) + L \ln|\mathcal{Y}|). \quad \text{(D4)}$$

### Appendix E: Designing Agents

To design a predictive thermodynamic agent, its hidden states must match the states of the $\epsilon$-machine at every time step $X_i = S^\theta_i$. 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:

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

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 $S^\theta_{i+1} = x'$ given that the current causal state is $S^\theta_i = x$ and output of the $\epsilon$-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 hidden states of the agent match the transitions of the causal states of the $\epsilon$-machine.

In this way, if $y_{0:L}$ is a sequence that could be produced by the $\epsilon$-machine, we have designed the agent to stay synchronized to the causal state of the input $X_i = S^\theta_i$, so that the ratchet is predictive of the process $\Pr(Y^\theta_{0:\infty})$ 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 $\epsilon$-machine cannot produce $y$ from the causal state $x$. This corresponds to a disallowed transition of our model $\theta^{(y)}_{x\to x'} = 0$. In these cases, we arbitrarily choose the next state to be the same $\delta_{x,x'}$. There are many possible choices for this case; 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 $\Pr(Y^\theta_{0:\infty})$ and its $\epsilon$-machine $\theta$ to the logical architecture of a maximum work-producing agent $M$.

As yet, this does not fully specify agent behavior, since it leaves out the estimated input distribution $\Pr(Y^\theta_i = y, X^\theta_i = x)$. 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 $\epsilon$-machine's causal states, we know that the distribution matches the distribution over causal states states and inputs:

$$\Pr(Y^\theta_i = y_i, X^\theta_i = s_i) = \Pr(Y^\theta_i = y_i, S^\theta_i = s_i) \ ,$$

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

$$\Pr(Y^\theta_{0:i+1} = y_{0:i+1}, S^\theta_{0:i+2} = s_{0:i+2}) = \delta_{s_0,s^*} \prod_{j=0}^{i} \theta^{(y_j)}_{s_i\to s_{i+1}} \ .$$

Summing over the variables besides $Y^\theta_i$ and $S^\theta_i$, we obtain an expression for the estimated agent distribution in terms of just the $\epsilon$-machine's HMM:

$$\Pr(Y^\theta_i = y_i, X^\theta_i = s_i) = \sum_{y_{0:i}, s_{0:i}, s_{i+1}} \delta_{s_0,s^*} \prod_{j=0}^{i} \theta^{(y_j)}_{s_j\to s_{j+1}} \ .$$

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


[1] W. Thomson. Kinetic theory of the dissipation of energy. *Nature*, pages 441–444, 9 April 1874.

[2] J. C. Maxwell. *Theory of Heat*. Longmans, Green and Co., London, United Kingdom, ninth edition, 1888.

[3] T. Sagawa. Thermodynamics of information processing in small systems. *Prog. Theo. Phys.*, 127:XXX, 2012.

[4] J. M. R. Parrondo, J. M. Horowitz, and T. Sagawa. Physics of information. *Nature Physics*, 11(2):131, 2015.

[5] S. Shalev-Shwatrz and S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[6] 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.

[7] C. H. Bennett. Demons, engines and the Second Law. *Sci. Am.*, 257(5):108–116, 1987.

[8] R. Landauer. Irreversibility and heat generation in the computing process. *IBM J. Res. Develop.*, 5(3):183–191,

1961.

[9] L. Szilard. On the decrease of entropy in a thermodynamic system by the intervention of intelligent beings. *Z. Phys.*, 53:840–856, 1929.

[10] A. B. Boyd and J. P. Crutchfield. Maxwell demon dynamics: Deterministic chaos, the Szilard map, and the intelligence of thermodynamic systems. *Phys. Rev. Lett.*, 116:190601, 2016.

[11] 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.

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

[13] J. M. Gold and J. L. England. Self-organized novelty detection in driven spin glasses. *arXiv:1911.07216*, 2019.

[14] 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.

[15] D. Jimenez Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

[16] 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.

[17] J. P. Crutchfield and D. P. Feldman. Regularities unseen, randomness observed: Levels of entropy convergence. *CHAOS*, 13(1):25–54, 2003.

[18] S. Deffner and C. Jarzynski. Information processing and the second law of thermodynamics: An inclusive, Hamiltonian approach. *Phys. Rev. X*, 3:041003, 2013.

[19] M. Esposito and C. van den Broeck. Second law and Landauer principle far from equilibrium. *Europhys. Lett*, 95:40004, 2011.

[20] S. Deffer and E. Lutz. Information free energy for nonequilibrium states. *arXiv:1201.3888*, 2012.

[21] G. E. Crooks. Entropy production fluctuation theorem and the nonequilibrium work relation for free energy differences. *Phys. Rev. E*, 60(3), 1999.

[22] T. Speck and U. Seifert. Distribution of work in isothermal nonequilibrium processes. *Phys. Rev. E*, 70(066112), 2004.

[23] A. Kolchinsky and D. H. Wolpert. Dependence of dissipation on the initial distribution over states. *J. Stat. Mech.: Th. Expt.*, page 083202, 2017.

[24] P. M. Riechers and M. Gu. Initial-state dependence of thermodynamic dissipation for any quantum process. *arXiv:2002.11425*, 2020.

[25] 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.

[26] A. B. Boyd, D. Mandal, and J. P. Crutchfield. Identifying functional thermodynamics in autonomous Maxwellian ratchets. *New J. Physics*, 18:023049, 2016.

[27] 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.

[28] 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.

[29] N. Merhav. Sequence complexity and work extraction. *J. Stat. Mech.*, page P06037, 2015.

[30] N. Merhav. Relations between work and entropy production for general information-driven, finite-state engines. *J. Stat. Mech.: Th. Expt.*, 2017:1–20, 2017.

[31] A. J. P. Garner, J. Thompson, V. Vedral, and M. Gu. Thermodynamics of complexity and pattern manipulation. *Phys. Rev. E*, 95(042140), 2017.

[32] N. Barnett and J. P. Crutchfield. Computational mechanics of input-output processes: Structured transformations and the $\epsilon$-transducer. *J. Stat. Phys.*, 161(2):404–451, 2015.

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

[34] J. G. Brookshear. *Theory of computation: Formal languages, automata, and complexity.* Benjamin/Cummings, Redwood City, California, 1989.

[35] G. D. Kirchhoff. *Ann. Phys.*, 75:1891, 1848.

[36] J. W. Gibbs. *The Scientific Papers of J. Willard Gibbs.* Longmans, Green, New York, New York, 1906.

[37] J. C. Maxwell. *A Treatise on Electricity and Magnetism, vol. I and II.* Dover Publications, Inc., New York, New York, third edition, 1954.

[38] L. Onsager. Reciprocal relations in irreversible processes, I. *Phys. Rev.*, 37(4):405–426, 1931.

[39] I. Prigogine. Modération et transformations irréversibles des systèmes ouverts. *Bulletin de la Classe des Sciences, Académie Royale de Belgique*, 31:600–606, 1945.

[40] I. Prigogine and P. Glansdorff. *Thermodynamic Theory of Structure, Stability and Fluctuations.* Wiley-Interscience, London, 1971.

[41] G. Falasco, R. Rao, and M. Esposito. Information thermodynamics of Turing patterns. *Phys. Rev. Let.*, 121:108301, 2018.

[42] A. M. Turing. The chemical basis of morphogenesis. *Trans. Roy. Soc., Series B*, 237:5, 1952.

[43] R. Hoyle. *Pattern Formation: An Introduction to Methods.* Cambridge University Press, New York, 2006.

[44] M. Cross and H. Greenside. *Pattern Formation and Dynamics in Nonequilibrium Systems.* Cambridge University Press, Cambridge, United Kingdom, 2009.

[45] W. Heisenberg. Nonlinear problems in physics. *Physics Today*, 20:23–33, 1967.

[46] D. Ruelle and F. Takens. On the nature of turbulence. *Comm. Math. Phys.*, 20:167–192, 1971.

[47] 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.

[48] C. E. Shannon. A mathematical theory of communication. *Bell Sys. Tech. J.*, 27:379–423, 623–656, 1948.

[49] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* Wiley-Interscience, New York, second edition,

2006.

[50] 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.

[51] C. E. Shannon. The bandwagon. *IEEE Transactions on Information Theory*, 2(3):3, 1956.

[52] A. Turing. On computable numbers, with an application to the Entschiedungsproblem. *Proc. Lond. Math. Soc.*, 42, 43:230–265, 544–546, 1937.

[53] N. Wiener. *Cybernetics: Or Control and Communication in the Animal and the Machine*. MIT, Cambridge, 1948.

[54] W. Ross Ashby. *An Introduction to Cybernetics*. John Wiley and Sons, New York, second edition, 1960.

[55] E. T. Jaynes. Information theory and statistical mechanics. *Phys. Rev.*, 106(4):620–630, 1957.

[56] E. T. Jaynes. The minimum entropy production principle. *Annu. Rev. Phys. Chem.*, 31:579–601, 1980.

[57] J. P. Crutchfield. Between order and chaos. *Nature Physics*, 8(January):17–24, 2012.

[58] 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.

[59] N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. Shaw. Geometry from a time series. *Phys. Rev. Let.*, 45:712, 1980.

[60] 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.

[61] J. P. Crutchfield and B. S. McNamara. Equations of motion from a data series. *Complex Systems*, 1:417 – 452, 1987.

[62] L. Brillouin. *Science and Information Theory*. Academic Press, New York, second edition, 1962.

[63] C. H. Bennett. Thermodynamics of computation—a review. *Intl. J. Theo. Phys.*, 21:905, 1982.

[64] 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.

[65] J. P. Crutchfield and O. Gornerup. Objects that make objects: The population dynamics of structural complexity. *J. Roy. Soc. Interface*, 3:345–349, 2006.

[66] J. England. Statistical physics of self-replication. *J. Chem. Phys.*, 139(121923), 2013.

[67] Viviana Serreli, Chin-Fa Lee, Euan R. Kay, and David A. Leigh. A molecular information ratchet. *Nature*, 445(7127):523–537, 2007.

[68] J. Thompson, A. J. P. Garner, V. Vedral, and M. Gu. Using quantum theory to simplify input–output processes. *npj Quant. Info.*, 3(6), 2017.

[69] S. Loomis and J. P. Crutchfield. Thermal efficiency of quantum memory compression. *Phys. Rev. Lett.*, in press, 2020. arXiv:1911.00998.

[70] 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.

[71] P. Dayan, G. E. Hinton, R. M. Neal, and R. S. Zemel. The Helmholtz machine. *Neural computation*, 7(5):889–904, 1995.

[72] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521:436–444, May 2015.

[73] R. Landauer. Information is physical. *Physics Today*, pages 23–29, May 1991.

[74] C. C. Strelioff and J. P. Crutchfield. Bayesian structural inference for hidden processes. *Physical Review E*, 89:042119, 2014. Santa Fe Institute Working Paper 13-09-027, arXiv:1309.1392 [stat.ML].

[75] T. Wu and M. Tegmark. Toward an artificial intelligence physicist for unsupervised learning. *Phys. Rev. E*, 100(3):033311, 2019.

[76] J. P. Crutchfield. The calculi of emergence: Computation, dynamics, and induction. *Physica D*, 75:11–54, 1994.

[77] C. Jarzynski. Equalities and inequalities: Irreversibility and the second law of thermodynamics at the nanoscale. *Annu. Rev. Cond. Matt. Phys.*, 2(1), 2011.

[78] 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.